Fuzzy Scheduling

Wolfgang Slany

Christian Doppler Laboratory for Expert Systems
E184/2, TU Wien, A-1040 Vienna, Austria, Europe
Phone: +43–1–58801–6141 Fax: +43–1–5055304
URL: http://www.dbai.tuwien.ac.at:8080/staff/slany.html
E-Mail: wsi@vexpert.dbai.tuwien.ac.at

CD–Technical Report 94/66

Christian Doppler Laboratory
for Expert Systems

Technische Universität Wien
Institut für Informationssysteme
Abteilung für Datenbanken und Expertensysteme

DISSERTATION

Fuzzy Scheduling

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Technisch-Naturwissenschaftliche Fakultät

von

Dipl.-Ing. Wolfgang Slany
Mariannengasse 21/5, A-1090 Wien
Matrikelnummer: 85 25 493
geboren am 14. November 1966 in Wien

Wien, im Juni 1994

To my wife Kyōko,
and to my family,
for love and support

Deutsche Kurzfassung der Dissertation

Fuzzy Logic hat sich bereits in vielen praktischen Anwendungen weltweit bewährt. Typische Anwendungsgebiete in der Industrie sind die Überwachung und Steuerung von einfachen technischen Prozessen oder die Vorhersage von schwer erfaßbaren technischen Größen. Auch im Bereich der Planung verspricht der Einsatz von Fuzzy Logic Verbesserungen. Um die möglichen Synergien mit dem Bereich "Wissensbasiertes Scheduling" zu untersuchen, wurden in dieser Arbeit Methoden der Fuzzy Logic mit jenen aus dem Bereich zeitlicher Planung im Produktionsbereich kombiniert. Die Erwartung war, daß Fuzzy Logic einerseits dabei helfen kann, unscharf formuliertes Expertenwissen einfach zu modellieren, andererseits dabei, ungenaue Daten so zu verarbeiten, daß die vorhandene Ungenauigkeit besser ausgenützt werden kann. Das Ziel war, die Qualität der erzeugten Pläne zu erhöhen, bei den Produktionskosten zu sparen, den Durchsatz zu verbessern, die Auslastung der Maschinen, Arbeitskräfte und sonstigen Ressourcen zu optimieren, und schließlich in Notfällen schnell eine gute Alternative zum Originalplan parat zu haben.

Konkret wurden allgemeine Werkzeuge zur Modellierung und Verarbeitung von unscharfen Regeln (Einschränkungen) und unsicheren Daten (Meßwerten, Zeiten) erstellt. Als beispielhaftes Anwendungsgebiet dient der Stahlerzeugungsprozeß. Die von mir entworfenen Programme FLIP++ (Fuzzy Logic Inference Processor in C++), ConFLIP++ (Fuzzy Constraints, aufbauend auf FLIP++) und DynaFLIP++ (Dynamische Constraints-Generierung, aufbauend auf ConFLIP++, zur direkten Verwendung im Planungsprogramm DéjàVu geeignet) erlauben es, diese Fuzzy Einschränkungen mit graphischer Unterstützung (InterFLIP++) zu erstellen, zu verändern, zu verarbeiten und zur Erstellung von Plänen zu benützen. In der vorliegenden Arbeit wird auch ein von mir entwickeltes Verfahren erläutert, mit dessen Hilfe die beschreibenden Parameter zuverlässig optimiert werden können. Mit Hilfe dieses Verfahrens kann auf leicht verständlich Weise überprüft werden, ob die Wissensbasis aller Fuzzy Einschränkungen eine "vernünftige" Entscheidungsbasis bezüglich einer Menge von früheren Referenzentscheidungen darstellt. Die Unscharfe Einschränkungen erlauben es, auf *präzise* Art und Weise anzugeben, bis zu welchem Wert Einschränkungen verletzt werden dürfen, und gleichzeitig *exakt* festzuhalten, welche Werte als wie wünschenswert einzustufen sein sollen. Beispielsweise muß unter anderem sichergestellt werden, daß bei zwei hintereinander zu produzierenden Stahlsorten die Überschneidung der Analysenintervalle für Kupfer $\geq 0.03\%$ sein soll, damit sie ohne zusätzliche Maßnahmen hergestellt werden können. Was nun, wenn diese Ungleichung nicht absolut gilt und ein Wert von 0.029% auch noch akzeptiert werden könnte, insbesondere dann, wenn sich alle anderen Werte in "sicheren" Bereichen befinden und sich dadurch eine wesentlich bessere Produktionsreihenfolge ergäbe? Was, wenn wir angeben wollen, daß ein Wert von 0.036% als "sicherer" einzustufen sein soll als ein Wert von nur 0.031%? Durch Fuzzy Regeln können dem Planungs-Programm solche *akzeptablen* Verletzungen von Einschränkungen auf einfache Art und Weise mitgeteilt werden. Ebenso wird dadurch festgelegt, welche Verletzung einer Einschränkung als wie gut oder wie schlecht einzustufen sein soll. Weiters fällt es mit Fuzzy Logic sehr leicht, anzugeben, welche Art von Kompromissen erlaubt sein soll. Es ist auch sehr einfach, die zu berücksichtigenden Kriterien verschieden stark zu gewichten, um ihren unterschiedlichen Bedeutungen Rechnung zu tragen.

Weitere Aspekte, die sich sehr leicht mit Fuzzy Logic modellieren lassen, betreffen die Verarbeitung von Möglichkeitsverteilungen von Werten. So können Größen, deren genaue Werte zum Planungszeitpunkt noch nicht festliegen, z.B. die Gießgeschwindigkeit während der Verarbeitung eines bestimmten Auftrags auf der Stranggußanlage, mittels Möglichkeitsverteilungen modelliert und dadurch trotzdem für die Planerstellung berücksichtigt werden. Auch die Vorhersage von möglichen Werten bestimmter wichtiger Parameter mittels zum Teil nicht völlig bekannten Daten wird mittels Fuzzy Logic leichter handhabbar. Man stelle sich vor, die Verwendungsdauer für einen bestimmten Teil einer Anlage betrage im Normalfall 240 Minuten, kann aber, abhängig von verschieden Parametern, die zum Teil erst zur Produktionszeit festgelegt werden, auch nur 100 oder aber bis zu 300 Minuten betragen. Nun ist es meistens für den Menschen zu mühsam, die vielen zum Teil nur ungenau bekannten Einflußfaktoren zu berücksichtigen, um einen jeweils neuen Schätzwert für diesen Parameter zu berechnen. Der Computer hat es da mit Hilfe der Fuzzy Logic wesentlich leichter, sogar eine ganze Möglichkeitsverteilung für den Parameter zu schätzen und bei weiteren Entscheidungen zu berücksichtigen. Daher wurde in dieser Arbeit auch das Wissen und das notwendige Modell für eine solche Aufgabe zur Ermittlung der Möglichkeitsverteilungen der voraussichtlichen Lebensdauer eines Gießrohres als Teil einer Stranggußanlage erläutert.

Die theoretische Komplexität von zeitlichen Planungsproblemen wird zwar durch die Wissensrepräsentation mittels Fuzzy Einschränkungen und Möglichkeitsverteilungen nicht reduziert, da der Suchraum im allgemeinen eher um einige Kompromißlösungen vergrössert wird. Andererseits erlaubt die graduelle Erfüllung der Einschränkungen den Einsatz und die effiziente Steuerung von Such-Heuristiken, die sich in der Praxis bereits extrem bewährt haben. In der vorliegenden Arbeit wurden erstmals mehrere solche auf vollständigen aber suboptimalen Lösungen operierende Such-Heuristiken mit den Fuzzy Methoden zur Wissensrepräsentation kombiniert und damit wesentlich bessere Ergebnisse erzielt, als mit traditionellen Such-Algorithmen ohne Fuzzy Wissensrepräsentation.

Die vorgestellten Methoden sind auch zur Lösung zahlreicher anderer Entscheidungsprobleme aus der Realität sehr gut geeignet. Trotzdem konzentriert sich diese Arbeit auf das Problem der zeitlichen Planung unter Unsicherheit, da die Forschung und Entwicklung auf diesem Gebiet in den letzten Jahren einen bedeutenden Aufschwung erlebt hat. Im Anhang findet sich eine ausführliche Bibliographie, zum Teil mit Kommentaren versehen, zum Thema der Dissertation.

Abstract

Real-world scheduling is decision making under vague constraints of different importance, often using uncertain data, where compromises between antagonistic criteria are allowed. The author explains in theory and by detailed examples a new combination of fuzzy set based constraints and iterative improvement repair based heuristics that help to model these scheduling problems. He simplifies the mathematics needed for a method of eliciting the criteria's importances from human experts. He introduces a new consistency test for configuration changes. This test also helps to evaluate the sensitivity to configuration changes. He describes the implementation of these concepts in his fuzzy logic inference processor library FLIP++, in his fuzzy constraint library ConFLIP++, in his dynamic constraint generation library DynaFLIP++, and in his heuristic repair library DéjàVu. All these libraries are implemented in a layered framework enhanced by his common user interface InterFLIP++. The benchmark application to compare his fuzzy constraint iterative improvement repair heuristic with constructive methods based on classic constraints is a scheduling system for a continuous caster unit in a steel plant. In addition, an earlier fuzzy scheduling system that was applied to another steel plant, as well as a fuzzy expert system that predicts maintenance intervals for the continuous caster unit are described. This thesis also discusses research issues and challenges as well as previous work done in the field of fuzzy scheduling and related areas, and provides an exhaustive and partly annotated bibliography concerning its subject. An online-version of the thesis is located at URL: "ftp://mira.dbai.tuwien.ac.at/pub/slany/thesis.ps.Z".

Keywords

Fuzzy scheduling; fuzzy constraint satisfaction problems; fuzzy multiple criteria optimization; fuzzy qualitative modeling; fuzzy decision making; trade-offs; compromising; importance scale; priority of constraints; repair based heuristic (iterative improvement) versus constructive algorithm; tabu list; fuzzy logic for production control and CIM; scheduling in steelmaking; fuzzy resource allocation; fuzzy planning and design; non-fuzzy uncertainty management in scheduling; fuzzy knowledge representation; possibility distribution; possibilistic modeling; knowledge based scheduling; fuzzy constraint relaxation; real-world scheduling; maintenance interval prediction; fuzzy expert system; continuous caster scheduling; fuzzy linear programming; computational complexity; benchmarking.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The question of whether a computer can think is no more interesting than the question whether a submarine can swim.

Edsger W. Dijkstra

A few years ago, one of the authors happened to be dining in a Beverly Hills restaurant near Mel Brooks. The waitress appeared and listed the evening specials for him. One appetizer, she said, was yellowtail grilled on one side and raw on the other. "Hey, what is this? It's either sushi *or it* isn't!" *he cried ...*

Daniel McNeill and Paul Freiberger, Fuzzy Logic

I was fully cognizant that I was doing something that would spark controversy.

Lotfi A. Zadeh

This Chapter introduces fuzzy logic as a part of artificial intelligence in general and motivates the choice of the thesis's subject, fuzzy scheduling. Furthermore, it gives a condensed overview about the structure of the thesis.

1.1 Fuzzy logic in artificial intelligence

In 1948, Alan Turing wrote a paper [403] marking the begin of a new era, the era of the intelligent machine, which raised questions that still remain unanswered today.

This era was heavily influenced by the appearance of the computer, a machine that allowed humans to automate their way of thinking.

However, human thinking is not exact. If you had to park your car *precisely* in one place, you would have extreme difficulties. To allow computers to really mimic the way humans think, the theories of fuzzy sets and fuzzy logic were created. They should be viewed as formal mathematical theories for the representation of uncertainty, which is essential for the management of real world systems as it mimics the crucial ability of the human mind to summarize data and focus on decision relevant information.

Marvin Minsky, one of the founding fathers of artificial intelligence, once defined the latter as

> *"... the science of making machines do things that would require intelligence if done by men."*

Similarly, Lotfi A. Zadeh, who in 1965 wrote the founding paper on fuzzy set theory [447], once described the aim of this theory as being

> *"the construction of smarter machines."*

Zadeh recently coined the term MIQ (machine intelligence quotient) to refer to this particular aspect of the growing number of intelligent consumer products and industrial systems [221].

Proponents of the so-called 'strong' artificial intelligence believe that eventually, these machines will be as intelligent as we human beings are now. Thinking positively about technology, everything that is conceivable to be solved by artificial means will eventually be realized if it is interesting enough. Of course some intellectual processes have been shown to be emergent properties, such as 'consciousness'. The concept of emergent properties of complex systems was first observed by von Bertalanffy [24] in the 1920s in his study of complex biological systems. He noticed that complex assemblies of entities organized in particular ways can reveal unique properties not possessed by the individual entities alone. Emergent properties cease to exist if the whole is broken into components or if the components are organized in a different way. Additionally, emergent properties cannot be understood by the study of isolated components. Similar to the notion of a critical mass in physics, an emergent property will suddenly pop up when a sufficient amount of mass has been accumulated. Contrary to reductionistic approaches, these approaches normally assume a holistic view of the world, i.e. something complex can be more than simply the accumulation or 'sum of its parts'. Of course, as with the atomic bomb, which was in a certain sense the first artificial application of the physical effect described above, the ethical aspects have to be carefully considered. One has to be aware that any technology can be used for good or for evil. However, not the technology in itself is good or bad, but instead the humans that use it are so, since technology has so far been only a tool for human beings. In the case of intelligence, this might be not true anymore, since advanced intelligence may entail new ethical needs, but these new forms of intelligence have not yet reached a level where ethical aspects become prevalent.

It is important to note that the term fuzzy logic is used in two distinct senses. In its narrower sense, fuzzy logic is only one branch of fuzzy set theory. Fuzzy set theory was invented by Zadeh to be able to better represent such everyday notions as the set of 'tall persons'. Of course, this set is defined vaguely, and persons will more or less be a member of it, i.e. member to a certain degree. Fuzzy logic in this narrow sense deals in a natural way with the representation and inference from such vaguely formulated or uncertain knowledge, similarly to classical logic which deals with crisp knowledge where statements can only be either true or false (well, almost, at least if you do not count the findings of Kurt Gödel). In recent years, however, it has become increasingly common to employ the term fuzzy logic in a much broader sense, making the difference between the notions of fuzzy set theory and fuzzy logic vanish. To avoid confusion, we follow in this Section the trend to use fuzzy logic in its general sense. In all other Chapters the term fuzzy logic is used in its narrower sense.

James Bezdek, editor in chief of the IEEE transactions on fuzzy systems, defined fuzzy logic in a delightful essay [25] to be one part of 'computational intelligence', altogether with such research areas as neural networks, evolutionary computation, and genetic algorithms. Bezdek contrasts the ABC's on intelligence: artificial, biological and computational. In the strictest sense, computational intelligence "depends on numerical data supplied by manufacturers and [does] not rely on 'knowledge'." Artificial intelligence, on the other hand, uses what Bezdek calls 'knowledge tidbits'. Heuristically constructed artificial intelligence such as an expert system is an example. Practicing knowledge engineers and neural smiths know the distinction is at times not precise. Expert extraction of feature data for training a layered perceptron certainly falls in the area of artificial intelligence. Using these features to train the layered perceptron is primarily computational. Fuzzy inference engines crafted by experts fall into the definition of artificial intelligence. Algorithmic tuning of the engine with raw data, however, is computational intelligence.

Even though the boundary between computational intelligence and artificial intelligence is not distinct, we can, making certain assumptions, monitor the volume of research activity in each. Indeed, the separate identities of computational intelligence and artificial intelligence are confirmed by inspection of the recent volume of publishing and patent activity [268].

However, the term 'Computational Intelligence' itself is not undisputed, since it had already been widely used to mean artificial intelligence before it was redefined by Bezdek, see for example the journal 'Computational Intelligence', published since

1985, the conference 'Computational Intelligence' taking place annually since 1988, and numerous other publications and organizations using the term in this traditional sense.

In both cases, artificial intelligence as well as fuzzy logic, one tries in some sense to imitate life in its problem-solving capability. The ways how to achieve this goal are different in many respects, but there are also many common points where the two fields overlap: Robert Marks [268] counted 4811 entries on fuzzy logic in the INSPEC data base from 1989 to 1993, containing citations from over 4000 selected journals, books, conference proceedings and technical reports – "22% of them [were] cross categorized in the expert system category, and 12% with neural networks." Based on various 'bean countings', Marks concludes that the overlapping areas cover, depending on the way to count, from 14% to 33%.

It should not be left untold that there has been a lot of scientific antagonism between fuzzy logic and artificial intelligence, and, accordingly, skeptics on both sides exist and treat the other side with reservation, if not with open hostility. There are many reasons for this, e.g. some critics of fuzzy logic credit the word 'fuzzy' for being too controversial and misleading in itself, others maintain that anything that can be done with fuzzy logic and fuzzy set theory can be done equally well with classical logic and probability theory [62][1], and still others insist on denying fuzzy logic the status of a logic itself [128]. Of course these claims were refuted [275, 192, or see discussions in the archives of the news-groups mentioned on page 16]. Fuzzy logic in its narrow sense is simply a logic *of* fuzziness, not a logic which *itself* is fuzzy. Just as the laws of probability are not random, so the laws of fuzziness are not vague.

On the other hand, critics of artificial intelligence have observed that the sometimes over-ambitious predictions made in the past did not come true. Some even go as far as to deny that there has been even one successful expert system implemented that really became used. Others believe that the aim to create artificial intelligence is useless and impossible on philosophical grounds. However, such views are likely to become muted with the passage of time and a better understanding of the basic ideas underlying the theories of both artificial intelligence and fuzzy logic. We observe nevertheless that, nurtured by the current success of fuzzy logic in the real world, dangerously unrealistic predictions and claims appear again. Bart Kosko, a respected scholar in the field and author of a best-selling textbook on 'Neural Networks and Fuzzy Systems' [228] for instance predicts for the next few decades fuzzy logic based natural language understanding, machines that write interesting novels and screenplays in a selected style such as Hemingway's, or even sex robots with a humanlike repertoire of behavior [275]. Some researchers suggest however that as attempts are made to make fuzzy systems larger, they will encounter sim-

[1] But Cheeseman also rejects nonmonotonic reasoning, default logic, and Dempster Shafer's theory, arguing that probabilities are better suited to model the world, and the other methods are at most harmless if not outright wrong. For an outline of his paper, see the annotation to his paper [62].

ilar difficulties as conventional reasoning methodologies. Fuzzy logic is certainly not a philosopher's stone solving all problems that confront us today. But it has a considerable potential for practical applications. The management of uncertainty will be of growing importance. This uncertainty can have various reasons, ranging from uncertainty due to the lack of knowledge or evidence, due to an abundance of complexity and information, to uncertainty due to the fast and unpredictable development of scientific, political, social, and other structures nowadays.

The applications of fuzzy technologies fall mainly into two categories: fuzzy control applications, which are often rather simple but very efficient fuzzy rule-based systems, such as autofocusing systems in cameras, washing machines, automobile transmissions, subway control, or even handwriting recognition. In these applications, fuzzy logic is used as a powerful knowledge representation technique that allows to hide unessential details and to handle uncertain data. However, their efficiency depends also heavily on the use of sensors and effectors, thus their success should really be explained by the interaction of these various parts. The second category consists of those much more complex systems that aim at supporting or even replacing a human expert. Such applications are exemplified by medical diagnosis systems, securities funds and portfolio selection systems, traffic control systems, fuzzy expert systems, and fuzzy scheduling systems. In this second category, there are still many problems that remain to be addressed, and there is an equally pressing need for a better understanding of how to deal with knowledge-based systems in which knowledge is both uncertain and imprecise.

Areas where fuzzy logic and artificial intelligence meet in current research include: fuzzy expert systems (e.g. for medical diagnosis or intelligent tutoring systems), theoretical investigations (e.g. combinations of fuzzy logic with modal logics and other forms of defeasible reasoning, i.e. based on questionable knowledge), machine learning (e.g. combinations of fuzzy logic with neural networks, genetic algorithms, associative memories, symbolic learning methods such as case based reasoning), robotics (involving motion control and planning capabilities, e.g. when flying a fully automated helicopter or driving a car on a freeway), pattern matching (e.g. face recognition), or constraint satisfaction problem solving methods (applied for example in manufacturing process scheduling, as in this thesis, or in bridge design).

Let us take a closer look at fuzzy expert systems as the archetypical spin off coming from the combination of techniques from fuzzy logic and artificial intelligence. Classical expert systems are computer programs that emulate the reasoning of human experts or perform in an expert manner in a domain for which no human expert exists. This could be due to a dangerous working environment or simply because of a domain that is to large for one human being. These expert systems typically reason with uncertain and imprecise information, using various methods besides fuzzy logic to handle them. There are many sources of imprecision and uncertainty. The knowledge that the expert systems embody is often not exact, in the same way as a human's knowledge is imperfect. The facts or user-supplied information are also often uncertain.

An expert system is typically made up of at least three parts: an inference engine, a knowledge base, and a working memory. The inference engine uses the domain knowledge together with acquired information about a problem to provide an expert solution. The knowledge base contains the expert domain knowledge for use in problem solving, very often in form of explicit facts and IF-THEN rules.

A fuzzy expert system is an expert system that uses a collection of fuzzy membership functions and rules to reason about data. The rules in a fuzzy expert system are usually of a form similar to the following:

IF heat *is* low AND pressure *is* high THEN valve *is* closed

where 'heat' and 'pressure' are input variables, i.e. names for known data values, 'valve' is an output variable, i.e. a name for a data value to be computed, 'low' is a linguistic term with an associated fuzzy membership function, i.e. a fuzzy subset defined on 'heat', 'high' is a linguistic term defined on 'pressure', and 'closed' is a linguistic term defined on 'valve'. The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each output variable. The set of rules in a fuzzy expert system is known as the rulebase or knowledge base.

The general inference process proceeds in three (or four) steps.

1. In the fuzzification step, the linguistic terms defined through their associated fuzzy membership functions are matched with the actual values of the input variables, to determine the degree of truth for each rule's premise.

2. In the inference step, the truth values for the premises are propagated to the conclusion part of each rule. This results for each rule in one fuzzy subset that is assigned to an output variable. Usually, only *minimum* or *product* are used as inference methods. In *minimum* inferencing, the output membership function is clipped off at the height corresponding to the rule premise's computed degree of truth. In *product* inferencing, the output membership function is scaled by the rule premise's computed degree of truth.

3. In the composition step, all fuzzy subsets assigned to output variables are combined to form a single fuzzy subset for each output variable. Again, usually *maximum* or *sum* are used. In *maximum* composition, the combined output fuzzy subset is constructed by taking the pointwise maximum over all of the fuzzy subsets assigned to the output variable by the inference rule. In *sum* composition, the combined output fuzzy subset is constructed by taking the pointwise sum over all of the fuzzy subsets assigned to the output variable by the inference rule.

4. The optional defuzzification step is used when it is necessary to convert the output fuzzy set to a crisp number. There are at least 30 different defuzzification methods. Two of the more common techniques are the *centroid* and *maximum* methods. In the *centroid* method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value. In the *maximum* method, one of the variable values at which the fuzzy subset has its maximum truth value is chosen as the crisp value for the output variable.

To cite one of the most prominent and successful fuzzy expert systems, we have to refer to a very long ranging project initiated as early as 1976 by Klaus-Peter Adlassnig and resulting in a system in use today. 'CADIAG-2', which is currently evolving to become 'CADIAG-3', is a medical diagnosis system based on fuzzy expert system technology ([221] contains a recent paper about this very large project which has resulted in a huge amount of publications). A typical rule of this system looks as follows (the rule has been slightly simplified for this example):

IF	fever	*is*	frequent	AND
	high fever	*is*	frequent	AND
	knee dropsy	*is*	rare	AND
	carditis	*is*	very-rare	AND
	articular pain	*is*	almost-always	AND
	erythema	*is*	frequent	AND
	previous tonsillitis	*is*	very-frequent	AND
	staphylokokkus	*is*	never	AND
	increased AST	*is*	almost-always	
THEN	rheumatic fever	*is*	plausible	

Here, one can see again the two key concepts which play a central role in the use of fuzzy logic in expert systems. The first is that of a linguistic variable such as 'high fever', that is, a variable whose values are terms from a natural or synthetic language, such as 'frequent' or 'rare'. The other is that of a fuzzy IF-THEN rule in which the antecedent and consequent are propositions containing linguistic variables. Linguistic variables granularize the domain of variables. In effect, the use of linguistic variables and fuzzy IF-THEN rules results — through granulation — in soft data compression which exploits the tolerance for imprecision and uncertainty. Of course, the effective membership functions represented by terms such as 'very-rare' have also to be determined and must be known at inference time to the inference engine.

For a detailed account of what expert systems in general and fuzzy expert systems in particular are and how they work, we refer to [212, 358, 461].

To emphasize again in what respect artificial intelligence and fuzzy logic can mutually benefit from each other, we want to point out that all complex systems and machines so far required more than just one basic technology in order to be successful. In a large measure, techniques from artificial intelligence and from fuzzy logic are complementary rather than competitive. We believe that it is possible to fruitfully combine techniques from both fields in many areas. The resulting hybrid systems will be more and more important in the future. Following the line of reasoning given at the begin concerning emergent properties, the synergy effect resulting in this combination is necessary to achieve the ultimate goal of creating machines that act more and more intelligently for the benefit of mankind.

For readers interested in gaining a better understanding of one of the two fields, fuzzy logic and artificial intelligence, we would like to refer to some good introductory texts such as Winston's book on artificial intelligence [434], or, more recently, McNeill and Freiberger's book on fuzzy logic [275]. For those wanting to dig deeper or to answer more elaborate questions, we recommend to consult some of the following texts and media (the list could of course be much longer, but we limit ourselves to the most accessible items):

- The excellent 'Encyclopedia of Artificial Intelligence' edited by Shapiro [358] covers almost all possible subjects related to this field, including numerous articles on fuzzy logic [9, 38, 40, 211, 297, 451, 462].
- The 'Readings in Fuzzy Sets for Intelligent Systems' [119] to rapidly find the most influencing articles published in this field, as well as the 'Selected Papers by L. A. Zadeh' [438].
- The internet news-groups comp.ai and comp.ai.fuzzy, also accessible electronically via various mailing lists and blackboards, including their respective frequently-asked-questions (with answers) lists, which contain pointers to other electronic sources of information such as world-wide-web-servers, pointers to the most important conferences, major journals, scientific societies, research centers, major scientific projects, book-lists, as well as names of persons-to-know and companies related to the respective fields. These news-groups are also forums to discuss all topics related to the two fields, and are equipped with searchable archives extending over several years [192, 214].

For readers searching references covering primarily the intersection of artificial intelligence and fuzzy logic, we have compiled a list of some important textbooks [113, 212, 231, 294, 452] and conference proceedings [180, 221] in the bibliography.

1.2 What is scheduling?

Scheduling has been examined in the operation research literature since the early fifties [69]. It has been defined by Baker [8] as *"the allocation of resources over time to perform a collection of tasks."*

Scheduling is a particularly important function in the field of production and operations management, and thus most relevant terminology derives from this source. Variations in problem types are typically illustrated using the manufacturing domain. For example, a *job* is a term used to designate a single item or batch of items that require processing on the *machines*. The processing of a particular job through a particular machine is called an *operation*.

Usually a job is identified with a deliverable product that has to meet a certain quality. A job may have a release and a due date, the combination of which being often called a delivery date. Associated with each job is a formal specification of the product to be produced. Resources are typically those tools, units, materials, and personnel which are used or consumed in the production process. Associated with each resource is some formal specification of its characteristics and capabilities.

A planner considers the specifications of the jobs and the resources and generates a set of operations called a process plan that produces the desired result with a set of explicit ordering constraints on the operations and a set of resource requirements. Often these process plans are fixed for certain products. In contrast to a job shop, the sequence of operations is fixed in a flow shop.

When several jobs are to be executed together, the composition of their resource requirements implies additional ordering constraints that prohibit simultaneous demands on non sharable resources. A scheduler must satisfy both the explicit ordering constraints imposed by the plans and the implicit ordering constraints derived from the availabilities of the resources. The scheduler has also to consider release dates, expected due dates, setup times, and maintenance intervals.

Thus, the order or sequence in which jobs are processed gives rise to a common problem classification [23]:

- General job shop scheduling — where every job may have a different routing through the machines.
- Flow shop scheduling — where every job has the same routing through the machines.
- Permutation scheduling — where the same job-sequence applies to all the machines.

It should be noted that the second and third classes are really special cases of the general job-shop problem. It is easy to relate these concepts to other areas where scheduling is significant. For example, in the management of a hospital, patients can be viewed as *jobs*, and beds, doctors, nurses, etc., as the *resources* that correspond to machines. In the case of the Hubble space telescope scheduling problem [277], astronomical experiments would be the jobs, and energy as well as the optical and various other instruments the corresponding machines.

Dorn and Froeschl [100] proposed another possible classification of scheduling problems by the methods used to solve them:

- Mathematical-analytical methods developed in the context of operations research can be characterized theoretically sound and yielding optimal solutions. These methods do have their limitations, particularly in the field of modeling nasty real world applications. However, they have also their success stories [464], and have successfully been combined with fuzzy methods to solve scheduling problems. We refer here only to work of Zimmermann [454, 458, 460, 462, 463], but our Bibliography lists several hundreds references relevant to the subject, which should be consulted by interested readers.

- Knowledge based methods try to represent all constraints of a scheduling problem *explicitly*, even if some of them are only vaguely known. The main advantage is that the knowledge-base is separated from the inference engine, and that knowledge can be manipulated on its own. Several techniques of artificial intelligence have been developed to master the knowledge representation issues. These methods have also been combined with fuzzy methods to solve scheduling problems, albeit to a much smaller degree. Section 1.3 will further elaborate on this subject.

The two approaches are often interpreted as antagonistic [464], and there are differences of fundamental nature. For instance, while constraints are used in both fields, artificial intelligence researchers use them in a quite different setting, also exemplified by the meaning of the abbreviation 'LP' common to both fields: in the mathematical-analytical case, it stands for 'linear programming', where constraints are quite simple mathematical inequalities, whereas in the artificial intelligence case, it stands for 'logic programming', where constraints can be used to unify variables, and where constraints represent knowledge that can be updated and manipulated in many ways. The question "Which of these approaches is the better one?" is a hot topic in many discussions. We believe that both approaches, rather, are complementary to each other, and often techniques from one field can inspire new ideas to the other one. Pragmatically, this means that both fields could start tackle the problems together instead of fighting with each other.

1.3 Why fuzzy scheduling?

In manufacturing industry such as steelmaking, the distinction between commercial viability and failure often lies in the ability to control the production process through efficient scheduling. Government as well as industry require practical approaches to a diverse set of complex scheduling and planning problems. While scheduling has been studied in isolation for many years, recent advances in artificial intelligence and operations research indicate a renewed interest in the area [100]. In addition, the scheduling problem is being defined more generally, and work is beginning to consider the closed-loop use of scheduling systems in operational contexts. However, a primary source of difficulty in constructing good schedules stems from the conflicting nature of the objectives.

As with many real life decision-making situations, it is usually not possible to fulfill perfectly all objectives when building new schedules. This applies to class-room schedules, staff-roostering, as well as production schedules in manufacturing. Existing approaches to scheduling have tended to reduce the complexity of the problem by considering only a *small subset* of objectives. In real world situations, it would often be more realistic to find *viable compromises* between the objectives. For many problems, it makes sense to partially satisfy objectives. The satisfaction degree can then be used to evaluate the achieved compromise. In addition, real objectives are often prioritized, therefore it is necessary to weight their satisfaction with importance factors. One especially straightforward way to achieve these two aspects of scheduling problems — to satisfy constraints to a certain degree, and to take into account relative importances — is the modeling of these constraints through fuzzy constraints. Fuzzy constraints are particularly well suited for modeling, since constraints can be written in a format easily understood by human experts, and because they feature a robust behavior which needs almost no tuning to yield reasonable control. In addition, the evaluation of their gradual satisfaction can be very efficiently used to guide heuristic search methods as introduced later in this thesis, in order to find approximate 'good' solutions while at the same time greatly simplifying the complexity of the scheduling problem. Real world descriptions naturally contain vaguely formulated relations, because more details are not known or would anyway not lead to better results as they would be canceled out through uncertain data. These uncertain data values can be well modeled through the use of possibility distributions, which are special interpretations of a part of fuzzy set theory. The combination of fuzzy constraints and possibility distributions is realized through fuzzy scheduling, as exposed in this thesis. Thus, the down-to-earth reason behind our choice of fuzzy logic as a basis for knowledge representation is that it allows straightforward modeling of typical scheduling problems and is perfectly combinable with heuristics that find 'good' solutions in acceptable time.

The respected reader may still ask "Why aren't probabilities used instead?" The reason for choosing fuzzy logic and not probabilities as the fundamental knowledge representation technique for uncertainty and vagueness is that, while probabilities and possibilities (fuzzy logic) express different concepts, they can be used to simulate each other, as has been shown by Kosko [228], and acknowledged by Cheeseman [62] already earlier. So, basically, they are equivalent. Then, why do we prefer fuzzy models? On the one hand, even Cheeseman agrees that modeling with subjective probabilities can be a very tedious task. On the other hand, methods based on fuzzy logic have been accepted very well by users all over the world because of the easiness to model in a human-like way many types of complex relationships, i.e. they capture well the vagueness in such everyday expressions as when describing that for instance a car is running *fast*, possibly requiring an appropriate reaction. While methods based on probabilities do have valid application domains where the underlying physical relation is known to be of stochastic nature, these methods require, depending on the exact formalism that is used, unrealistic assumptions such as the independency of random variables, or the judgemental estimate of a large number of parameters for which no empirical support would be available for many other domains. Therefore, the fuzzy approach seemed much more natural to us, and we have chosen to investigate its potential to help solve scheduling problems as encountered in the real world. It is a bit like, though not as extreme as, programming a computer in machine language versus programming in a high level language more suitable for humans. Both have their advantages, but often it is much simpler to solve a problem in PROLOG than to write an assembler program for it.

Other non-standard logics are not further considered because fuzzy logic seems to be the best fit for the general real world scheduling problem in terms of easiness to model the inherent properties of the problem description and the easiness to combine it with available heuristics.

1.4 What's new in this thesis?

- A new combination of fuzzy constraints and repair based heuristics that fit together very well:

  The fuzzy constraints (Chapter 5) help model scheduling problems in better ways then previous models, since they allow a better representation of importance of constraints, and a better representation of how far compromises should go. Repair based heuristics (Chapters 3 and 6) have a much better efficiency to solve typical *large* scheduling problems compared to constructive or enumerative algorithms. In particular, they need no explicit constraint relaxation to still be able to implicitly assess trade-offs between conflicting constraints when the latter are modeled using the mentioned fuzzy constraints. Further, these repair based heuristics do not need to prune search space to still yield very good benchmark results. Indeed, almost all other fuzzy constraint satisfaction algorithms found in the literature rely on search space pruning to achieve better performance, but often explicitly do not look at possibly better compromise solutions (in particular methods that prune all paths where α-cuts fall below a certain level), implying that a solution featuring an unimportant subconstraint with very low satisfaction but constituting nevertheless a global optimum because of the other, more important constraints being satisfied to a higher degree than other instantiations, could be neglected forever. In this sense, the method proposed in this thesis could be seen as an — albeit not 100% perfect — solution to the question raised by Zimmermann [461, p. 371] whether fuzzy set theory can solve large and complex problems computationally efficiently.

- A simplified mathematical method to elicit domain knowledge concerning the importance of constraints:

  It is a major concern in decision making problems how to correctly elicit knowledge from human experts. Section 5.7 simplifies the mathematics of the method developed by Saaty [334] as given by Ibrahim and Ayyub in [193] for practical usability.

- A new consistency test for configuration changes:

  Especially when many human expert have to agree on a problem description such as the rules involved, the importances of certain criteria, etc., it is important to have a method that allows to make reasonable and consistent changes to the parameters of the problem description. In Section 5.8, we present a new test that highlights all inconsistencies in configuration changes. It also provides a possible way to allow automatic learning of problem descriptions.

- New software for the implementation of our ideas (Chapters 5 and 6):

  - FLIP++: a fuzzy logic inference processor library.
  - ConFLIP++: a fuzzy constraint library.
  - DynaFLIP++: a dynamic constraint generation library.
  - InterFLIP++: a common user interface for the other parts.
  - DéjàVu: a heuristic repair library usable for scheduling applications.

- A comprehensive collection of references in the field of fuzzy scheduling (start on page 130), partly enriched with annotations (start on page 105), as well as a fresh look at research issues and challenges in fuzzy scheduling in general (Chapter 2).

1.5 Organization of the thesis

Since you read so far, you have probably already seen the Table of Contents starting on page 4. For the sake of clarity, the following list contains short descriptions of the contents of the thesis's Chapters.

- This introductory Chapter covers fuzzy logic in artificial intelligence in general and motivates the choice of the thesis's subject, fuzzy scheduling.

- Chapter 2 discusses research issues and challenges in fuzzy scheduling systems. It presents various approaches to fuzzy scheduling and to related fields, compares their pros and cons, and discusses some hot research topics.

- In Chapter 3, we present a steelmaking scheduling problem taken from the real world, and investigate how schedules can be generated using fuzzy methods.

- The following Chapter 4 presents a fuzzy expert system that predicts maintenance intervals for a continuous caster unit in a steel plant. This is a partial task required to show how possibility distributions in data can be accommodated in fuzzy scheduling.

- In Chapter 5, we explain in theory and by detailed examples fuzzy set based constraints that help to model general multiple criteria optimization problems. We simplify the mathematics needed for a method of eliciting the criteria's importances from human experts. We introduce a new consistency test for configuration changes. This test also helps to evaluate the sensitivity to configuration changes. We describe the implementation of these concepts in our fuzzy constraint library ConFLIP++ based on our fuzzy logic inference processor library FLIP++, and in our dynamic constraint generation library DynaFLIP++ based on ConFLIP++.

- In Chapter 6, the methods introduced in the previous chapter are extended by iterative improvement repair based heuristics needed to deal with complex real world multiple criteria optimization problems, similar to the one described already in Chapter 3. Here, we describe the more mature implementation of these concepts in our heuristic repair library DéjàVu which uses the DynaFLIP++ and ConFLIP++ libraries introduced in Chapter 5. The benchmark application to compare our fuzzy constraint iterative improvement repair heuristic with constructive methods based on classic constraints is a scheduling system of a continuous caster unit in a steel plant.

- Finally, in Chapter 7, we draw general conclusions about the achieved results and present interesting topics and open problems for future research.

- One purpose of this thesis is to show the state of the art in the field of fuzzy scheduling. Consequently, it contains a Bibliography with a large amount of publications dealing with fuzzy scheduling and related areas. The references are partly annotated in a separate section preceding the Bibliography section. We owe many references to the helpful persons listed in the Acknowledgements section on page 154. Since this thesis is — at least to our current knowledge — the first comprehensive look at fuzzy scheduling, we included these references as a service to others willing to use them for further investigations into this interesting field of research. An ASCII-version of the bibliography in BibTeX-format is located online at

URL: “ftp://mira.dbai.tuwien.ac.at/pub/slany/fuzzy-scheduling.bib.Z”.
We will be happy to insert any updates sent to us through electronic mail to wsi@vexpert.dbai.tuwien.ac.at.

- A doctoral dissertation serves first of all its academic raison-d'être. Therefore, its organization has sometimes to deviate from those of mainstream publications. A Curriculum Vitae was required to be included as the last part of the thesis. Looking at some predecessors' work, we decided to make it a worthwhile part to read. While everything written there is true, it should be taken *cum grano salis* since the author could not resist to brag more than required by university law. If you bother to take a close look at the author's face on page 155, you will understand how it was meant ... and since we didn't spare quotations in this thesis, let's add one more by the great American thinker Noam Chomsky, himself quoting Cato:

  “Ceterum censeo: don't believe the hype.”

Chapter 2

Research issues and challenges in fuzzy scheduling

Life is what happens to you
while you're busy making other plans.

John Lennon

To estimate the time it takes to do a task: estimate the time you think it should take, multiply by 2, and change the unit of measure to the next highest unit. Thus we allocate 2 days for a one-hour task.

Westheimer's Rule, Murphy's Law Complete

This Chapter discusses general aspects of fuzzy scheduling systems. It presents various approaches to fuzzy scheduling and to related fields, compares their pros and cons, and discusses some hot research topics.

2.1 Motivation and complexity issues

Scheduling is a hard problem both in theory and practice. Theoretical scheduling problems, which are concerned with searching for optimal schedules subject to a limited number of constraints, suffer from excessive combinatorial complexity: The general job-shop scheduling problem with n jobs and m machines has an infinite number of of feasible solutions since idle times between operations can vary. A *semi-active* schedule [69] minimizes idle times, but the number of possible solutions is still $(n!)^m$. Simply put, the number of feasible schedules grows exponentially along each dimension (machines, tools, orders, etc.). Many of the most commonly encountered scheduling problems have been proven to belong to the NP hard problems. This implies that these problems are 'at least as hard' as NP complete problems. According to Garey and Johnson [152], NP complete problems are known to be

> *"... 'just as hard' as a large number of other problems that are widely recognized as being difficult and that have been confounding the experts for years.*
> *... the knowledge that [a problem] is NP complete does provide valuable information about what lines of approach have the potential of being most productive. Certainly the search for an efficient, exact algorithm should be accorded low priority."*

The main result is that an exact and efficient solution for NP hard problems has eluded many researchers until now. These problems have therefore been termed intractable. It is however possible to relax one of two criteria, either exactness or efficiency, in which case the other criterion can be fulfilled in many cases. One suggestion could be to relax the problem somewhat in its unimportant characteristics, i.e. to model a simplified version that does yield an acceptable solution efficiently. This is often sufficient for real world scheduling situations. Another suggestion is indicated by the second sentence in above quotation, which is worth some more investigation. In particular, it is interesting to know that NP complete problems can be solved in polynomial time by a nondeterministic computer. The scenario is often such that a solution has to be 'guessed', for instance by consulting an 'oracle', followed by calling a polynomial algorithm to check whether the guessed solution is correct. This would suggest that an algorithm that intelligently 'guesses' a complete instantiation and then checks whether it is a solution could be used to construct an algorithm that finds a 'reasonably good' solution for 'almost all' problems.

The following paragraph provides a little more background about the introduced notions. It is an open problem of complexity theory whether NP is equal to P, P being the problems solvable in polynomial time, i.e. the tractable problems. However, most researchers believe that P and NP are different. This would mean that NP complete problems would remain, at least in the worst case, intractable, i.e. their execution time grows more than polynomially when the structural parameters grow linearly. In such a case, doubling the speed of the computer does not really help since only negligible larger problems will be solvable by that computer, which is usually by far not enough. The NP complete problems are characterized by the fact that they are NP problems and that *all* other NP problems can be polynomially reduced to these NP complete problems. This means that NP complete problems are at least as difficult as any other NP problem. To prove that a problem A is NP complete, it is sufficient to show that A belongs to NP and that one other problem B known to be NP complete can be polynomially transformed into A. A general search problem H belongs to the NP hard problems if and only if there exists a *polynomial time algorithm* for some decision problem C known to be NP complete, assuming that H could be used arbitrarily often for further computations at a computational cost of one unit-time interval by the polynomial time algorithm solving C (i.e., C must be polynomial time Turing-transformable into H).

Practical scheduling problems, although more highly constrained, are complex due to the number and variety of the constraints themselves, many of which are 'soft' e.g. potentially relaxable human preference constraints, rather than 'hard' physical constraints. In addition, a 'good' schedule often needs to be evaluated against a number of potentially conflicting goals which themselves may not be precisely defined. Use of analytic techniques to solve practical scheduling problems has in the past been limited due to the lack of suitably expressive languages for constraint and goal representation. The main application area of fuzzy set theory to scheduling is in the systematic framework it provides for the representation, application, propagation and relaxation of imprecise constraints, and the evaluation of schedules with respect to vaguely defined goals. Thus, fuzzy scheduling may essentially be regarded as a class of fuzzy multiple criteria optimization problems (see Chapters 5 and 6) in which symmetry exists between goals and constraints, essentially both being treated in exactly the same way. In this connection, the result of the 'optimization' must not necessarily be the globally optimal solution to the problem. Instead, optimization refers here pragmatically to the search for the best solution that can be found using all the available resources such as *available* computers, *available* time, and *available* algorithms. The task is to search for a schedule which has the maximum degree of satisfaction of the specified goals and constraints, both of which may be subject to imprecision. Because of the symmetry between goals and constraints in fuzzy scheduling problems, we will henceforth use the term constraint to cover both, as well as for all other overloaded terms for side-conditions, such as 'criteria', 'objectives', 'aims', 'aspiration levels', 'domains' of variables, or 'importances' of certain objects. On a conceptual level, there is a difference between these notions. For instance, 'criteria' more or less specify what a solution *must* look like, while 'aspirations' specify what a solution *should* look like. However, in an engineering context all these 'side-conditions' are usually formulated in one and the same framework, namely by overloading the term 'constraint' with all these notions. In an operations research context, Zimmermann [462] does not distinguish between 'constraints' and 'objectives', arguing that it empirically models the behavior of decision makers quite well.

Most current approaches to automated scheduling organize problem solving into two components, a decision making component responsible for choosing amongst scheduling decisions in order to reach an acceptable schedule, and a constraint management component which uses deductive constraint propagation techniques to combine pre-defined constraints with new constraints introduced by the decision making component to determine new constraints on the decisions remaining to be made. The main impact of introducing fuzzy set theory into scheduling would appear to be in terms of its implications for the constraint management system, and this is indeed where the main thrust of research has been.

2.2 Types of imprecision in scheduling

In scheduling, there are three main types of imprecise constraint that are suitable for being handled with fuzzy sets. Imprecision stemming from constraints that are blurred in definition includes vaguely defined release and due dates, approximately specified constraints on elapsed times between successive operations, desirable values of WIP[1] etc. Such constraints are often expressed by human schedulers using linguistic variables definable operationally by mapping the preferred range of values of the (usually continuous) parameters on to corresponding values of a membership function of the fuzzy set representing that constraint. The mappings may represent the subjective preferences of a human scheduler, or may be derived from a hierarchical constraint management system to reflect current preferences between possibly conflicting strategic organizational goals.

A closely related class of imprecision arises from ill defined preferences between a limited number of disjunctive alternatives that are themselves crisp constraints - a common example being the relative ordering of two sequential operations. Preferences for different alternatives may again be mapped on to fuzzy sets. However, it may be that there are several different situational arguments both for and against each alternative which may have varying degrees of both match and importance with respect to the current situation. A further application of fuzzy set theory is to the combination of different and possibly antagonistic arguments which may be expressed by weighted fuzzy rules whose conditional parts are fuzzily matched to the current situation and which give varying degrees of support to one or other of the alternatives. The OPAL system [18] provides a means of combining the preferences or 'pieces of advice' inherent in such rules using a weighted voting scheme in which advice for alternative decisions is accumulated from different rules through the weighted cardinality of the fuzzy set. A similar method is used by Dorn et al. [90]. In Section 5.5 of this thesis, we present a mathematically correct way to aggregate several fuzzy constraints as defined in Section 5.4. In particular, the presented method allows to compromise between antagonistic constraints in a well-defined way. Section 5.6 develops the idea to allow constraints of different priorities and shows that these different 'weighting schemes' do make sense only in combination with a certain method to aggregate the satisfaction degrees of the fuzzy constraints.

[1] WIP: work in progress.

How constraints and their weights can be optimized for a certain problem is then explained in Sections 5.7 and 5.8.

The third type of imprecision stems from uncertainty about the values of crisply defined scheduling parameters such as process times, material arrival times etc. The constraint management system should be capable in such circumstances of propagating the uncertainty to dependent events so that it accumulates in such a way that reflects how knowledge of event timings becomes increasingly imprecise as the number of uncertain dependencies increases. This would potentially allow a 'graceful' degradation of precision as the schedule extends into the future. It can also allow future events whose timings are precisely known, e.g. scheduled maintenance periods, to act as 'islands of certainty' from which to plan. Various methods have been proposed for propagating temporal uncertainty including probabilistic approaches [335, 22], and interval logic, e.g. [2, 98]. However fuzzy set theory provides an alternative and convenient framework for handling this if temporal parameters are expressed in terms of fuzzy numbers, and constraints are propagated according to the rules of fuzzy arithmetic. Tests for satisfaction of temporal constraints expressed as temporal inequalities will involve comparisons between fuzzy numbers, so any schedule constructed with fuzzy arithmetic will satisfy each individual temporal constraint, including crisp constraints, to some degree between 0 and 1. This again leads to schedules in which degrees of constraint satisfaction are possible. An important area of research thus lies in the development of temporal constraint management systems based on fuzzy arithmetic, exemplified by the work of Kerr and Walker [215] and Dubois and Prade [115].

Systems for temporal constraint propagation ensure that whenever an operation is scheduled, temporal constraints elsewhere in the system are modified accordingly. A major advantage of fuzzy temporal constraint propagation is in its potential for 'decoupling' different regions of the schedule which, because of imprecision in the parameters, may be regarded as non-interacting. This is a very important issue in reactive scheduling in which the effects of (usually frequent) unexpected events and occurrences can be localized to that part of the schedule where it is reasonably certain they will have a dominant impact, rather than being propagated out to the limits of the known horizon where in reality they would be swamped by uncertainties arriving from other sources. A complementary application lies in use of this approach to protect schedules against the effects of uncertainty as has been investigated by Chiang and Fox [67] for hedging against machine failures.

2.3 Fuzzy schedule construction

In the general case, the construction of a fuzzy schedule will involve:

1. establishing a knowledge base of empirical mapping functions, fuzzy rules and weighting factors from which mapping functions can be derived which express the decision makers preferences and state of knowledge and from which the degree of constraint satisfaction of any given schedule can be computed.

2. searching for a schedule with maximal satisfaction of the fuzzy goals and constraints

The 'desirability' of a particular schedule is given by the degree to which it simultaneously satisfies all the goals and constraints, which may be interpreted as the schedule's degree of membership of the intersection of fuzzy constraint/goal sets.

In fuzzy schedule construction, the combination of vaguely defined goals and constraints, coupled with lack of precise predictive knowledge of the extent to which these will be satisfied by any given schedule, implies there will be a much larger class of schedules about which the decision maker is 'indifferent' or which are indistinguishable within the accuracy of the time estimates, than would exist in the crisp case. This can drastically cut down the size of the search space, and a current challenge in fuzzy scheduling is to exploit the representation it provides of varying relative degrees of precision to concentrate search effort in regions of maximum certainty, and conversely, to avoid becoming 'bogged down' seeking improvements in parts of the schedule where the decision maker is indifferent, or adequate predictive knowledge is lacking.

Approaches to schedule construction tend to rely heavily on finding appropriate means of problem decomposition, e.g. using hierarchical approaches, distinguishing between resource-based and order-based perspectives, or the identification and scheduling of critical activities. Fuzzy scheduling does not so much represent an alternative to these but rather a means of enriching the way in which the constraints in such systems can be expressed and propagated. Thus any existing approach to scheduling which currently uses crisp parameters could potentially benefit from 'fuzzification'. This could also extend to iterative improvement scheduling techniques such as simulated annealing, genetic algorithms, and neural network techniques, as it was done in Chapter 6 and in [102, 155, 360] for genetic algorithms and other iterative improvement techniques. Only a limited number of approaches have so far been fuzzified. These include include branch and bound (Dubois et al. [118]), schedule generation and repair techniques in which an initial schedule is generated to which 'improvements' are then sought (Dorn et al. [90], Chapter 3), and progressive narrowing of time windows by making sequencing decisions between pairs of jobs (Bensana et al. [18], Kerr and Walker [215]).

2.4 Research challenges in fuzzy scheduling

One of the most important research challenges in fuzzy scheduling lies in finding appropriate ways to operationalize the intersection relation between different fuzzy constraints, by which the degree of constraint satisfaction of the overall schedule is established. Simply to take the minimum value of the membership functions of the individual constraints as is conventional in classical fuzzy set theory allows no possibility of trade-off between constraints, which in reality may be a significant feature of the decision space. One of the advantages of fuzzy set theory is the relative flexibility available in operational definitions subject to certain key axioms, and of course to the fact that the schedule will equate to the corresponding crisp case when membership functions are restricted to $\{0, 1\}$. Most systems developed so far recognize the difference between 'hard' and 'soft' constraints, and it is usually between the 'soft' or potentially relaxable constraints that trade-offs can occur. In Chapter 5 we have developed a different approach in which, basically, every constraint is considered soft, with the important addition that all instantiations to variables getting evaluation scores above zero are considered as possibly *satisfying* the constraint to that degree, whereas all instantiation to variables evaluating to zero *absolutely violate* the constraint. This approach allows us to specify 'hard barriers' that should never be crossed when relaxing a soft constraint. The concept of these hard barriers has emerged from the actual need to specify hard limits to ranges for certain parameters, up to which trade-offs can be allowed, while still being able to grade the possible instantiations. Soft constraints *without* hard barriers can easily be specified, too. The membership functions of such a soft constraint without hard barriers must simply be defined such that it never reachs zero, though it can approach zero up to any coefficient $\varepsilon > 0$, always indicating that the respective instantiation is inferior compared to others with larger evaluation scores. This definition of soft constraints with hard barriers allows compensation of *partially satisfied* constraints by other constraints being satisfied to a higher degree, while *violated* constraints cannot be counterbalanced by the satisfaction of other constraints. Therefore, it is in accordance with the remarks by Dubois et al. [120] about what can be correctly termed a constraint satisfaction problem. For examples and applications, please refer to Chapter 5. The hard barrier should never be crossed when constraint relaxation occurs. Thus a violated constraint will not propagate through to the evaluation function for a complete instantiation of the problem. A hard constraint can be viewed as a special case that requires no further attention. It is necessary to devise a means of combining individual soft constraint membership functions in such a way that important trade-off preferences between constraints are not lost. The use of weighting factors, as in conventional multiple criteria optimization, is one possible approach and is explored in Chapter 3, but this method basically resumes to a weighted averaging which cannot be combined with other aggregation operators such as for instance t-norms, and where even absolutely *violated constraints* can be compensated unless complicated countermeasures are taken. This method is therefore not in accordance with what can be correctly termed a constraint satisfaction problem as specified by Dubois et al. [120]. In Chapter 5, we investigate weighting schemes having mathematically attractive properties and allowing their combination with more and-like aggregation operators. Indeed, a crisp decision making problem requires a solution to satisfy all constraints, i.e. corresponding to a conjunction of all constraints into one logical formula. In many real world problems, trade-offs are allowed, thus pure and-like operators (the t-norms presented in Section 5.5) are not really the best choice for most applications. We present in Section 5.5 soft-and operators embracing most of the nice properties requested to model real world problems, such as the hard barrier feature, the capability to allow compromises within the hard barriers of constraints, symmetry with regard to the arguments, and allowing an intuitive but mathematically correct way to specify weights to prioritize constraints among each other.

A closely related problem is in the establishment of systematic means of logically combining different arguments for and against the satisfaction of individual crisp constraints, as in Dubois et al. [118], and choosing empirical values for subjective weights, preferences and degrees of knowledge, which when combined according to the chosen framework, adequately reflect the cognitive processes of a rational human decision maker. Clearly, in any realistic system, a degree of 'tuning' would be required both of the subjective preferences and uncertainty values, and of the system for combining them. This is a difficult and contentious issue which applies equally to the field of decision analysis where it has received considerable attention in the literature. As an answer to this problem of finding an adequate 'configuration', which would also embrace such aspects as to find adequate models for the sets of fuzzy linguistic terms and variables with their associated membership functions, as well as weights of constraints and even appropriate aggregation operators, we have proposed in Sections 5.7 a method to elicit the priorities for constraints from human experts through an intuitively understandable relative scale of priorities which can be mathematically transformed through matrix calculations into a ratio scale usable in the weighting scheme proposed in Section 5.6. Section 5.8 develops a complementary test of general configuration changes that checks in a very pragmatic way whether changes in any dimension of the problem description, such as weights or membership functions, are consistent with a set of previously remembered reference decisions. The test is very general in nature, and could therefore be applied to any decision making problem where configuration changes have to be done from time to time. Reasons to modify the configuration could be the introduction of new machines, or simply individual preferences of different human experts. The consistency test is used such that, if the human experts are dissatisfied with a ranking produced by the optimizing system, they can slightly change the weights of some constraints, or the exact form of some membership function (e.g. to specify that the hard barrier is actually located slightly higher), or any other parameter of the problem, such as the aggregation operator used. The consistency test will then check whether the new configuration is consistent with the rankings for a set of reference pairs of instantiations. This is done by applying the new configuration, e.g. the set of new weights, to all the remembered ordered pairs of instantiations, and calculating their evaluation scores with this new configuration. If for each reference pair the order between the two reference instantiations remains unchanged, this indicates that the new configuration does not invalidate any previous reference ordering. It is compatible with all decisions made in the past that became reference ranking pairs. Through this method, it is possible to lead *several* human experts to agree on a common, undisputed subset of some reference ranking pairs of instantiations, or at least to establish several different sets that correspond to configurations which can be further characterized by (and saved for later reuse under) such names as 'risky/cost-cutting', 'highest-quality', 'observe-temporal-constraints', 'standard-mix', etc., indicating their general tendency for decision making. Whether this consistency test can be further developed to allow automatic learning of problem descriptions is a question open for future investigations.

The development of search strategies which are explicitly designed to exploit the characteristics of fuzzy constraint representation is also an area that requires attention. As outlined above, this representation can assist in pruning the search space and in focusing search in more profitable areas where preferences are stronger and predictive accuracy expected to be higher. Care must however be taken not to prune away the optimum. Indeed, we found that almost all fuzzy constraint satisfaction algorithms found in the literature rely on search space pruning to achieve better performance, but often explicitly do not look at possibly better compromise solutions (in particular methods that prune all paths where α-cuts fall below a certain level), implying that a solution featuring an unimportant subconstraint with very low satisfaction but constituting nevertheless a global optimum because of the other, more important constraints being satisfied to a higher degree than other instantiations, could be neglected forever. This danger is not present when combining fuzzy constraints with repair based or iterative heuristics as proposed in Chapters 3 and 6, since candidates are selected stochastically. No part of the search space has to be pruned, no constraints need to be explicitly relaxed to achieve efficiency and still find solutions near the optimum.

New forms of problem decomposition might be envisaged which focus for example on individual parts of the schedule surrounding 'islands of certainty', between which only weak coupling exists. This is an important consideration in the context of reactive scheduling and the construction of schedules which are robust with respect to unexpected contingencies.

Another potentially very interesting approach to fuzzy constraints used for scheduling has so far attracted only very limited attention. We refer to unification based constraint satisfaction with fuzzy constraints as proposed by Matyska [273]. The approach is still limited, since only finite fuzzy constraints (no continuous membership functions) are supported, because unification on continuous domains is a conceptually difficult problem. So far, only toy problems can therefore be handled by this approach, but it is a line of thought that deserves further investigation as fuzzy unification could provide a powerful means to solve combinatorial problems with vague descriptions. Complexity reduction will of course be an important issue, and it might well be that iterative repair based methods again show a valid path to solve fuzzy unification problems.

Finally, an important issue in fuzzy scheduling lies in the problem of quantifying the benefits of the approach. Although the performance of fuzzy and non-fuzzy schedules may easily be compared in terms of the time taken to compute an 'acceptable' schedule, it is more difficult to compare the actual schedules produced without either field trials or controlled simulation experiments containing parameters and events whose probability distributions are unknown to the scheduler. So far there appears to be a dearth of studies which address this issue.

Chapter 3

Uncertainty management in production process scheduling

鉄は熱いうちに打て！

Japanese proverb

Measure with a micrometer.
Mark with chalk.
Cut with an axe.

Ray's Rule for Precision, Murphy's Law Complete

In this Chapter, we present a steelmaking scheduling problem taken from the real world and investigate how schedules can be generated using fuzzy methods. The steelmaking scheduling problem differs from the one that will be discussed in Chapter 6, and constituted our first attempt to combine fuzzy constraint techniques with repair based heuristics. Chapter 5 will go deeper into details concerning methods for knowledge representation using fuzzy constraints, therefore we limit ourselves in the present Chapter to the description of a typical steelmaking scheduling problem and give only general hints about where and how fuzzy knowledge representation techniques can be applied.

3.1 Introduction

We present a scheduling methodology where the generation of schedules is constrained by antagonistic and vague knowledge. Besides temporal and capacity constraints, compatibility constraints between consecutive jobs are managed. We model the vague constraints and uncertain data by fuzzy set theory. The importance of jobs is defined based on the different constraints. A preliminary schedule is generated by first considering the jobs that are important and difficult to schedule. Easy or not so important jobs are scheduled later. The initial schedule is 'repaired' through various intermediate steps until a schedule that attains a given level of satisfaction is found. The 'goodness' of solutions is rated through the use of methods based on fuzzy sets. As a side effect and through careful modeling of the domain constraints, schedules that are robust with respect to small changes in actual production parameters will be preferred because they get higher satisfaction degrees compared to weaker but otherwise equal schedules. However, if no robust solution is found, weaker ones will get their chance. This methodology is appropriate for applications in process engineering where uncertain knowledge is dominant. We explain the methodology with a case study from a steelmaking plant for high grade steel.

In the application described in this Chapter, approximately 45 jobs have to be sequenced for one production line. In a mathematical model with no help of heuristics, the scheduler has to check 45! or more than $1.19622211 \times 10^{56}$ possible sequences of jobs for constraint violations. Complexity will be even higher if conflicts between production lines are examined. If arbitrary idle times between operations are allowed, the solution space that will have to be scanned for the optimal solution will even be infinite. The methodology described in this Chapter manages this complexity by applying heuristics that the human experts use too.

Uncertainty and vagueness are difficult problems in the domain. The durations of operations are only approximately known, and constraints are often specified vaguely. We solve this problem by applying qualitative reasoning based on fuzzy logic.

Classical models assume a very idealistic view of scheduling problems. For example, for a number of machines M and a number of jobs N with a goal function 'minimize makespan', a solution is computed. However, in most realistic domains the devil is in the 'nuts and bolts'. If an additional constraint must be satisfied, e.g., machine m_1 should not operate simultaneously with machine m_2, a new model must be developed. In contrast, an additional rule or constraint is added very easily in a knowledge based model.

3.2 Description of the steelmaking process

In a joint project between the Alcatel Austria-Elin Research Center and the Christian Doppler Laboratory for Expert Systems, an expert system was developed for the Böhler company in Styria, Austria, one of the most important European producers of high grade steel. The system supports the technical staff in the steelmaking plant in generating schedules of steel heats for one week [89]. The system was implemented with Pamela, a rule based system developed by the Alcatel Austria-Elin Research Center [10]. Although this system works well, we generalize the problem in this Chapter and set the applied method on a firmer ground by making it robust with respect to the influence of uncertainty and vagueness.

The Böhler company is divided into several plants. The steelmaking plant is the first in the production process. The produced steel is delivered to subsequent plants such as the rolling mill or the forges. The steelmaking plant receives orders from these plants to produce slabs or ingots of a certain quantity and quality. The destination is important for the scheduling process, because the working hours of these plants must be considered. Sometimes products are stored for several days in intermediate stock yards, because the next plant cannot process the jobs in the same sequence as the steelmaking plant. The differing sequencing criteria of jobs cause considerable costs for the company. Moreover, since the steel cools down it must be warmed up again in the next plant. To reduce costs and to improve quality, some orders have delivery dates.

Wednesday morning, engineers of the different plants meet to discuss orders for the next week. Compatible orders that may have different destinations are used to form jobs. A list of jobs for one week is worked out manually for each production line of the steelmaking plant. Usually, the first shift in the steelmaking plant starts Sunday evening, and the last shift ends Saturday. Sometimes fixed sequences of two or three jobs are specified in order to facilitate scheduling in the next plant. The task of the scheduler is to find a possible temporal assignment for all jobs while violating as few compatibility constraints between jobs as possible, and to allocate resources over time without violating temporal and capacity constraints. The result of this scheduling process may be that some orders are rejected and shifted to the next week. To reduce the number of rejected orders, general rules that define which combinations of orders can be produced in one week are given to the subsequent plants. Nevertheless, these constraining rules may be violated to produce important orders.

Pig iron produced in blast furnaces contains usually more than 4% carbon and is therefore brittle. To get a deformable product, carbon is reduced down to 2% in pig iron, to produce steel. For many high tech products the quality of steel must be even higher. High grade steel is crude steel refined by adding alloying metals like manganese, tungsten, chromium, and others. These alloying metals increase aspects like compression strength, impact strength, and many more. High grade steels are for example stainless steel, high speed steel, tool steel, and structural steel. To reduce material costs, scrap iron with high percentages in the desired alloying metals is used to obtain high grade steel.

The steelmaking plant in Kapfenberg consists of three main production lines that share some aggregates. For every steel quality there is a process plan that describes which operations must be performed on which aggregates to produce the specified quality. These operations and sequences must be replanned only when a failure occurs.

The steelmaking process starts with the charge of crude steel and scrap iron in one of the electric arc furnaces (eaf). The filling of one furnace is called a heat and already contains the main alloying elements. The furnaces have different capacities, from 17t to 55t. The duration of the melting process depends on the ingredients and on external factors. Since the furnaces consume a lot of electric energy, they have sometimes to be switched off due to voltage peaks. Up to five hours can be required for one melting, but usually two to two and a half hour are enough. A fixed setup and maintenance time of altogether twenty minutes is included in this interval.

The melted steel is poured into ladles that are transported by a crane to a ladle furnace (LF). If the preceding heat has a long processing time in the ladle furnace, the current heat must wait. This slack time may not exceed two hours. The next step is a heat treatment in the ladle furnace where the fine alloying takes place. The duration of this heat treatment is usually about the same as the melting time. Later a special treatment may be performed in the vacuum oxygen decarburation (VOD) unit or in the vacuum decarburation (VD) unit. The VD-unit can be converted into a VOD-unit. This conversion takes about three to four hours.

The next step is either the processing of the steel in a two stranded horizontal continuous caster (HCC) to form slabs, or the casting of the steel into moulds to form ingots. The teeming rate for the HCC is about 50t/h. If the casting format has to be altered, a setup time must be taken into calculation, too. For casting ingots, space in one of the four teeming bays (TB) is required, where ingots can solidify in moulds. As a rule of thumb, the solidification time for ingots in hours is half of the weight of the ingots in tons. For example, a big ingot of 52t needs about one day. The storage places for big forging grade ingots (>19t) are limited. On the other hand, the effort to cast *many small* ingots is greater than for a *few large* ingots. About 70% of the jobs are cast into ingots.

The Böhler-Electro-Slag-Topping (BEST)-technology is a special casting technology for big ingots. These ingots are treated additionally in the BEST-unit. Only one place in the teeming bays (TB1) exists for them. All aggregates and the routings for heats are shown in Figure 3.1.

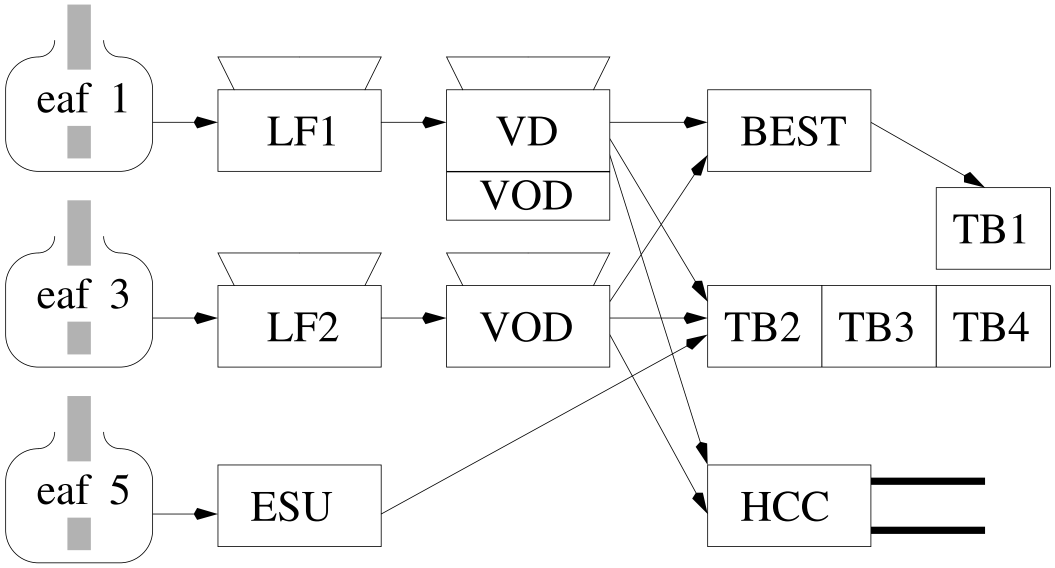


Figure 3.1: Aggregates in the steelmaking plant.

3.3 Constraints in steel production scheduling

During the construction of a schedule several constraints have to be satisfied. These constraints are often vague, and they conflict with each other. The engineer has no pretension to generate an optimal schedule, knowing that the uncertainty in the execution of the plan would break this optimality. The engineer can decide that some schedules are better than others, but cannot give algorithms to construct the optimal schedule. The engineer considers the following constraints:

- Compatibility constraints

  The main problem in scheduling is that residuals of one heat in the electric arc furnace may pollute the next heat. The engineers use as a rule of thumb that 3% of a chemical element in a heat remain in the wall of the electric arc furnace and 3% of the difference of the elements in two consecutive heats will be assimilated by the second heat. Of course, the 3% are always on the safe side, and the expert can sometimes relax this factor.

  > Example: *Assume that a heat h_1 contains 20% nickel and a heat h_2 that is processed next in the furnace should have 4% nickel. The second heat will take approximately $(20-4)\times 3/100 = .48\,\%$ nickel from the wall. When scrap iron is inserted in the furnace, this amount is taken into account, therefore only 3.52% of nickel are added. However, if the second heat has to contain less than .58% of nickel, then*

this sequence of heats is not allowed since there is no way to reduce the amount of nickel in that case.

The rule is effective for 42 chemical elements, but usually only 8 main elements are considered. Due to the diversity of qualities, these constraints often cannot be met. Actually, eight to nine percent of all heats are destroyed. They must be melted again and may be reused for another less critical order. Besides avoiding such destructions, it is expensive to waste rare elements. If one job demands a high percentage of an expensive element like cobalt, the next job should use as much of this residual as possible. Although these compatibility criteria hold for every aggregate in the production process (including ladles), usually only the electric arc furnace constraints are observed.

- Temporal constraints

  Since some steel qualities require the steel to be hot for a subsequent treatment like forging, there will be an appointment between the steelmaking plant and the next plant that must be met within a tolerance of ± 2 hours. The average number of jobs with such delivery dates is about 10%. Of course, these constraints are not really hard since they may be adapted through negotiations with the subsequent plant. However, it is desirable to meet these delivery dates in order to reduce the time needed for renegotiations.

  For some jobs no appointments are made, even though their subsequent treatment should be started immediately after casting. These jobs should not be scheduled at the end of the week because the subsequent plants are usually not working then. Some jobs with difficult treatments should be performed during day shifts so that an engineer can supervise these jobs. The treatments in the aggregates behind the electric arc furnace are usually shorter than the duration of the melting. However, for very high qualities the duration in the ladle furnace is longer. Finally, the scheduler must guarantee that waiting intervals between operations may not exceed a certain limit. An objective for the production is to keep these intervals as small as possible. This results in a minimization of the makespan which will reduce production costs. However, this objective is only a secondary goal. The main objective is to have as few heats as possible that do not meet their quality requirements. Since this objective is difficult to attain perfectly, it is seldom possible to consider the secondary goal.

- Capacity constraints

  If a heat should be cast in many small ingots, the load for the workers that set up and strip off the moulds is larger than for few large ingots, because the handling of every mould takes approximately the same time. The workers do not like to have many heats cast into small ingots during a short time period.

One objective of the scheduler is to achieve a uniform load distribution over the planning horizon for the workers.

The solidification of big BEST-ingots (52t) takes about one day. Since only one slot exists for such big ingots, only one can be produced per day. The space for smaller forging grade ingots is also very limited. Since there is only one continuous casting unit, only one furnace may supply this unit during a certain period. If two subsequent heats of one furnace should both be cast on the continuous caster and have approximately the same steel quality, the caster should operate continuously. These jobs are called serial castings. As few delays as possible should occur between the consecutive jobs. If an amount of steel that does not make a full heat is ordered, it can be combined with another one of compatible quality, forming a double- or triple-casting. This means that only a part of the heat is poured in the refinement ladle. Durations of treatments in the refinement and casting process will be shorter in this case.

3.4 Heuristics used by the experts

The experts of the plant use heuristics to construct schedules. These are used to master the complexity of the construction, but they are not used to evaluate a constructed schedule. If no schedule is found, some constraints are relaxed since it is known that usually there will be enough freedom during execution to correct the violated constraints. Again, this relaxation is controlled by heuristics.

Before the expert system project was started, an attempt was made to schedule the heats with traditional software methods. This project was canceled because the program handled the constraints too rigidly. It was not able to relax constraints. It scheduled a lot of jobs correctly that were easy to schedule, but some of the difficult jobs remained always unscheduled.

An important concept for the scheduling process is the *alloying cycle*. This is a series of heats in which the amount of a chemical element is decreasing. For example, the concentration of nickel could decrease over several heats from 26% to .5%. Several alloying cycles for one element may occur in sequence and several alloying cycles for different elements may overlap in time. Heats should be scheduled in this order. Additionally, the quality that is produced at the end of a week affects the heats in the beginning of the next week. Figure 3.2 illustrates the overlapping alloying cycles. It visualizes the amount of nickel and chrome for the jobs of Schedule 3-1 from Table 3.6 that was generated from the orders of Table 3.1.

One important task driven by heuristics is the recognition of possible alloying cycles. This is supported by the experience of the engineers in the plant: the amount of an element should decrease slowly over a sequence of heats, but can increase very fast. Additionally, the number of peaks in this curve of a saw blade is kept to a minimum if possible.

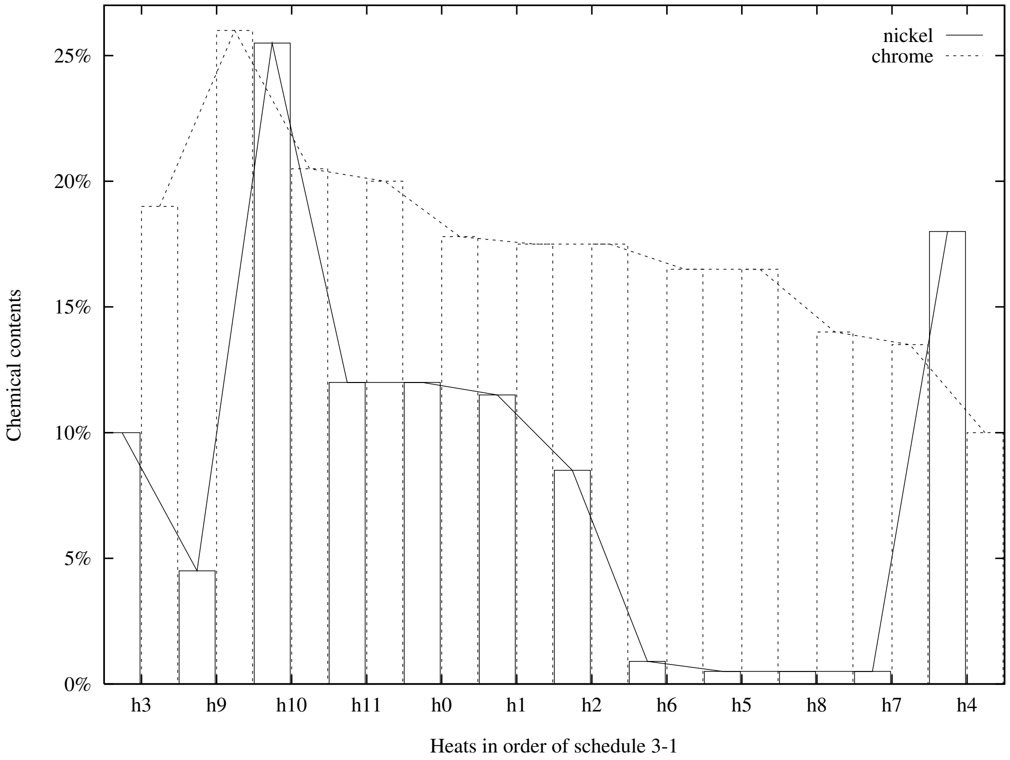


Figure 3.2: Overlapping of alloying cycles. The graph visualizes the amount of nickel and chrome for the jobs of Schedule 3-1 from Table 3.6 that was generated from the orders of Table 3.1.

No.	Name	Time	T	Ni	Cr	Co	Mn	Fe	V	W	Mo	Size
h_0	A101		H	12.0	17.8	.0005	1.80	69	.10	.005	2.800	23/1.6
h_1	A300		K	11.5	17.5	.0005	1.50	67	.10	.005	2.300	3/1.31, 16/2
h_2	A506	morn.	K	8.5	17.5	.0005	2.00	69	.10	.005	.005	15/1, 16/1.6
h_3	A604		C	10.0	19.0	.0005	1.50	69	.10	.005	.005	65-20
h_4	A700		H	18.0	10.0	.0005	1.50	70	.10	.005	.100	1/33
h_5	N310		H	.5	16.5	.0005	1.50	81	.10	.005	.300	12/1.6, 21/1
h_6	N335		H	.9	16.5	.0005	.80	80	.10	.005	1.100	16/1, 15/1.6
h_7	N540		H	.5	13.5	.0005	.50	84	.10	.005	.500	13/3
h_8	N678		A	.5	14.0	.0005	.50	82	2.00	.000	.500	36/1
h_9	H304		H	4.5	26.0	.0005	1.30	65	.10	.005	.050	8/1.6, 24/1
h_{10}	H525		H	25.5	20.5	.0005	1.30	53	.10	.000	.000	27/1, 6/1.6
h_{11}	H550		H	12.0	20.0	.0005	1.30	66	.10	.000	.000	12/1, 16/1.6

Table 3.1: Characteristics of jobs for furnace eaf3.

BEST-ingots introduce some problems. Typically there are groups of such jobs that all have the same chemical quality requirements. They are usually forging grade ingots, should be delivered hot to the forge, and are low alloyed, which means that their amount of alloying metals is very low. From a compatibility point of view they should be produced in sequence. Unfortunately, they solidify slowly, and there is only one place for them in the teeming bay.

Therefore BEST-ingots are scheduled at a rate of only one per day. Since they are low alloyed, the furnaces will probably be high alloyed in the in-between periods. Since they should be delivered hot, they cannot be scheduled before weekends or public holidays. Therefore at weekends the furnaces shall be high alloyed, and as a consequence these orders contain many small ingots.

The HCC-unit is most economical if several heats with the same quality and format are cast continuously without breaks. If such a group of jobs exists, these jobs should be scheduled one after the other. However, all jobs of this group must be in time for their casting. After the processing of some jobs on the caster, a maintenance interval must be scheduled. Additionally, a setup time of some hours must be reserved when a format conversion has to be performed.

Since the durations of the operations in the steelmaking process are uncertain, the engineers prefer to charge the caster only with heats from one furnace. However, they sometimes change furnaces once in a week. If this happens, they decide in advance when this change should take place before starting to schedule single jobs. Further, they accept only one format conversion. The engineers schedule this conversion a priori. Later, each job that requires the HCC-unit can be assigned to one part of the week.

3.5 Schedule construction and repair

Our approach to solve the problems mentioned in the introduction is as follows: for a given planning horizon a preliminary schedule is generated by first considering very important jobs and those that are difficult to perform. To manage the given complexity, the schedule is constructed without chronological backtracking. The importance of jobs is dynamic, which means that the importance of one job may grow over time and depends also on the pool of other jobs to be scheduled.

A preliminary schedule may not contain all jobs and still violate some constraints. In such a case, jobs in the schedule will be exchanged to find a proper schedule. A hill climbing search method controls this exchange.

To compare solutions, the system uses an evaluation function that is based on the given constraints. We use fuzzy set theory to model this evaluation. Motivation for this choice was the ease to formulate knowledge reflecting the complex non linear behavior formulated by the engineers. Additionally, fuzzy sets are well suited to model knowledge containing vague human like formulations. Such formulations can often be heard from human experts explaining their domain.

After introducing a small case from the application, we show how the constraints are represented by fuzzy sets and how an evaluation for a complete schedule can be computed. Then we explain the generation of a preliminary schedule. The system

No.	Name	Time	T	Ni	Cr	Co	Mn	Fe	V	W	Mo	Size
h_0	M100		H	.1	1.2	.0005	1.30	95	.10	.005	.005	34/1, 8/1.65
h_1	M200		H	.1	2.0	.0005	1.60	94	.10	.000	.230	2/24
h_2	M238		B	1.2	2.1	.0050	1.60	90	.10	.005	.250	1/52
h_3	M238		B	1.2	2.1	.0005	1.60	90	.10	.005	.250	1/52
h_4	K460		C	.1	.6	.0005	1.15	94	.15	.600	.005	157-13
h_5	K460		C	.1	.6	.0005	1.15	94	.15	.600	.005	157-13
h_6	K455		H	.1	1.2	.0005	.40	92	.20	2.100	.005	16/1, 18/1.6, 4/2
h_7	K600		H	4.2	1.4	.0005	.50	91	.10	.005	.300	33/1.6
h_8	S600	11 am	F	.2	4.3	.0005	.35	78	1.90	6.700	5.200	50/1
h_9	S600		F	.2	4.3	.0005	.35	78	1.90	6.700	5.200	50/1
h_{10}	W300		H	.2	5.2	.0005	.50	88	.50	.005	1.400	1/24,1/1.1,10/1.6
h_{11}	W302		H	.2	5.2	.0005	.50	89	1.10	.005	1.400	14/1, 17/1.6, 8/1.3

Table 3.2: Characteristics of jobs for furnace eaf1.

iteratively generates schedules. Important jobs are scheduled first, then gaps in the schedule are filled, and finally other jobs are scheduled. The schedules generated in this phase may violate constraints. Additionally, some jobs may exist that were not scheduled due to conflicting constraints.

Therefore, the last phase is a repair phase that searches a better schedule. This approach is similar to those of Minton et al. [276, 277] and Zweben et al. [465]. They have shown empirically that repair based methods perform orders of magnitude better than traditional backtracking techniques. We explain the technique in this Chapter in Section 3.9 on an example, and with more detail including empirical results in Chapter 6.

3.6 Example

We take a small case study from the described application to illustrate the proposed technique. We restrict the case study to two furnaces and the planning horizon to one day. Additionally, we consider only a subset of the given constraints to reduce the complexity of the example. The input for the 'scheduler' are two lists of jobs for the electric arc furnaces eaf1 and eaf3, as given in Tables 3.1 and 3.2.

The name of each job identifies the quality of the steel. The column 'Time' gives the delivery date or the preferred time. The column 'T' (like in 'type') is used to provide further information about the processing: 'C' stands for continuous casting, 'B' for BEST-technology, 'H' for hot delivering, and 'F' for fixed delivery. In the last column the number and size of ingots are given. Each pair represents the number of ingots and the ingot's size in tons. For C-type jobs, the format of the produced slab is given.

3.7 Evaluation of schedules

The knowledge of the application can be put into three main groups: knowledge about a particular job, temporal constraints, and constraints on the compatibility of jobs. All of these are represented by fuzzy values.

Temporal fuzzy values can be used to describe the duration of operations and whether jobs are too early or too late. The fuzzy value describes a degree of uncertainty in both directions. The following linguistic terms[1] can be identified: *very-early, early, in-time, late, very-late.* For the evaluation of a schedule it makes no difference whether jobs are too early or too late. Therefore, the five values are mapped onto three values: *in-time, nearly-in-time,* and *not-in-time.* From these values a schedule can be evaluated with respect to its temporal constraints:

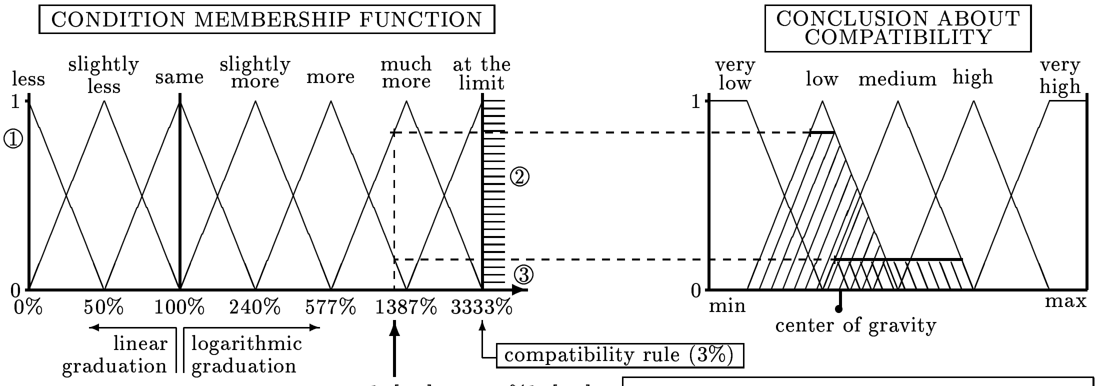
$$\text{timeliness}(S) \stackrel{\text{def}}{=} \bigwedge_{i=1}^{N} \text{timeliness}(H_i) \tag{3.1}$$

The fuzzy *and* operator could be realized by taking the minimum of the arguments, but other more appropriate mathematical models are explained in Section 5.5.

Prade [315], and more recently Bel et al. [12], Dubois [114], Dubois and Prade [115], and Kerr and Walker [215], have successfully employed fuzzy logic to represent temporal constraints for knowledge based scheduling. In their approaches, each crisp interval is preceded and followed by a slack time. For each moment of this slack time, there is an associated fuzzy membership grade defining the uncertainty that the corresponding slack time is correct. Therefore, those systems can cope with small perturbations causing delays. The creation of robust schedules is facilitated, since smaller intervals get higher scores, while being considered identical to larger ones by systems without fuzzy evaluation. Our approach generalizes the other ones to include, besides such temporal constraints, more kinds of constraints, such as like chemical or organizational ones.

The compatibility of two jobs integrates different chemical elements and the work load of workers. The compatibility between two jobs is calculated by first evaluating the compatibility for each factor separately to get restricted compatibility measures. Accordingly, six fuzzy linguistic terms for the global as well as for each restricted compatibility are defined: *very-high, high, medium, low, very-low,* and *no-compatibility.* The latter is a special case, since a sequence being classified incompatible can never be scheduled in this order because of hard chemical constraints. In the lower part of Table 3.3, rules defining this compatibility measure for different factors are listed. The specifications of the ingredients are sometimes upper limits and sometimes nominal values. These rules can be interpreted directly as fuzzy inference rules.

[1]For more details regarding fuzzy linguistic terms, please refer to Section 1.1 and Section 5.10.



① fuzzy number range [0,1].
② region of physical incompatibility.
③ $H_0[E]$ (percentage of element E in H_0), in % of $H_1[E]$.
H_0 is the heat (job) preceding the heat H_1.
E is a chemical element like nickel or cobalt.

Example:
We compute the nickel-compatibility for h_3 preceding h_4, both as specified in Table 3.2. Using the fuzzy inference rules from below, we find that only rule 5 and 6 contribute to the result calculated as drawn above. According to this result, the nickel-compatibility for h_3 preceding h_4 is more *low* than *medium*.

The fuzzy inference rules with linguistic variables and terms:

1. IF the percentage of chemical element E in heat H_0 is *less* than in heat H_1, THEN the E-compatibility of H_0 preceding H_1 is *medium.*
2. IF the percentage of chemical element E in heat H_0 is *slightly-less* than in heat H_1, THEN the E-compatibility of H_0 preceding H_1 is *high.*
3. IF the percentage of chemical element E in heat H_0 is the *same* as in heat H_1, THEN the E-compatibility of H_0 preceding H_1 is *very-high.*
4. IF the percentage of chemical element E in heat H_0 is *slightly-more* than in heat H_1, THEN the E-compatibility of H_0 preceding H_1 is *high.*
5. IF the percentage of chemical element E in heat H_0 is *more* than in heat H_1, THEN the E-compatibility of H_0 preceding H_1 is *medium.*
6. IF the percentage of chemical element E in heat H_0 is *much-more* than in heat H_1, THEN the E-compatibility of H_0 preceding H_1 is *low.*
7. IF the percentage of chemical element E in heat H_0 is *just-below* the physical limit imposed by the element's presence in H_1, THEN the E-compatibility of H_0 preceding H_1 is *very-low.*
8. IF the percentage of chemical element E in heat H_0 is *over* the physical limit imposed by the element's presence in H_1, THEN there is *no-compatibility* for H_0 preceding H_1.

Table 3.3: Fuzzy inference to compute chemical compatibility between two heats.

The calculation of the nickel-compatibility is illustrated in the upper part of Table 3.3. In this case, only the two rules 5 and 6 contribute to the result. The condition parts of the rules contain statements about the percentage of a chemical element in the first heat compared to the following heat. In the example taken from Table 3.2, the heat h_3 must contain $h_3[\mathrm{Ni}] = 1.2\%$ of the chemical element nickel, while heat h_4 should contain only $h_4[\mathrm{Ni}] = .1\%$. The relative percentage of $h_3[\mathrm{Ni}]$ is therefore 1200% of $h_4[\mathrm{Ni}]$. Considering only nickel, the question is whether the sequence h_3 preceding h_4 is allowed or not, and if yes, how good this sequence is compared to other sequences. To decide this with the given fuzzy inference rules, the vague linguistic terms and crisp but uncertain numeric values must be matched. This is done with fuzzy membership functions as defined in Table 3.3, both for the condition and for the conclusion part. Similar graphs representing the membership functions associated with fuzzy inference rules can be found for example in Maki et al. [266] and Kanemoto et al. [213] where the computations are done in a comparable way.

In our example, the numeric input of 1200% relates more or less with the linguistic terms *more* and *much-more*. Following the dotted lines to the conclusion membership functions for rules 5 and 6, two membership functions $low[\mathrm{Ni}](h_3,h_4)$ and $medium[\mathrm{Ni}](h_3,h_4)$ appear as a result of the calculation. Their combination

$$\mathrm{comp}_{[Ni]}(h_3, h_4) \stackrel{\mathrm{def}}{=} low_{[Ni]}(h_3, h_4) \vee medium_{[Ni]}(h_3, h_4) \tag{3.2}$$

is a new membership function defining the nickel-compatibility of h_3 preceding h_4. We obtain as a result that the nickel-compatibility for h_3 preceding h_4 is a possibility distribution more resembling the linguistic term *low* than *medium*. The fuzzy *or* operator could be realized by taking the maximum of the arguments, but other mathematical models more appropriate in many cases can be found in Section 5.5.

The conditions of the fuzzy inference rules consider only relative values for the percentage of elements like nickel in the two compared heats. Absolute values are for the compatibility problem of minor interest, but could easily be modeled by introducing more complex three dimensional membership functions. We chose a half logarithmic graduation to be able to handle the relative values. Since the compatibility rule is asymmetric and only restricts the second heat to a minimal value for a certain chemical element that must at least be present in this second heat, the graduation is asymmetric by being logarithmic only on the right half. Besides simplifying the visualization, this logarithmic scale has an additional positive effect, since positions on the right side of the 100% mark that are still near the center are preferred and get more attention per unit than positions more close to the physical limit on the far right. This reinforces the natural meaning of the fuzzy linguistic terms positively. In addition, it shows how easy non linearities in the domain can be modeled through fuzzy knowledge representation methods. For more details, e.g. regarding priorities between constraints and how to elicit them and fuzzy membership functions from the human expert, refer to Chapter 5.

The fuzzy inference rules like those in Table 3.3 express several fuzzy judgments about the compatibility between heats. These judgments are in the form of membership functions and can be simplified to the linguistic term to which the judgment mainly pertains. The resulting fuzzy values can all be combined by computing a weighted mean of the membership functions for each component to get one overall value for the two heats:

$$\mathrm{comp}(H_i, H_j) \stackrel{\mathrm{def}}{=} \sum_{E \in \{Wl, Ni, Cr, \ldots\}} g(E)\mathrm{comp}_{[E]}(H_i, H_j) \tag{3.3}$$

In this formula, g(E) is the normalized weight of a rule and E is a member of the set of all factors influencing the compatibility, namely work load (Wl) and the 8 chemical elements like nickel or chromium. A more elaborate and correct model to represent and compute weighted aggregation of constraints can be found in Chapter 5 on knowledge representation through fuzzy constraint.

The compatibility can also be defuzzified by calculating the center of gravity of the surface and then taking the value of its x-coordinate as the result. This value can be computed by the following formula:

$$\mathrm{defuzzy}(\mathrm{comp}(Hi, Hj)) \stackrel{\mathrm{def}}{=} \frac{\displaystyle\int_{x=min}^{max} x\,\mathrm{comp}(H_i, H_j)(x)\,dx}{\displaystyle\int_{x=min}^{max} \mathrm{comp}(H_i, H_j)(x)\,dx} \tag{3.4}$$

This computation is done for every pair of jobs that may be scheduled. The result is a matrix of defuzzified values where the value of one cell describes how compatible the sequence of the job of a column after the job in a row is according to all rules. Table 3.4 shows this matrix for our example, the jobs being the ones taken from Table 3.2. It will be used for the construction of the preliminary schedule and during the improvement process. The values in Table 3.4 are the previously used linguistic terms, since for the sake of understandability we have replaced the defuzzified values with the name of the fuzzy term to which the defuzzified value mainly belongs. Of course the defuzzified real values are still used for further numerical computations.

To evaluate schedules during improvement steps, an evaluation value for the compatibility of the entire schedule must be computed. This can be achieved with a fuzzy *and* operator. Again, the fuzzy *and* operator could be realized by taking the minimum of the arguments, but other mathematical models more appropriate in many cases can be found in Section 5.5. For a given schedule S with N jobs, the evaluation function is given by:

$$\mathrm{comp}(S) \stackrel{\mathrm{def}}{=} \bigwedge_{i=1}^{N-1} \mathrm{comp}(H_i, H_{i+1}) \tag{3.5}$$

In a real world application like scheduling a steelmaking plant, many optimization criteria compete with each other. Chen et al. [64] describe one approach to

$H_0 \setminus H_1$	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
h_0	—	high	high	high	high	high	high	med	med	med	high	high
h_1	med	—	high	high	med	med	med	med	med	med	med	high
h_2	low	med	—	v.-high	low	low	low	high	med	med	med	med
h_3	low	med	v.-high	—	low	low	low	high	med	med	med	med
h_4	high	med	med	med	—	v.-high	med	med	high	high	med	med
h_5	high	med	med	med	v.-high	—	med	med	high	high	med	med
h_6	med	low	med	med	med	med	—	low	high	high	low	low
h_7	v.-low	v.-low	high	high	v.-low	v.-low	v.-low	—	low	low	low	low
h_8	low	low	v.-low	v.-low	low	low	med	low	—	v.-high	med	high
h_9	low	low	v.-low	v.-low	low	low	med	low	v.-high	—	med	high
h_{10}	med	med	med	med	low	low	med	low	high	high	—	high
h_{11}	med	low	low	low	v.-low	v.-low	low	low	v.-high	v.-high	v.-high	—

Table 3.4: Compatibility matrix for heat sequences on furnace eaf1, the jobs being the ones taken from Table 3.2. H_0 precedes H_1, e.g., the compatibility of heat h_2 preceding h_1 is *medium*, whereas h_1 preceding h_2 is *high*. To save place, the linguistic terms have been abbreviated in obvious ways.

handle multiple objective scheduling using fuzzy sets. Similarly, the system presented in this Chapter uses operators from fuzzy set theory to compound those conflicting objectives. Each different objective is introduced through the calculation of an importance measure for jobs.

The importance of jobs is used to control the generation of a schedule by scheduling the most important jobs first. In this context, the importance is a combination of the difficulty to schedule a job in general and the importance to schedule it for the actual planning horizon.

A job that requires a bottleneck resource like the continuous caster or the teeming bay for a BEST-ingot is usually difficult to schedule. However, the difficulty depends on the number of jobs with such characteristics. If only one job has to be performed on the continuous caster, then this job is not difficult.

A job with a certain delivery date is *urgent*, because it must be scheduled in the planning horizon in which the delivery date falls. Jobs that are not that important may be shifted to the next planning horizon. To get such a shifted job ever scheduled, it is necessary that the importance of the job increases over time. The range of fuzzy values to represent this importance is: *urgent, very-important, important, medium,* and *not-important.*

The classification of jobs in the list depends on the situation in the actual planning horizon. If many large ingots are produced, these orders are difficult to schedule, because there is not enough space for the solidification process. If many heats that are cast into small ingots are to be scheduled, these are difficult jobs, because of the objective to achieve a uniform distribution of work load.

The inverse evaluation is necessary for chemical ingredients: if for the actual planning horizon many jobs with a high chromium-nickel-alloy exist, as it is in

urgent	$\{h_8\}$
very-important	$\{h_2, h_3, h_4, h_5\}$
important	$\{h_7, h_9\}$
medium	$\{h_6, h_{10}, h_{11}\}$
not-important	$\{h_0, h_1\}$

Table 3.5: Classification of jobs from Table 3.2.

Table 3.1, then a high percentage of chromium (Cr) is no problem. On the other hand, when there are only few jobs with high nickel (Ni) percentages, these jobs can be difficult to schedule. Job h_7 in Table 3.2 has a disproportionate amount of nickel in relation to the other jobs in the list and must be scheduled early. For the jobs of Table 3.2, we obtain the classification of jobs shown in Table 3.5.

One objective of our strategy is to schedule as many jobs as possible. However, in order not to forget the difficult jobs, these are scheduled first. Furthermore, the evaluation function for an entire schedule must contain a factor representing the importance of jobs. Hence, an evaluation function is defined to assign an importance value to a schedule, with N the number of jobs:

$$\text{importance}(S) \stackrel{\text{def}}{=} \bigwedge_{i=1}^{N} \text{importance}(H_i) \tag{3.6}$$

Again, the fuzzy *and* operator could be realized by taking the minimum of the arguments, but other more appropriate mathematical models can be found in Section 5.5.

3.8 Constructing a preliminary schedule

To generate a preliminary schedule, the jobs are classified according to their importance. Then they are scheduled in the sequence of their importance. The *urgent* and *very-important* jobs are scheduled first. To be scheduled means that a temporal interval is assigned to them that describes the time when a job is to be processed in the electric arc furnace. The assigned intervals can spread over the entire planning horizon because of temporal and resource constraints. To simplify our example we assume slots of two hours in the schedule. In reality the duration of jobs varies up to five hours and this variation must be considered by the system, too.

During this scheduling process, empty intervals may remain between scheduled jobs. The compatibilities with the jobs before and after these empty intervals are not considered. If empty intervals with a duration of approximately one job remain, they are filled with compatible jobs as soon as possible. During this scheduling process, the compatibility matrix as shown in Table 3.4 is used.

Schedule 1-1:		med	med	low	v.-high	high	v.-high	med	med	high	v.-high	med
eaf1	h_2	h_1	h_7	h_8	h_9	h_{11}	h_{10}	h_6	h_0	h_4	h_5	h_3
time:	5am	7am	9am	11am	1pm	3pm	5pm	9pm	11pm	1am	3am	5am

Schedule 3-1:		high	high	high	high	high	low	med	high	high	low	high
eaf3	h_3	h_9	h_{10}	h_{11}	h_0	h_1	h_2	h_6	h_5	h_8	h_7	h_4
time:	5am	7am	9am	11am	1pm	3pm	5pm	9pm	11pm	1am	3am	5am

Table 3.6: Intermediate schedules for example heats on eaf1 and eaf3.

A special strategy is applied to prune the search space. It is comparable to preprocessing techniques in constraint satisfaction problems (CSP) as described in Dechter and Meiri [78]. The objects in our CSP are the heats. For every heat a set of possible successors exists. With constraint propagation the set of successors can be reduced. If one job is the only possible successor of another job, it cannot be any more the successor of a third job. If for a heat H_0 only one heat H_1 has a good compatibility value, then the two heats can be interpreted as one job consisting of two heats. H_1 is the definitive successor of H_0. If one of these heats is scheduled, the other one is scheduled automatically, too. If a heat H_0 has two possible successors, H_1 and H_2, and H_1 is scheduled after another heat, the heat H_2 will be assigned as the definitive successor of H_0. Jobs with no sequence-compatibility are not scheduled one after the other.

To illustrate the generation we explain the search for a schedule from the orders given in Table 3.2. We use the classification of jobs given in Table 3.5. The whole schedule for both furnaces is shown in Table 3.6. The system considers the jobs in the following sequence:

1. h_8: In the list of jobs given in Table 3.2, job h_8 has a delivery date and is compatible with only few jobs. Therefore it was classified *urgent* and must be scheduled first. It is scheduled at 11am.

2. h_2, h_3: Next, jobs h_2 and h_3 are scheduled because they are *very-important* jobs. They need a long time span between each other because they are cast into BEST-ingots. One is placed in the first and the other in the last slot of the schedule.

3. h_4, h_5: Jobs h_4 and h_5 should be scheduled one after the other since they are cast with the same format on the HCC-unit. In the list for furnace eaf3, there is another job that will be produced on the caster. Since this job is cast with a different format, a maintenance interval is necessary between these jobs. The single job should be scheduled as early as possible and the two jobs of the first

list as late as possible. Consequently, h_4 and h_5 are scheduled before the last slot.

4. h_7: *Urgent* and *very-important* jobs have now been scheduled. No small intervals exist so that the system can proceed with the *important* job h_7. This job is difficult to schedule because it has only few compatible successors. The two best fitting jobs h_2 and h_3 are not available. There are four potential successors with low compatibility. To generate fewer small intervals, job h_7 is scheduled before h_8.

5. h_9: Job h_9 is then scheduled optimally after h_8.

6. h_1: At that time the strategy is changed and a job that fits best in the slot between h_2 and h_7 is sought. Job h_1 is a good candidate.

7. h_{11}: Four jobs remain for two empty intervals. Since job h_{11} has a request time 'day shift', it should be scheduled as early as possible. It is placed after h_9.

8. h_{10}: Since h_{10} is a very good successor, it is scheduled thereafter.

9. h_0, h_6: Job h_0 should be scheduled before h_4 since h_6 does not fit well.

The resulting schedule is illustrated in Schedule 1-1 of Table 3.6. The compatibilities are shown in the line above the heat sequence. We assume that Schedule 3-1 was constructed for the other furnace eaf3. The problems in the list for this furnace are the molybdenum- and manganese-compatibilities. The algorithm to construct an initial schedule is simple and it is sketched in Table 3.7.

3.9 Improving the schedule by repair

Usually, some jobs cannot be scheduled because they will always violate some compatibility constraints. In addition, some empty intervals may remain in the schedule, and the compatibility between the jobs adjacent to these intervals is usually poor. Instead of taking back the last scheduling decisions by backtracking, we try to repair or improve such a preliminary schedule. In our example no empty intervals exist and no jobs remain. However, there are some ways to improve this preliminary schedule.

To improve a schedule, an evaluation function is required. One potential evaluation is the sum of violated constraints minus the correctly scheduled jobs. Unfortunately, the violation of constraints can have far reaching consequences. The violation of a temporal constraint can cause more resources such as additional energy to be consumed, or may require rescheduling in the next plants. The violation of a chemical constraint can result in the loss of a heat that would be an important financial damage. On the one hand, hard constraints that may not be relaxed must

```
schedule     list_of_jobs::construct_initial_schedule
{
  s ← new schedule;
  matrix ← new list_of_jobs.classify.build_matrix;
  repeat
    s ← s.insert(list_of_jobs.get_most_important_job,matrix);
  until list_of_jobs.no_very_important_job_left;
  repeat
    while s.has_single_gaps
      s ← s.fill_gap(list_of_jobs,matrix);
    repeat
      s ← s.insert(list_of_jobs.get_most_important_job,matrix);
    until s.has_single_gaps or list_of_jobs.is_empty or s.is_full;
  until list_of_jobs.is_empty or s.is_full;
  return s;
}
```

Table 3.7: Pseudo-code of an algorithm that constructs an initial schedule.

be satisfied, and on the other hand, constraints may be relaxed to a certain degree to get a feasible schedule with as many jobs as possible. To evaluate antagonistic constraints, an evaluation function based on the fuzzy values seems to be adequate, since the grade of the satisfaction of a constraint is evaluated too.

We have defined a repair strategy based on fuzzy evaluations. The actual schedule is called the 'currently best schedule'. This schedule can usually be improved. To improve it, the system looks for a constraint being insufficiently satisfied. For the first furnace, such a violation is found between heat h_7 and h_8. Therefore one of them is taken out of the schedule. There are two reasons to remove h_7: heat h_8 has a delivery date, and h_7 probably causes the conflict because it is a *very-difficult* job to schedule. A better place is sought, such as the one before h_3.

There are two possibilities to clear this slot. All jobs between h_1 and h_3 could be shifted by one place, or h_5 could be taken out of the schedule. The first alternative is achieved easier. The result is shown in Schedule 1-2 in Table 3.8. The disadvantage of this schedule is that the delivery date cannot be met exactly. However, it is better than the first schedule.

With the second strategy a better schedule cannot be found straight away since it is not possible to schedule h_5 in the morning. Job h_5 should be scheduled before h_4. To achieve this, heat h_0 can be scheduled into the old slot of h_7. The result is shown in Schedule 1-3. It will become the 'currently best schedule' that may be improved further. Especially if we consider the aspect of the load of the workers,

Schedule 1-2:	med	med	v.-high	high	v.-high	med	med	high	v.-high	med	high	
eaf1	h_2	h_1	h_8	h_9	h_{11}	h_1	h_6	h_0	h_4	h_5	h_7	h_3
time:	5am	7am	9am	11am	1pm	3pm	5pm	9pm	11pm	1am	3am	5am

Schedule 1-3:	med	med	med	v.-high	high	v.-high	med	med	high	v.-high	med	
eaf1	h_2	h_1	h_0	h_8	h_9	h_{11}	h_{10}	h_6	h_5	h_4	h_7	h_3
time:	5am	7am	9am	11am	1pm	3pm	5pm	9pm	11pm	1am	3am	5am

Table 3.8: Intermediate schedules for example heats on eaf1.

more improvements are possible.

Every exchange of jobs in the schedule, every exchange between jobs in the schedule and jobs in the list, and each shift of jobs can be interpreted as an operator in a search process. The search for better schedules is guided by heuristics based on our evaluation function. This heuristic search is a kind of hill climbing method. Unfortunately, the disadvantage of such a method is that it can be caught in local maxima. Glover [156, 157] describes a technique called tabu search that can be used to overcome this problem. This technique allows the system to choose a slightly worse schedule as 'current best schedule' to escape the local maxima. To restrict the search and to avoid pathological cycles, a tabu list in form of a ring buffer describing which operations may not be performed anymore for a certain number of steps in the search process is used.

If no further constraint violation can be detected or no further improvement is achievable, the search for the best schedule ends. Judging whether an improvement can still be achieved is generally difficult. It makes sense to define a distance function between an optimal schedule where all compatibilities would be *very-high*, and all the other constraints would be satisfied, too. Thus, the distance function is the sum of the deviation of all constraints from their optimum. If such a function is available, one can restrict the search effort by a ratio between distance and search effort. It would be fruitless to invest much more search effort if only a small distance exists or with great effort only small improvements are achieved. On the other hand, if the distance is large, one should search longer for a better schedule. A simplified version of this repair algorithm is given in Table 3.9.

Chapter 6 presents results of a combination of repair based tabu heuristics together with gradual constraint satisfaction, and compares these results with several other methods, showing that the results achieved using the presented method are in general very good.

```
schedule    current_best_schedule::repair(list_of_jobs,matrix,limit)
{
  s ← new current_best_schedule;
  tabu ← new tabu_list;
  search_effort ← 0:
  repeat
    job1 ← s.find_violation(tabu);
    op ← s.choose_best_repair_operator(job1,matrix);
    case op:
      shift: s ← s.shift(job1);
      x_out: job2 ← list_of_jobs.find_exchange_job(job1,s,matrix);
             s ← s.exchange(job1,job2);
             list_of_jobs ← list_of_jobs.exchange(job2,job1);
      x_in:  job2 ← s.find_exchange_job(job1,s,matrix);
             s ← s.exchange(job1,job2);
    end case;
    if s.eval > current_best_schedule.eval
    then current_best_schedule ← s;
    else tabu ← tabu.add_pair(job1,op);
    search_effort ← search_effort + 1;
  until limit > current_best_schedule.distance / search_effort;
  return current_best_schedule;
}
```

Table 3.9: Pseudo-code of an algorithm that repairs a schedule.

3.10 Comparison to related systems

Numao and Morishita [299] and Stohl et al. [383] have shown that steel production is a worthwhile domain for the application of knowledge based scheduling systems. In contrast to the application presented in this Chapter, these systems are used in plants for mass steel production where steel qualities do not vary as much as in our application. The continuous casters are the main problem and bottleneck resource in these applications. Since the casting process should be continuous, heats must be ready in time for casting. On the other hand, heats should not wait too long, because the steel would consolidate. A backward scheduling strategy is applied in these systems, reasoning temporally from the last operation in the process plan of one job to the first operation. In Chapter 6 we apply the methods developed in the present Chapter together with the knowledge representation methods presented in Chapter 5 to the problem described by Stohl et al. [383] to test the universality of our scheduling strategy.

In contrast to the work of Numao and Morishita [299] and Stohl et al. [383], we apply in this Chapter a kind of forward reasoning since most jobs do not use the continuous caster and the main problem are the chemical constraints in the first unit — the electric arc furnace. In the described systems the required resources are always the same. Only one of several equal units can be chosen. In our application there are different process plans for different steel qualities.

A minimization of waiting time is often given as an evaluation criterion for schedules. In our application this would not be appropriate, because the execution is uncertain and the minimization would be only theoretical. This is also the case for other applications [215, 376]. Therefore no optimal schedule is computable. The goal function of our system is simply to find a feasible schedule violating as few constraints as possible and optimizing the schedule by local improvements.

In handling the problem of scheduling under uncertainty the main difference to other approaches, such as probability calculus [21], is our pragmatic focus on simple modeling. One difficulty with probabilistic approaches is that they usually require judgmental estimates of many parameters for which little or no empirical support is available, and are very tedious computationally. A further problem lies in the contentious conceptual basis for manipulating subjectively derived probabilities in the same way as classical probabilities obtained from empirically observed frequency distributions [215]. Fuzzy set theory on the other hand has had a considerable degree of success in capturing human ability to reason in terms of vague quantities. Additionally, fuzzy logic is a well grounded mathematical theory derived from fuzzy set theory that does not lead to conceptual problems like for example certainty factors [231]. Nevertheless, we must concede that some membership function tuning is required to really get an application right. Boverie et al. [39] have shown that the overlapping of membership functions is a major influencing factor in the design, whereas their number and exact shape seem to be of minor importance.

Woodyatt et al. [435, 436] have used fuzzy set theory to successfully satisfy collections of customer orders while minimizing the number of steel qualities actually produced. They assign metallurgical grades to steel to select the specific applicable grades and then dress the customer orders according to the likelihood of a grade meeting the customer's specifications. Finally, they combine orders with matching fuzzy grades to optimize the productivity and yield of a continuous caster. Their approach is similar to ours since in both cases fuzzy set theory is used to identify compatible orders. We nevertheless go further by applying fuzzy techniques to a much broader set of constraints used to actually schedule all orders available.

3.11 Conclusion

Due to unreliable data, vague formulation of knowledge, and conflicting objectives in scheduling applications, mathematical-analytical methods as traditionally used are often insufficient. We have illustrated this problem for a steelmaking plant. To overcome this deficiency we have developed a solution that combines two sound AI-techniques for problem solving: approximate reasoning and constraint satisfaction. Our knowledge representation technique covers the uncertainty of problem domain knowledge and supports the straightforward generation of a schedule based on the importance of the jobs. However, due to this ad hoc generation of schedules, some jobs usually remain unscheduled. We have proposed a control strategy that deals with several types of constraints (temporal, spatial, chemical), and supports the dynamic relaxation of conflicting constraints.

Additionally, the generation of robust schedules is stimulated by using fuzzy sets. The heats that are cast on the continuous casters are scheduled on different ends of the schedule because this improves the robustness of the schedule. We can describe the duration between both heats as a temporal fuzzy value. A better evaluation of the schedule will be result if the interval between these heats is longer.

The presented control strategy can also be used to handle emergency cases. If some event like a delay occurs, the schedule is evaluated again, taking into account the changed parameters. In case that this value is worse than a specified level of quality, a repair is necessary. By applying the repair strategy, we obtain reactive scheduling behavior, and it becomes possible to react dynamically to events during job execution.

Additionally, improper conditions for consecutive jobs require immediate and dynamic adaptation. The strategy supports this adaptation by assisting the human expert in relaxing constraints. Using this approach, it becomes possible to evaluate different scenarios before actual activities are performed. We call this kind of problem solving 'what-if' games. Such a simulation prevents human experts from causing troubles with improper decision making. Finally, the decision process is more transparent. However, to support the evaluation and experimentation with chemical element constellations as well as production constraints, we have to develop a sophisticated human computer interaction concept. In particular, the condition membership functions for inference rules as shown in Table 3.3 should remain under control of the human expert. The compatibility rule, element constellation, shape of the membership functions, and the weights of the fuzzy inference rules should be considered during what-if games with the schedule to support estimates of schedule modifications. The condition membership functions in Table 3.3 for example can be adapted for each element in two ways. First, their general shape can be altered to get sharper or softer transitions from one linguistic term to the next. Second, the compatibility rule can be changed from 3% to 2.5%. Additionally, the relative weights of the fuzzy inference rules can be adapted to the relative importance of the playing factors, e.g., the work load constraint could be classified to be more important and thus receive higher weights than chemical constraints. These adaptations need a lot of fine tuning, therefore the engineers should have the opportunity to experiment with the system to be able to match their own way of decision making more accurately. An enriched, spreadsheet-like environment is the proper interaction technique for this type of correlated information. In such an environment, the change of one dimension can be traced simultaneously with the remaining dimensions. This immediate feedback enhances the way the engineers can experiment with their assumptions to find better values for the system's parameters. In Section 5.8, we have further developed this idea to a complete methodology which allows to build consistent configurations comprising constraints with weights and associated membership functions, and the chosen aggregation operators. In addition, the human expert can save different version of the environment for later reuse or experimentation, i.e. dealing with different instantiations of the knowledge base to represent different general optimization criteria during schedule generation.

The idea behind our methodology is to allow easier modeling of the activities of human scheduling experts. The system presented in this Chapter is successful in simulating the human performance. We believe that using the described techniques, the development cycle for scheduling systems becomes shorter and the knowledge representation easier. We assume that with the given techniques, better schedules can be generated since the human expert can easily tune the problem solving process. On theoretical grounds, the search space will normally increase if constraints are defined in a meaningful way, since more compromise solutions that perform trade-offs between antagonistic constraints of differing priorities are taken into account. Thus, chances are raised to find good compromises that would not have been envisioned in a classical setting. At the same time, the modeling capability provided by the model presented in this Chapter, to be further developed in Chapter 5 regarding knowledge representation using fuzzy constraints, and in Chapter 6 regarding repair based strategies, induces a graduation of the search space which guides and facilitates the use of heuristics while at the same time allowing a much richer but still easily understood representation of the domain knowledge.

Chapter 4

Fuzzy expert system to predict maintenance intervals in a continuous caster

Thus in all these cases the Romans did what all wise princes ought to do; namely, not only to look to all present troubles, but also to those in the future, against which they provided with the utmost prudence.

Niccolo Machiavelli, The Prince

When any mechanical contrivance fails, it will do so at the most inconvenient possible time.

Johnson's First Law, Murphy's Law Complete

The following Chapter presents a fuzzy expert system that predicts maintenance intervals for a continuous caster unit in a steel plant. This is a partial task required to show how possibility distributions in data can be accommodated in fuzzy scheduling.

During short term scheduling in a steel plant, one problem is to know the expected service life and maintenance intervals of a particular equipment in advance. In this Chapter we propose a fuzzy expert system for a tundish in a continuous casting shop such that the results from this system can be used as input for preparing the short term production schedules for the shop as a whole. Fuzzy inference rules are used to process input data and to compute the life expectancy of the tundish. Wheras human operators tend to use pessimistic values in order to be on the safe side, the proposed system performs better in predicting the life expectancy of tundish since it takes into account the interacting influences of several variables and also is able to reason with up-to-date information. Therefore, the final schedule thus prepared is closer to real life situations, thereby reducing waste, minimizing production delays, and improving product quality.

Both systems are investigated in a joint project between the Austrian Industries Holding and the Christian Doppler Laboratory for Expert Systems. The scheduling system is described in Chapter 6.

4.1 Introduction

Optimal scheduling of tundish is possible only when the expected life of tundish is known in advance. One strategy when scheduling is to minimize tundish changes in casting sequences so as to extend the utilization time of each tundish. This time is mainly limited by wear and nozzle blockage. Another scheduling strategy is to group the heats together on the basis of quality only. Human scheduling experts normally play safe and assume a life time of 240 minutes for one tundish. Problems arise when the real life time is shorter or longer than 240 minutes. This results in quality degradations or even the need to reschedule remaining heats, with possibly far reaching consequences for delivery dates to customers. Additionally, in case of interruptions for unrelated reasons, e.g. machine breakdowns, knowledge about the remaining life expectancy of the tundish becomes necessary in rescheduling the production.

A detailed analysis of the scheduling problem is given in Chapter 6. Mathematical or analytical methods as used traditionally are often inadequate for handling scheduling problems. This is due to three reasons: The imprecise information of the production process, combinatorial complexity of the search space, and conflicting objectives for production optimizing. The combination of several knowledge based techniques, especially approximate reasoning and constraint satisfaction techniques, offer a promising method to handle these problems. A case study to demonstrate how knowledge based scheduling works with the desired capabilities to schedule short term production is described in Chapter 3. The applied knowledge representation technique covers the vagueness which is inherent in the problem domain by using fuzzy set theory. Based on this knowledge representation, the importance of jobs is defined. This classification of jobs is used for the straightforward generation of a schedule.

An ideal control strategy should incorporate several types of constraints, namely organizational, spatial, and chemical ones. This will allow dynamic relaxation of conflicting constraints for improving the schedule. As an example for the benefits of using this strategy the generation of a schedule for one day is explained in detail.

The present Chapter deals with only one part of the scheduling problem, namely the computation of the tundish's life expectancy. The critical part of the tundish is a pipe called the submerged entry nozzle. It is attached to the bottom of the tundish for transferring liquid metal into the mould of the caster. At the surface of the liquid metal in the mould, aggressive slag attacks the submerged entry nozzle. To extend the utilization time of one tundish, the submerged entry nozzle is increasingly immersed. Thus the slag exposed surface of the submerged entry nozzle changes with time, therefore avoiding a premature wear of the submerged entry nozzle. However, if the bottom of the submerged entry nozzle breaks away, the tundish becomes unusable.

4.2 Fuzzy expert system

The basic principles of construction of fuzzy expert systems can be found in [294] and [212]. Related work on using knowledge based systems for mold bath level control of continuous casters has successfully been carried out by [349]. They faced difficulties in maintaining an optimal bath level with the help of a PID controller and a slide gate controller with fixed parameters, because the characteristics of casting conditions fluctuated during operation. In order to solve this problem, an expert system has been applied to the mold bath level control system. In this traditional expert system, operator knowhow regarding control parameter adjustment is represented in the form of a knowledge base, and that knowledge base is driven by an inference engine only when a significant fluctuation of the mold bath level occurs. This system has been applied to a continuous round caster, and has effectively regulated the mold bath level fluctuations by adjusting the controlling parameters to optimize the state of operation when the fluctuations did occur.

The expert system proposed in this Chapter predicts the life expectancy of submerged entry nozzles. It uses fuzzy inference rules to process input data such as steel qualities, actual and predicted casting speeds, type of submerged entry nozzle, and immersion history of submerged entry nozzle, and computes the life expectancy. The inference rules used by this fuzzy expert system are meaningful because:

- uncertain or rapidly changing data, such as the condition of the submerged entry nozzle or the casting speed, can be used,
- vague rules, such as "IF the steel contains *little* carbon, THEN the slag will be *very aggressive*", are easy to formulate and will work as-is, and
- the scheduling systems described in Chapters 3 and 6 also use fuzzy values. Therefore, when combining the proposed system with a similarly constructed

scheduling system, a process called defuzzification is delayed until the last decision making step. Thus, any premature loss of information is avoided.

A sample fuzzy computation is explained below. As an intermediate step in the life expectancy computation of the submerged entry nozzle, the aggressiveness of the slag is calculated from the chemical contents of the metal. The concentration of six chemical elements has to be monitored simultaneously. Their influence can be simulated by a complex non linear function that normally is represented in several tables that are looked up by the engineers during their work. In the fuzzy expert system proposed here, these tables can more exactly be stated as rules describing the influence of the various factors on the outcome. The interpolation is done automatically through the fuzzy inference engine. This system also readjusts to the continuous changes in parameter values when slightly different grades of steel are produced one after the other through the same tundish.

The system performs better than humans in predicting the life expectancy since it considers more types of influence and reasons with up-to-date information. This is especially important when rescheduling has to be done in a short time. For these reasons, the final schedule matches closer to reality.

The interface between the system proposed here and the scheduling system such as those described in Chapters 3 and 6 works smooth in that all data are available to the scheduling expert system at any time. In case of a special situation, such as a broken submerged entry nozzle or a breakout of the strand, both systems are informed immediately by the process monitoring system that writes the data into a globally accessible database. The scheduling expert system is programmed to ignore all data coming from the expert system proposed in this Chapter in case of an ambiguous situation. Further, the proposed system can signal a sudden change in expected life time for the current tundish by a special high priority signal to the scheduling expert system so that the rescheduling procedure can be initiated. The information flow is more or less unidirectional, from the system proposed in this Chapter to the scheduling system.

4.3 Conclusion

In this case, a fuzzy expert system is the best choice regarding easiness of implementation, knowledge formulation, and in keeping the knowledge base up-to-date. Tuning the system is possible since a lot of test data were collected previously during regular production. Further, the problem is small, well defined, and stable. This is also the reason for not incorporating the function performed by the system into the scheduling system itself. The interplay between these two systems can be seen as the study of a real world application of distributed knowledge based systems.

The idea to use a fuzzy expert system for this type of problem came after doing a study of expert system technology in the Japanese steel industry [361] during a two year sabbatical at the University of Tokyo.

We are planning to extend the design of the system proposed in this Chapter to allow it to learn according to the feedback from the actual process data. In a similar fashion, Kominami et al. [226] have used a neural network in their Yawata Works plant to forecast the breakout of the strand in continuous casters. We are currently studying the combination of our technique based on fuzzy logic with the learning capability given by neural networks, genetic algorithm, and case based reasoning.

Chapter 5

Fuzzy multiple criteria representation

πάντα ῥεῖ …

ʿHηϱάκλειτοσ (500 B.C.)

Class schedules are designed so that every student will waste the maximum time between classes.

2nd Law of Class Scheduling, Murphy's Law Complete

In this Chapter we explain in theory and by detailed examples fuzzy set based constraints that help to model general multiple criteria optimization problems. We simplify the mathematics needed for a method of eliciting the criteria's importances from human experts. We introduce a new consistency test for configuration changes. This test also helps to evaluate the sensitivity to configuration changes. We describe the implementation of these concepts in our in our fuzzy constraint library ConFLIP++ based on our fuzzy logic inference processor library FLIP++, and in our dynamic constraint generation library DynaFLIP++ based on ConFLIP++.

5.1 Introduction

Scheduling is the task of allocating resources to jobs with respect to time. An example would be production scheduling in industrial environments, where some typical criteria to be optimized might be usage of equipment, space, material and human resources, as well as product quality, production timeliness, minimum total overdue time, minimum number of overdue parts, minimum mean processing time of parts, and robustness to changes due to machine breakdowns. Panwalkar and Iskander [313] list over 100 heuristic rules how to take into account such criteria. Scheduling has been studied in the operations research literature since the early fifties. Theoretical work has brought several improvements over the years, but there are still many problems with the formal-analytical approaches:

- the algorithms are too complex for real-world applications,
- the models consider only one or two criteria,
- only linear relations among parameters are considered,
- the models demand exact knowledge about durations and technical constraints,
- no antagonistic knowledge can be modeled, and
- the effort to formalize a new scheduling problem is considerable.

A combination of constraint satisfaction techniques and concepts handling vagueness and uncertainty offers a solution to these problems. First, constraint satisfaction techniques can be enhanced by heuristics to reduce the inherent complexity. Second, additional criteria do not necessarily increase the problem complexity when using heuristics, and constraints lend themselves perfectly to model an arbitrary number of criteria. Third, concepts handling vagueness and uncertainty offer the possibility to model non-linear relations intuitively, as well as to reason with incomplete or probabilistic knowledge. We adopted fuzzy set theory and fuzzy logic because they lend themselves well to the task of dealing with vagueness and uncertainty. Fourth, the combination of these techniques can be used to handle contradicting knowledge. And finally, in knowledge based systems, the available knowledge is described explicitly, therefore it is easy to develop and to maintain. Additionally, understandable explanations for decision made can be generated.

The algorithm described in Sections 5.2 and Chapter 6 combines repair based [277, 465] and tabu list [156, 157] techniques with fuzzy constraint relaxation techniques [64, 120, 133, 164, 179, 270, 462]. Its application to flow shop scheduling in a steelmaking environment has been described in [101].

The employed technique is highlighted on a conceptual level in Section 5.2 from the point of view of fuzzy multiple criteria optimization, as defined in [437, 462].

Constraints from application domains are often vaguely specified and thus lend themselves perfectly for being reformulated as fuzzy constraints. We develop methods of formalizing the relative importance of these constraints. We analyse also the associated problem of handling the chaotic behavior of the search algorithms when parameters describing constraints are changed. Additionally, sensor-measured process variables as well as production parameters are often uncertain, thus calling for the usage of plausible and approximate reasoning, in particular possibilistic logic [113] for constraint evaluation and inference. To facilitate experimentation, a method of softening or hardening complete constraint satisfaction problems is provided. Human experts can thus make trade-offs between for example higher production quality and the ability to schedule more difficult jobs.

Repair based methods as described in Chapter 6, sometimes called iterative improvement techniques in the literature, have proven to be very efficient heuristics for some types of problems with high computational complexity, solving for example the n-queens problem with linear time and space complexity [277]. They are well suited for cooperative and reactive scheduling problems, typical applications in manufacturing. Since unexpected machine failures can happen at any time, the ability to quickly reschedule while changes should be kept as local as possible in order not to disturb unrelated activities is an absolute necessity. Repair based methods are easily transformed into 'anytime' algorithms since good solutions can be found early, and, if time permits, an arbitrary amount of time can be used to find even better ones. The 'anytime' feature is important for reactive scheduling when decisions must be made quickly, time to think is scarce, and settling for a slightly suboptimal solution is acceptable. Since repair based hill climbing is a greedy search method, we combine it with a tabu list technique to avoid being stuck in local maxima or pathological cycles.

5.2 Fuzzy multiple criteria representation

Consider problems where many different 'criteria' have to be taken into account, various 'objectives', 'aims', or 'goals' must be fulfilled, some 'aspiration levels' should be aimed at, 'domains' of variables must be respected, predefined 'importances' of certain objects have to be considered, briefly, 'side-conditions' have to be observed. On a conceptual level, there is a difference between these notions. For instance, 'criteria' more or less specify what a solution *must* look like, while 'aspirations' specify what a solution *should* look like. However, in an engineering context all these 'side-conditions' are usually formulated in one and the same framework, namely by overloading the term 'constraint' with all these notions. Zimmermann [462] does not distinguish between 'constraints' and 'objectives', arguing that it empirically models the behavior of decision makers quite well. Following this usage, the present text does not differentiate between all these notions.

The theoretical structure needed to deal with problems defined by such constraints will be given in the next sections: Section 5.3 first outlines characteristics of fuzzy constraint satisfaction problems that are of immediate relevance for fuzzy multiple criteria representation issues. More details regarding fuzzy constraint satisfaction problems can be found in Chapter 6. Section 5.4 introduces fuzzy constraints formally. Section 5.5 explains aggregation operators fuzzy constraints, including a discussion of aggregation operators that allow trade-offs between several constraints. Section 5.6 goes on to explain how to combine constraints of different importance. Section 5.7 develops ways to compute importance ratio scales for constraints. Section 5.8 introduces a practical test enforcing consistency between earlier decisions and configuration changes. Section 5.9 presents the tools needed to guide the algorithms explained in Chapter 6, in particular how to build a decision function taking into account antagonistic constraints. Section 5.10 then describes the fuzzy constraint library ConFLIP++ implemented to test the hypotheses introduced in the previous sections. In Chapter 6 we blend all these concepts with constraint optimization techniques to tackle real-world scheduling problems. All sections come along with small examples to motivate the introduced techniques.

5.3 Fuzzy constraint satisfaction problems

Constraints are mathematical objects used to make explicit the logic behind a problem. They are used to model decision making problems of e.g. design, planning, or scheduling. Classical constraint satisfaction problems are usually composed of 'crisp' constraints, sometimes called 'boolean', 'yes-no', or 'hard' constraints, i.e. relations that can be either satisfied or not, without intermediate state. A solution must satisfy all constraints of the problem. If a problem has more than one solution, it is called an 'underconstrained problem'. To differentiate between these solutions, the decision maker has to consider additional constraints. If a problem has no solution at all, it is called 'overconstrained'. Some constraints must then be 'relaxed' to find acceptable solutions. Both cases are common in real-world situations, for example when buying shoes. Depending on the criteria and objectives considered, there can be many fitting offers, or none at all. Nevertheless, almost everybody will eventually be able to find acceptable shoes, more or less consciously optimizing and compromising between everything that could have some influence on the outcome. Additionally, it turns out that some criteria are easily formulated with words, but cannot as easily be put into a formula. It is for example not easy to formalize how well the shoes will match one's clothes. Human language is also vague when specifying that the new shoes should be *not too expensive*, although there is almost no uncertainty about the price that will have to be paid for a certain pair.

Additionally, fuzzy constraints are well equipped for their use in repair based constraint satisfaction algorithms as discussed in Section 6.3, since they allow to compare satisfaction degrees of constraints in a natural way.

5.4 Fuzzy constraints

A classical crisp k-ary constraint C_{crisp} between a set of variables $x_1, \ldots, x_k \in D_1 \times \ldots \times D_k$, where D_j is the domain of variable x_j, can be formalized as a relation R_{crisp} with its characteristic function $\mathbf{1}_{\text{crisp}}$.

$$\begin{array}{rrcl} \mathbf{1}_{\text{crisp}}: & D_1 \times \ldots \times D_k & \longrightarrow & \{0,1\} \\ & (x_1, \ldots, x_k) & \longmapsto & \mathbf{1}_{\text{crisp}}(x_1, \ldots, x_k) \end{array} \tag{5.1}$$

R_{crisp} assigns to each k-tuple $(x_1, \ldots, x_k)$ a value $\mathbf{1}_{\text{crisp}}(x_1, \ldots, x_k)$ from $\{0, 1\}$, with the obvious meaning that those k-tuples being assigned 1 are constraint-satisfying 'instantiations' of C_{crisp}, while the others violate the constraint.

In analogy, a soft k-ary constraint C_{soft} between a set of variables $x_1, \ldots, x_k \in D_1 \times \ldots \times D_k$, where D_j is the domain of variable x_j, can be formalized as a relation R_{soft} with its membership function μ_{soft}.

$$\begin{array}{rrcl} \mu_{\text{soft}}: & D_1 \times \ldots \times D_k & \longrightarrow & [0,1] \\ & (x_1, \ldots, x_k) & \longmapsto & \mu_{\text{soft}}(x_1, \ldots, x_k) \end{array} \tag{5.2}$$

R_{soft} assigns a fuzzy membership value $\mu_{\text{soft}}(x_1, \ldots, x_k)$ from $[0, 1]$ to each k-tuple $(x_1, \ldots, x_k)$. The function μ_{soft} represents the level of preference between different instantiations. A value of 1 means that $(x_1, \ldots, x_k)$ fully satisfies R_{soft}. A value of 0 means that $(x_1, \ldots, x_k)$ is incompatible with R_{soft}, i.e. corresponds to a constraint *violation*. An intermediate value means that the corresponding k-tuple *partially* satisfies the constraint. More generally, $\mu_{\text{soft}}(x_1, \ldots, x_k)$ can be interpreted as the degree of satisfaction of the soft constraint.

It is important to note that if $\mu_{\text{soft}}(x_1, \ldots, x_k) = 0$, then the k-tuple $(x_1, \ldots, x_k)$ does not satisfy relation R_{soft} *at all*, i.e. it is a forbidden k-tuple. This allows us to specify 'hard barriers' that should never be crossed when relaxing a soft constraint. Soft constraints *without* hard barriers can easily be specified, too. The membership functions μ of such a soft constraint without hard barriers must simply be defined such that μ never reachs zero, though it can approach zero up to any coefficient $\varepsilon > 0$, always indicating that the respective k-tuple is inferior compared to others with larger μ. This definition of soft constraints with hard barriers allows compensation of *partially satisfied* constraints by other constraints being satisfied to a higher degree, while *violated* constraints cannot be counterbalanced by the satisfaction of other constraints. Therefore, it is in accordance with the remarks by Dubois et al. [120] about what can be correctly termed a constraint satisfaction problem.

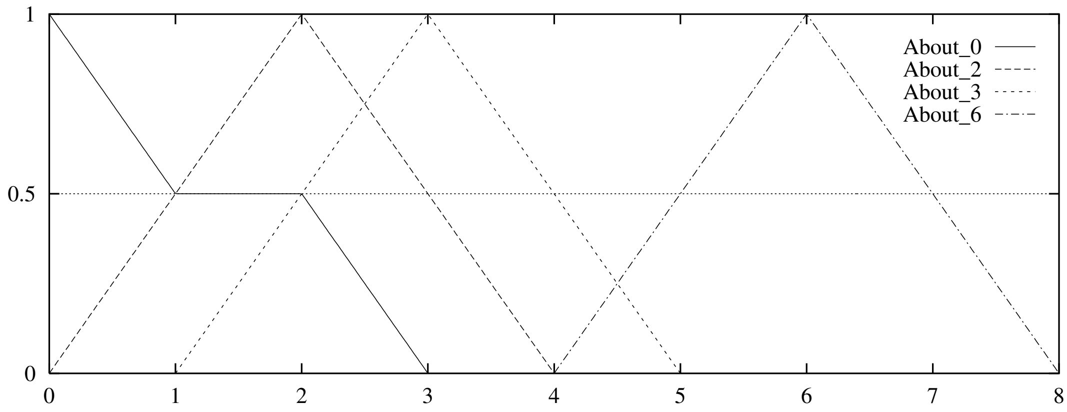


Figure 5.1: Membership functions of the soft constraints from (5.4). For instance, 'About_2' is a 'fuzzy number' such that $x_2 = 1.5$ will be a member of it to the degree 0.75. However, the 'tutorial' example as posed in (5.3) restricts variables to integers.

An example adapted from Fargier et al. [133][1] helps to illustrate the properties of these soft constraints. A tutorial is to be organized. The constraints specify that two professors will share the work: Prof. A will give the lecture part of the tutorial and can teach 2 to 4 sessions, with 3 being ideal (C_1); Prof. B will give the training part and can teach about 2 sessions, with 1 and 3 being half-acceptable (C_2); the tutorial should ideally be composed of 6 sessions, but 5 or 7 sessions would be still acceptable (C_3); additionally, Prof. A and Prof. B should teach each about the same number of sessions (ideally, exactly the same number, but a difference of 1 or 2 sessions is half-acceptable) (C_4). Additionally, all sessions must have a prespecified unit length, according to the rule "You can teach anything in mathematics — except something that takes longer than 45 minutes." These constraints can be rewritten formally as:

$$\begin{array}{ll} x_1 = \text{number of lecture sessions} & \text{with } D_1 = \mathbb{N} \\ x_2 = \text{number of training sessions} & \text{with } D_2 = \mathbb{N} \end{array} \tag{5.3}$$

$$\begin{array}{llll} R_1: & x_1 & \text{is} & \text{About\_3} \\ R_2: & x_2 & \text{is} & \text{About\_2} \end{array} \qquad \begin{array}{llll} R_3: & x_1 + x_2 & \text{is} & \text{About\_6} \\ R_4: & |x_1 - x_2| & \text{is} & \text{About\_0} \end{array} \tag{5.4}$$

$\mathbb{N}$ represents the set of all natural numbers. The membership functions corresponding to the fuzzy subset 'About_i' that describe the relation between values which can be taken by the variables and satisfaction degree of the constraints are depicted in Figure 5.1. Note that these soft constraints still have hard barriers. For example, $x_1 = 1$ can never be part of a solution for this problem given the membership function 'About_3' of Figure 5.1, no matter how much we compromise between constraints.

[1] The original problem and its solution as given in [133] contained some minor typing errors, in particular relations R_2 and R_4 on page 1130 did not correspond to the informal problem description, and $Sat(\vec{u}_1)$ on page 1131 was inconsistent with either problem description. The underlying line of thought of the paper by Fargier et al. [133] is of course correct.

This example will be followed up *à fur et à mesure* in the subsequent sections to illustrate the introduced concepts.

Additionally, any constraint C_{crisp} can be simulated by a constraint C_{soft} to yield the same satisfying instantiations by setting

$$\mu_{\text{soft}}(x_1, \ldots, x_k) \stackrel{\text{def}}{=} \mathbf{1}_{\text{crisp}}(x_1, \ldots, x_k) = 1 \tag{5.5}$$

in which case the solutions of the soft constraint satisfaction problem are exactly the same as those of the crisp constraint satisfaction problem.

The motivation behind defining soft constraints lies in their ability to measure the satisfaction of constraints. They allow

- in case of underconstrained problems, to specify preferences in the valid domains of variables, thus providing a natural way to grade solutions that would otherwise be all equal,
- in case of overconstrained problems, to specify margins for constraints where they can be relaxed while still yielding acceptable results, without the necessity to trigger any explicit constraint relaxation procedure,
- the natural definition of priorities among constraints. See Section 5.6 for a discussion of prioritized soft constraints,
- the propagation of uncertain values such as unknown durations as possibility distributions, where the values are ranked according to their level of plausibility as described in Dubois and Prade [113] and in Dubois et al. [120], and
- in both underconstrained and overconstrained cases, to compute in a natural way an objective function that considers all constraints relevant for the problem. This is particularly interesting for problems in the manufacturing domain. There it is usually impossible to find the overall *best* solution, and settling for a suboptimal but *good* solution is often acceptable. In Chapter 6 we describe how to combine the presented concepts and techniques with heuristic search methods, in order to take advantage of this feature.

However, not all problems can be solved by using soft constraints alone. The next sections will describe how compensatory aggregation operators and priorities of constraints can enhance the ability to represent a problem adequately.

5.5 Aggregating several fuzzy constraints

The next step in solving a general constraint satisfaction problem is to satisfy *several* constraints with one substitution, i.e. one instantiation of all variables satisfying all constraints at the same time. In classical constraint satisfaction problems, the solution is obtained by taking the intersection of all relations, i.e. calculating the conjunction of all constraints:

$$\mathbf{1}_{R_1\cap\ldots\cap R_n}(x_{j_1},\ldots,x_{j_m}) \quad = \quad \bigwedge_{i=1}^{n} \mathbf{1}_{R_i}(x_{i_1},\ldots,x_{i_k}) \tag{5.6}$$

with $\{x_{j_1},\ldots,x_{j_m}\} = \cap_{i=1}^{n}\{x_{i_1},\ldots,x_{i_k}\}$ and $m = |\cap_{i=1}^{n}\{x_{i_1},\ldots,x_{i_k}\}|$.

For soft constraints, as defined in formula (5.2), the same formula as (5.6) can be adapted by choosing a suitable fuzzy conjunction operator[2]. Zadeh proposed to use the *minimum* in his first article about fuzzy set theory [447]. It has been commonly employed since then, most notably in the very influencing article [16] by Bellman and Zadeh, but there exists a plethora of other operators and operator families. In particular, the use of t-norms (triangular norms) as conjunctions has often been advocated because of their pleasing mathematical behavior.

A t-norm is a binary operator $T : [0,1]^2 \rightarrow [0,1]$ such that for all $a,b,c,d \in [0,1]$:

$$\begin{aligned}
T(a,1) &\quad = \quad a && \text{(neutral element)} && (5.7)\\
a \le b \text{ and } c \le d &\quad \Rightarrow \quad T(a,c) \le T(b,d) && \text{(monotonicity)} && (5.8)\\
T(a,b) &\quad = \quad T(b,a) && \text{(commutativity)} && (5.9)\\
T(T(a,b),c) &\quad = \quad T(a,T(b,c)) && \text{(associativity)} && (5.10)
\end{aligned}$$

As one immediately notices, these t-norms are per definition associative by (5.10) and can therefore be extended to an n-tuple by recursively applying the t-norm operator, which is necessary for such formulae as (5.6).

Some basic t-norms with $a,b \in [0,1]$ are

$$\begin{aligned}
\text{minimum:} \quad & T_M(a,b) \stackrel{\text{def}}{=} \min(a,b) && (5.11)\\
\text{algebraic product:} \quad & T_P(a,b) \stackrel{\text{def}}{=} a\cdot b && (5.12)\\
\text{Łukasiewicz:} \quad & T_L(a,b) \stackrel{\text{def}}{=} \max(0,a+b-1) && (5.13)\\
\text{drastic product:} \quad & T_W(a,b) \stackrel{\text{def}}{=} \begin{cases} \min(a,b) & \text{if} \ \max(a,b)=1 \\ 0 & \text{otherwise} \end{cases} && (5.14)
\end{aligned}$$

and some parameterized operators that can be adapted while preserving the features described in (5.7)–(5.10), but that are not distributive:

$$T_{\text{Hamacher}}(a,b) \stackrel{\text{def}}{=} \frac{a\cdot b}{\gamma+(1-\gamma)(a+b-a\cdot b)},\ \gamma \ge 0 \tag{5.15}$$

$$T_{\text{Yager}}(a,b) \stackrel{\text{def}}{=} 1-\min(1,((1-a)^p+(1-b)^p)^{1/p}),\ p \ge 1 \tag{5.16}$$

[2]The use of the term ‘operator’ instead of ‘function’ in this context is recommended because fuzzy operators can take membership *functions* as arguments.

More examples including more parametrized operator families can be found in Zimmermann [461].

What t-norm operators T do for the intersection is done by t-conorm operator S for the union, corresponding to the disjunction in classical logic. Any t-conorm can be generated from its associated t-norm by applying the following equation:

$$S(a, b) \quad = \quad 1 - T(1 - a, 1 - b) \tag{5.17}$$

since the complement, corresponding to negation in classical logic, is usually defined for all $a \in [0, 1]$ as $co(a) \stackrel{\text{def}}{=} 1 - a$.

The minimum operator as well as all t-norms are actually very strict operators to aggregate constraints since the constraint satisfied the least influences the aggregated outcome maximally. It can be shown (e.g. Yager [439]) that for any t-norm T and for all $a, b \in [0, 1]$

$$T(a, b) \quad \leq \quad T_M(a, b) \tag{5.18}$$

Proof [439]: Without loss of generality assume $\min(a, b) = b$. Since $T(1, b) = b$, for all $a \in [0, 1]$, $a \leq 1$, and applying (5.7)-(5.9) once each:

$$T(a, b) \;\leq\; T(1, b) \;\leq\; b \;\leq\; \min(a, b) \tag{5.19}$$

This result together with the associativity (5.10) of t-norms implies that in multiple criteria decision making, the use of a t-norm type 'anding' forbids compensation for one bad constraint satisfaction. Similarly, it can be shown that for all $a, b \in [0, 1]$

$$S(a, b) \quad \geq \quad S_M(a, b) \tag{5.20}$$

with S_M denoting the maximum which is therefore the smallest t-conorm, implying that the use of pure t-conorm type 'oring' allows for no distraction for one good satisfaction. In both t-norm and t-conorm cases, an indifference to individual criteria versus an over-submission to extreme criteria is shown.

On the other hand, human aggregation procedures in decision environments have been analysed by Zimmermann [461] and it has been shown that humans are able to perform trade-offs between conflicting goals when compensation is permitted. In multiple criteria optimization, this is often a requested feature, and various solutions have been developed. Yager proposes in [439] another type of operator called an <u>o</u>rdered <u>w</u>eight <u>a</u>veraging (OWA) operator. The general definition given for this n-ary operator is restated here, with $a_1, \ldots, a_n, W_1, \ldots, W_n \in [0, 1]$, $\sum_i W_i = 1$, and b_j is the jth largest element in the collection $a_1, \ldots, a_n$:

$$\mathrm{OWA}_W(a_1, \ldots, a_n) \quad \stackrel{\text{def}}{=} \quad \sum_{i=1}^{n} W_i \cdot b_i \tag{5.21}$$

For instance, by specifying the weight vector

$$\vec{W}_{\min} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{5.22}$$

the normal minimum operator is obtained. Similarly, the maximum operator or a pure 'averaging' operator with $W_i = 1/n$ can defined.

Yager observes that for any t-norm T and any t-conorm S:

$$\underbrace{T_W \le T \le T_M}_{\text{and-type}} \le \underbrace{\text{OWA}}_{\text{and/or-type}} \le \underbrace{S_M \le S \le S_W}_{\text{or-type}} \tag{5.23}$$

where S_W denotes the t-conorm associated to T_W according to (5.17).

Nice features of this operator family are that all OWA operators are monotonic, invariant with respect to permutations of the input parameters (corresponds to commutativity in the binary case), and idempotent ($\text{OWA}(a, \ldots, a) = a$), but not associative. However, the latter is usually not required since the OWA operator can be defined for an arbitrary number of parameters, and that is exactly what associativity is normally needed for. Problems can arise when the number of arguments is not known in advance and recursive function calling is needed, e.g. when the number of constraints to be considered is unknown. This is acceptable when one bears in mind that the aggregation of already aggregated values is different from the aggregation of the same values taken directly. Such a situation is often encountered in real-world problems. To ensure that the hard barrier introduced in Section 5.4 is never crossed when constraints are aggregated, it is necessary that either the aggregation operator propagates a score of zero, or that the scores evaluating to zero get filtered out during the calculation process before aggregation is applied. t-norms propagate zero scores, but it is easy to extend the OWA operator to yield this property too by redefining (5.21) to:

$$\text{hard\_barrier\_OWA}_W(a_1, \ldots, a_n) \stackrel{\text{def}}{=} \begin{cases} 0 \text{ if } \exists\, i \in [1, n] \text{ such that } a_i = 0 \\ \sum_{i=1}^{n} W_i \cdot b_i \quad \text{otherwise} \end{cases} \tag{5.24}$$

Yager notes that the minimum operator is the *only* t-norm having the idempotency property, while all OWA operators have it. This property nevertheless is almost required for multiple criteria optimization. It makes sure that when all criteria are equally satisfied, the overall score is identical to this individual score of all criteria, independently from the number of constraints aggregated.

Interesting weight vectors for multiple criteria optimization problems are vectors that make trade-offs between conflicting goals when compensation is permitted.

Consider the following n-ary 'soft and'-like compensation weight vector and its associated OWA operator:

$$W_{\text{soft\_and\_}i} \stackrel{\text{def}}{=} \frac{2i}{n(n+1)} \tag{5.25}$$

For instance, $n = 4$ results in

$$\vec{W}_{\text{soft\_and\_4}} = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{pmatrix} \tag{5.26}$$

Actually, the OWA operator family is even more fascinating than can be demonstrated here. It permits to aggregate criteria under the guidance of a quantifier such as "more than four of the criteria must be satisfied." See [439] for a more profound discussion of the properties of the OWA operators, including a worthwhile discussion of their 'andness', 'orness', and 'dispersion'.

The operators introduced are illustrated by extending the tutorial example specified in (5.3) and (5.4). It is clear that there is no solution perfectly satisfying all constraints. However, there are several solutions that partially satisfy the constraints:

$$\begin{array}{rlcrl}
 & \quad x_1 \;\; x_2 & & & \quad\; R_1 \;\;\; R_2 \;\;\; R_3 \;\;\; R_4 \\
\vec{u}_1 = & (\ 2,\ 3)^\top & \text{with} & Sat_{(R_1,R_2,R_3,R_4)}(\vec{u}_1) = & (\,0.5, 0.5, 0.5, 0.5) \\
\vec{u}_2 = & (\ 3,\ 2)^\top & \text{with} & Sat_{(R_1,R_2,R_3,R_4)}(\vec{u}_2) = & (\quad 1, \quad 1, 0.5, 0.5) \\
\vec{u}_3 = & (\ 3,\ 3)^\top & \text{with} & Sat_{(R_1,R_2,R_3,R_4)}(\vec{u}_3) = & (\quad 1, 0.5, \quad 1, \quad 1) \\
\vec{u}_4 = & (\ 4,\ 2)^\top & \text{with} & Sat_{(R_1,R_2,R_3,R_4)}(\vec{u}_4) = & (\,0.5, \quad 1, \quad 1, 0.5) \\
\vec{u}_5 = & (\ 4,\ 3)^\top & \text{with} & Sat_{(R_1,R_2,R_3,R_4)}(\vec{u}_5) = & (\,0.5, 0.5, 0.5, 0.5)
\end{array} \tag{5.27}$$

where $Sat_{(R_1,R_2,R_3,R_4)}$ denotes the vector of the satisfaction degrees for the relations before they are aggregated to compute the actual satisfaction degree.

The aggregated constraint satisfaction values for the tutorial example using the introduced aggregation operators are given in Table 5.1. As one immediately notices, no useful information can be drawn from the results computed by using T_M. T_M cannot differentiate between the results since it considers only the extreme case, because it satisfies (5.7), and because it is idempotent as pointed out before. However, it is reasonable to further discriminate between the results since they partially violate a differing number of the constraints, e.g. $\vec{u}_3$ partially violates only R_2, while $\vec{u}_1$ partially violates all constraints. The results using the T_W, the T_L, or the S_M operators are also not very helpful in disambiguating the ranking of the solutions.

T_P and the OWA operator correctly order the solutions with respect to the information available. They indicate that $\vec{u}_3$ is preferable to $\vec{u}_2$ and $\vec{u}_4$, which are

	T_W	T_L	T_P	T_M	$\mathrm{OWA}_{\text{soft\_and\_4}}$	S_M
$Sat(\vec{u}_1)$	0	0	0.0625	0.5	0.5	0.5
$Sat(\vec{u}_2)$	0	0	0.25	0.5	0.65	1
$Sat(\vec{u}_3)$	0.5	0.5	0.5	0.5	0.8	1
$Sat(\vec{u}_4)$	0	0	0.25	0.5	0.65	1
$Sat(\vec{u}_5)$	0	0	0.0625	0.5	0.5	0.5

Table 5.1: Comparing aggregation operators for the tutorial example. The numbers to the right of $Sat(\vec{u}_j)$ are the aggregated evaluation scores computed with the respective operator heading the corresponding column.

preferable to $\vec{u}_1$ and $\vec{u}_5$. However, only the OWA operator is more or less indifferent to the size of the problem because of its idempotency property, while being at the same time sensitive to all the individual scores. In contrast, all t-norms besides T_M are not idempotent but are insofar 'pessimistic' as they never increase when an additional satisfaction degree is taken into account, even when that additional satisfaction degree is better than all degrees so far aggregated. The OWA operators behave differently: They will increase when aggregating additional better satisfaction degrees, decrease when aggregating additional worse satisfaction degrees, and stay equal when aggregating additional equal satisfaction degrees. Since OWA operators are not associative, it is necessary to switch to a similar OWA operator that has one more parameter when aggregating additional satisfaction degrees. Thus the behavior of the OWA operator family is often better suited for multiple criteria decision making in real-world situations where multistage inference steps are necessary, in order not to dilute knowledge excessively. More aggregation operators have been described in the literature, e.g. Zimmermann [461], but the operators introduced here illustrate well the main issues involved in a typical fuzzy multiple criteria optimization problem. Zimmermann [461, p. 42] correctly points out that the choice of the most appropriate aggregation operator largely depends on the context of the problem one deals with.

All the operators analysed so far are invariant with respect to permutations of the input parameters, i.e. satisfy the generalized commutativity property. This means that all constraints are equally important. Decisions are based only on the set of scores, but the a-priori importance of the constraint responsible for each individual score is not taken into account. Looking at the tutorial example, this means that the aggregated scores of the solution vectors $\vec{u}_2$ and $\vec{u}_4$ will always be equal (see corresponding rows in Table 5.1), using aggregation operators without considering possible different priorities between the constraints, even though the two solution vectors do not satisfy each constraint to the same degrees. In contrast, the aggregated scores of the solution vectors $\vec{u}_1$ and $\vec{u}_5$ will always be the same irrespective of any priorities, since they satisfy all constraints to the same degrees.

The preference ordering that can be concluded for the tutorial example using any of the aggregation operators introduced so far comprises the following three equivalence classes

$$\begin{array}{c} \{\vec{u}_1, \vec{u}_5\}, \{\vec{u}_2, \vec{u}_4\}, \{\vec{u}_3\} \\ \text{with} \quad Sat(\vec{u}_1|\vec{u}_5) \;\leq\; Sat(\vec{u}_2|\vec{u}_4) \;\leq\; Sat(\vec{u}_3) \end{array} \tag{5.28}$$

i.e. $\vec{u}_3$ will normally be selected, not considering any additional information. The notation $Sat(\vec{u}_1|\vec{u}_5)$ means that either $Sat(\vec{u}_1)$ or $Sat(\vec{u}_5)$ can be written instead of $Sat(\vec{u}_1|\vec{u}_5)$.

Fargier et al. [133] reach the result given in (5.28) using the minimum operator and non-numeric (inclusion-based and lexicographic-based) ordering methods, similar to the methods used in crisp constraint relaxation techniques (Freuder and Wallace [144] give a very understandable overview concerning crisp constraint relaxation techniques). However, Fargier et al.'s methods assume that a large part of the scores assigned to the constraints for the various instantiations are equal, such as the scores 0.5 and 1 in the tutorial example, since these methods otherwise just select the largest minimum of the individual constraint satisfactions. It is doubtful if not impossible that in a real-world multiple criteria optimization problem involving hundreds or thousands of variables, such as introduced in Section 6.4, the condition that most scores belong to a small finite subset of [0,1] holds. In the latter case, these inclusion-based and lexicographic-based ordering methods will therefore not be needed to disambiguate solutions.

5.6 Fuzzy constraints of different importance

Often criteria do not all have the same importance in real-world applications. It is thus reasonable to consider the relative priorities of constraints when instantiations are evaluated and compared. An intuitive requirement is that as a constraint becomes more important, it should play a more significant role in determining the overall decision function.

Following-up the tutorial example, we would like to take into account additional information about preference between constraints. For example, having a tutorial composed of 6 sessions might be *very* important because of the way the tutorial will be paid for. In addition, Prof. B's wishes might be *more* important than observing any other side-conditions since Prof. B is the Dean of the Faculty. Accordingly, the precedence of the constraints from (5.4) would reduce to:

$$\{R_2\} \overset{\text{prec}}{\succ} \{R_3\} \overset{\text{prec}}{\succ} \{R_1, R_4\} \tag{5.29}$$

Given this additional information, the five solutions listed in (5.27) do not obey any more the relation given in (5.28). $\vec{u}_1$ and $\vec{u}_5$ will remain the least desirable options. $\vec{u}_4$ is certainly preferable to $\vec{u}_2$, since $\vec{u}_4$ better satisfies the *more important* relation

R_3, in contrast with $\vec{u}_2$ better satisfying only the *normal* relation R_1, everything else being equal. However, no preference ordering can be asserted concerning how $\vec{u}_3$ should be positioned in relation to $\vec{u}_2$ and $\vec{u}_4$ without further numerical preference information. Any situation will be possible, depending on the relative importance assigned to the different constraints. Formally, the situation summarizes at this stage to

$$Sat(\vec{u}_1|\vec{u}_5) \;\leq\; Sat(\vec{u}_2|\vec{u}_3|\vec{u}_4) \quad \text{and} \quad Sat(\vec{u}_2) \;\leq\; Sat(\vec{u}_4) \tag{5.30}$$

of which (5.28) is one particular case.

Several approaches have so far been proposed in the literature to model and further remove the ambiguity remaining in situations such as (5.30). Yager [437] argues that ranking or weighting of objectives can be achieved by: linear orderings of objectives; intervals; relative ratio; or absolute ratings. These are progressively more difficult to obtain from a human expert. Forcing the latter to provide such information may yield incorrect answers if the expert simply *cannot* give this information accurately. Furthermore, as the assessment scale becomes more refined, it becomes more sensitive to noise and, consequently, more error prone.

Yager [437] proposed a new methodology that enables him to include different importances while requiring only an *ordinal* scale (i.e. a finite set of ordered symbols) for preference information. For a particular objective the negation of its importance acts as a barrier such that all ratings of alternatives that are below that barrier become equal to the value of the barrier. The motivation is that the implication $a \Rightarrow b$ can be interpreted as a fuzzy $\neg\, a \vee b$, with negation being the fuzzy complement and '$\vee$' being a fuzzy disjunction. That is, Yager disregards all distinctions lower than the barrier while keeping distinctions above the barrier. A detailed example about selecting a car is given in Yager's article [437].

The approach by Fargier et al. [133] is similar to Yager's [437]. Fargier et al. propose to order constraints with respect to each other by giving them a priority degree. A coefficient $w \in [0,1]$ is attached to each constraint, with a higher w indicating a comparatively higher importance. These priorities are transformed, without any loss of information, into constraint satisfaction degrees, assuming (5.2):

$$\mu_{\text{soft},w}(x_1,\ldots,x_k) \;\stackrel{\text{def}}{=}\; \max(1-w, \mu_{\text{soft}}(x_1,\ldots,x_k)) \tag{5.31}$$

Fargier et al. [133] motivate their choice by arguing that this formalism ensures that w represents "the degree to which [a constraint] *must* be satisfied [... The number $(1-w)$] measures to what extent it is possible to violate [a constraint]." Figure 5.2 visualizes in the plot titled "max(c,not(w))" the relation between the unweighted constraint satisfaction $c \stackrel{\text{def}}{=} \mu_{\text{soft}}$, the priority degree w, and the resulting weighted constraint satisfaction score $\mu_{\text{soft},w}$. We observe that for the case $w = 1$, $\mu_{\text{soft},1} = \mu_{\text{soft}}$. This means that $w = 1$ is a neutral priority degree, a reasonable feature shared with other schemes to implement priorities between constraints. Yager's [437] approach is slightly different in that he does not use numbers but a finite set of ordered symbols to represent the weights of the criteria, using the following linear ordering

$$\text{perfect} \geq \text{very high} \geq \text{high} \geq \text{medium} \geq \text{low} \geq \text{very low} \geq \text{lowest} \tag{5.32}$$

and defining the semantic of the '$\neg$' operator in an obvious way (e.g. $\neg\,\text{high} = \text{low}, \ldots$). The difference between Yager's method and the method specified by Fargier et al. is only of terminological nature when a 'suitable' fine-grained linear ordered set of symbolic priority values and the minimum operator are used. Other aggregation operators may produce intermediate values that are not necessarily among the linear ordered set characteristic for Yager's method, therefore making Fargier et al.'s more universal. This is largely dependent on the problem parameters, for instance how many criteria must be weighted and aggregated. Since the two models are so similar, the tutorial example serves again to exemplify their mechanics by use of the method given by Fargier et al. Results of applying some weight vectors consistent with the ordering given in (5.29) to the solution vectors from (5.27) are presented in Table 5.2.

The ranking shown for instance in (5.33) is computed through the following steps:

1. The weight vector $\vec{w}_{5.33} = (1/2, 1, 3/4, 1/2)^\top$ is applied to the respective unweighted solution vector $\vec{u}_j$ from (5.27). The resulting values are those written in the cells of the array (5.33), always being associated with one solution vector $\vec{u}_j$-row and one constraint relation R_i-column taken from (5.4).

2. These weighted evaluations of the individual constraints are aggregated by use of any 'and'-like operator, for instance the $\text{OWA}_{\text{soft\_and\_4}}$ operator as defined in (5.25) and (5.26). These aggregated values are not included in Table 5.2 as they depend upon the operator actually used. Since the resulting ranking is — with some important exceptions discussed below — independent of the 'and' operator used, only the result of the next step is included in the cells of the array (5.33).

3. The aggregated evaluations of each solution vector are used to rank them as indicated in the last column.

All the rankings found in Table 5.2 are consistent with the ranking relation given in (5.30), because all weight vectors are consistent with the priority ordering of the constraints as defined in (5.29). Additionally, all of them definitely resolve the ambiguity that remained in (5.30). Note that when using the T_M or the T_L operators, further disambiguation with the non-numeric (inclusion-based and lexicographic-based) ordering methods proposed by Fargier et al. [133] may be necessary. Avoiding these operators therefore helps to speed up the search for the best solution.

(5.33)	R_1	R_2	R_3	R_4	rank
$\vec{w}_{5.33}$	1/2	1	3/4	1/2	
$\vec{u}_1$	0.5	0.5	0.5	0.5	III
$\vec{u}_2$	1	1	0.5	0.5	II
$\vec{u}_3$	1	0.5	1	1	I
$\vec{u}_4$	0.5	1	1	0.5	II
$\vec{u}_5$	0.5	0.5	0.5	0.5	III

(5.34)	R_1	R_2	R_3	R_4	rank
$\vec{w}_{5.34}$	0	1	1/2	0	
$\vec{u}_1$	1	0.5	0.5	1	III
$\vec{u}_2$	1	1	0.5	1	II
$\vec{u}_3$	1	0.5	1	1	II
$\vec{u}_4$	1	1	1	1	I
$\vec{u}_5$	1	0.5	0.5	1	III

(5.35)	R_1	R_2	R_3	R_4	rank
$\vec{w}_{5.35}$	0	1	3/4	1/4	
$\vec{u}_1$	1	0.5	0.5	0.75	IV
$\vec{u}_2$	1	1	0.5	0.75	III
$\vec{u}_3$	1	0.5	1	1	II
$\vec{u}_4$	1	1	1	0.75	I
$\vec{u}_5$	1	0.5	0.5	0.75	IV

(5.36)	R_1	R_2	R_3	R_4	rank
$\vec{w}_{5.36}$	0	1	2/5	0	
$\vec{u}_1$	1	0.5	0.6	1	IV
$\vec{u}_2$	1	1	0.6	1	II
$\vec{u}_3$	1	0.5	1	1	III
$\vec{u}_4$	1	1	1	1	I
$\vec{u}_5$	1	0.5	0.6	1	IV

(5.37)	R_1	R_2	R_3	R_4	rank, operator dependent	
$\vec{w}_{5.37}$	0.29	1	3/4	0.29	T_P	$\text{OWA}_{\text{soft\_and\_4}}$
$\vec{u}_1$	0.71	0.5	0.5	0.71	IV	IV
$\vec{u}_2$	1	1	0.5	0.71	III	III
$\vec{u}_3$	1	0.5	1	1	II	I
$\vec{u}_4$	0.71	1	1	0.71	I	II
$\vec{u}_5$	0.71	0.5	0.5	0.71	IV	IV

Table 5.2: Some rankings of solutions with weighted constraints according to (5.31) depicted in Figure 5.2 in the plot titled "max(c,not(w))" for the tutorial example. The numbers to the right of $\vec{w}_i$ are the weights associated with the relations from (5.4). The numbers to the right of the solution vectors $\vec{u}_j$ below the weights are the scores of the weighted constraints. The rankings are based on the aggregated evaluations for the corresponding instantiation vector $\vec{u}_j$ (their calculation is left as an exercise for the interested reader), which are independent of the 'and' operator used, with the notable exception of the solutions in (5.37). Note that when using the T_M or the T_L operators, further disambiguation with the non-numeric (inclusion-based and lexicographic-based) ordering methods proposed by Fargier et al. [133] may be necessary. Avoiding these operators therefore helps to speed up the search for the best solution.

The ranking found in (5.33) is identical to the one found in Section 5.5, implying that the weights assigned to the constraints could not outweight the advantage of the solution vector $\vec{u}_3$ that satisfied perfectly a larger number of constraints. Note also that solution vector $\vec{u}_3$ can take rank I in (5.33), rank II in (5.34), or rank III in (5.36), depending on the weights used, each ranking being consistent with (5.30).

Case (5.37) is particularly interesting as it emphasizes the importance of fine tuning the weights for a particular aggregation operator: Depending on the operator used, solution vector $\vec{u}_3$ and $\vec{u}_4$ switch ranks I and II. The explanation for this somewhat erratic behavior is that the 0.71 score is just between the fixpoints of the aggregation operators for $\vec{u}_3$ and $\vec{u}_4$. A fixpoint of an operator is a value for which the operator will yield the same satisfaction scores for both solution vectors, thus ranking them identical. Since those fixpoints are different ($\sqrt{1/2} \simeq 0.70711$ for T_P versus $5/7 \simeq 0.71429$ for $\mathrm{OWA}_{\mathrm{soft\_and\_4}}$), there is a slim interval where the two operators can specify a different ranking.

The results given in Table 5.2 show that the method introduced so far is able to represent the importance of constraints adequately. Nevertheless, there are problems with this way to implement constraints of varying importance:

- For one, $\mu_{\mathrm{soft}} = 0 \not\Rightarrow \mu_{\mathrm{soft},w} = 0$ as the latter is equal to $1 - w$. However, the implication is needed to make sure that the hard barrier introduced in Section 5.4 is never crossed when constraint relaxation occurs. Thus a violated constraint will not propagate through to the evaluation function for a complete instantiation. One solution to avoid crossing the hard barrier is to test it at evaluation time and propagate the test result as an additional boolean flag. Another solution sketched in Dubois et al. [120] and termed 'safeguarding' constraints is to add a crisp constraint (with lower importance) that makes sure that a certain hard barrier is never crossed.

- Additionally, using this type of importance characterization just allows to *decrease* importance, but never to *increase* it, as one immediately observes in Figure 5.2 in the plot titled "max(c,not(w))". This can be shown by the fact that for all $\mu, w \in [0, 1], \max(1-w, \mu) \geq \mu$, and remembering that *higher* scores influence the outcome *less* when using any 'and'-like aggregation operator, as it is usually done in multiple criteria decision making. A solution is to first set all constraint priorities to an intermediate value such as 0.5 and then to increment or decrement this value.

- The ordering behavior of the priority values is somewhat chaotic, as illustrated by the rankings in Table 5.2. The rankings seem to be sensitive to certain threshold values for the weights. Since there is no unique way to compute the weights for a given problem, there is also no clear way to relate the weights to the wishes of the field expert regarding priorities, besides experimenting and fine tuning by testing different weight vector variants. Section 5.8 indicates how this tuning can be done rationally while avoiding inconsistencies with former decisions.

Another priority-scheme that remedies the problems just mentioned is analysed in the following paragraphs. Yager suggested to use the minimum operator for aggregation and to use formula

$$\mu_{\mathrm{soft}}^{\mathrm{Yager},w}(x_1, \ldots, x_k) \quad \stackrel{\mathrm{def}}{=} \quad (\mu_{\mathrm{soft}}(x_1, \ldots, x_k))^w \tag{5.38}$$

instead of the one given in (5.31), with $w \in [0,1]$, to account for importance of a constraint. Yager notes that the importance measures can be expressed on the unit interval because of

$$\min(x_1^{kw_1}, x_2^{kw_2}) \quad = \quad (\min(x_1^{w_1}, x_2^{w_2}))^k \tag{5.39}$$

with $kw_1, kw_2 \in \mathbb{R}^+$ and $w_1, w_2 \in [0,1]$. However, Equation 5.39 is only true for the minimum operator, but it does not universally hold for other t-norms nor for the OWA operator family. Additionally, the case $\mu_{\text{soft}} = 0$ and $w = 0$ must be considered in the light of the deliberations regarding the hard barrier issue raised in Section 5.4. Setting $\mu^w_{\text{soft}}(x_1, \ldots, x_k) \stackrel{\text{def}}{=} 1$ when $\mu_{\text{soft}} = 0$ and $w = 0$ would specify that a constraint with weight zero is 'turned off'. This constraint would further have no relevance for the decision process, even when it is totally violated[3]. Therefore, (5.38) is redefined for the general case and $w \in \mathbb{R}^+$ such that:

$$\mu^w_{\text{soft}}(x_1, \ldots, x_k) \quad \stackrel{\text{def}}{=} \quad \begin{cases} 1 & \text{if} \quad \mu_{\text{soft}} = 0 \text{ and } w = 0 \\ (\mu_{\text{soft}}(x_1, \ldots, x_k))^w & \text{otherwise} \end{cases} \tag{5.40}$$

Figure 5.2 visualizes in the plots titled "Yager" the relation between the unweighted constraint satisfaction $c \stackrel{\text{def}}{=} \mu_{\text{soft}}$, the priority degree w, and the resulting weighted constraint satisfaction score μ^w_{soft} for $w \in [0,1]$, $w \in [0,2]$, and $w \in [0,10]$. Note that μ^w_{soft} is not differentiable at the point $\mu_{\text{soft}} = 0$ and $w = 0$ (because 0^0 is undefined), but is artificially set to 1 for that point; it is 1 for all points on the line $w = 0$, though it is 0 for all other points on the line $\mu_{\text{soft}} = 0$. This is a natural behavior: One can expect μ_{soft} to change more often than w, because w is usually set for a particular constraint for as long as one believes that the constraint should have that particular importance, i.e. often for the whole lifetime of the constraint. Setting the priority of a constraint to zero implies that it should be turned off, i.e. neglected for the decision making process. If an OWA-type operator is used for aggregation, the constraints having weight zero should not be included in the aggregation process since they otherwise influence the outcome inappropriately. In all other cases, $\mu_{\text{soft}} = 0$ should imply that the corresponding constraint is 100% violated and the corresponding instantiation does not belong to the solutions of the soft constraint satisfaction problem.

Another property worth noting is that μ^w_{soft} is linear and equal to μ_{soft} for $w = 1$, which means that weight 1 is a neutral importance value, as one observes on the plot with $w \in [0,1]$ titled "Yager" in Figure 5.2. Additionally, for all $w < 1$, $\mu^w_{\text{soft}} > \mu_{\text{soft}}$. A smaller importance factor than 1 increases the membership function of the prioritized constraint and therefore makes it less constraining for 'and'-like operator based decisions. Conversely, for all $w > 1$, $\mu^w_{\text{soft}} < \mu_{\text{soft}}$. A larger importance

[3]Note however that contrary to how it is defined in (5.40), it is possible to set $\mu^w_{\text{soft}}(x_1, \ldots, x_k) \stackrel{\text{def}}{=} 0$ when $\mu_{\text{soft}} = 0$ and $w = 0$. In that case, even when setting the priority of a constraint to zero, this constraint could still be absolutely violated, and the instantiation would not be a valid solution.

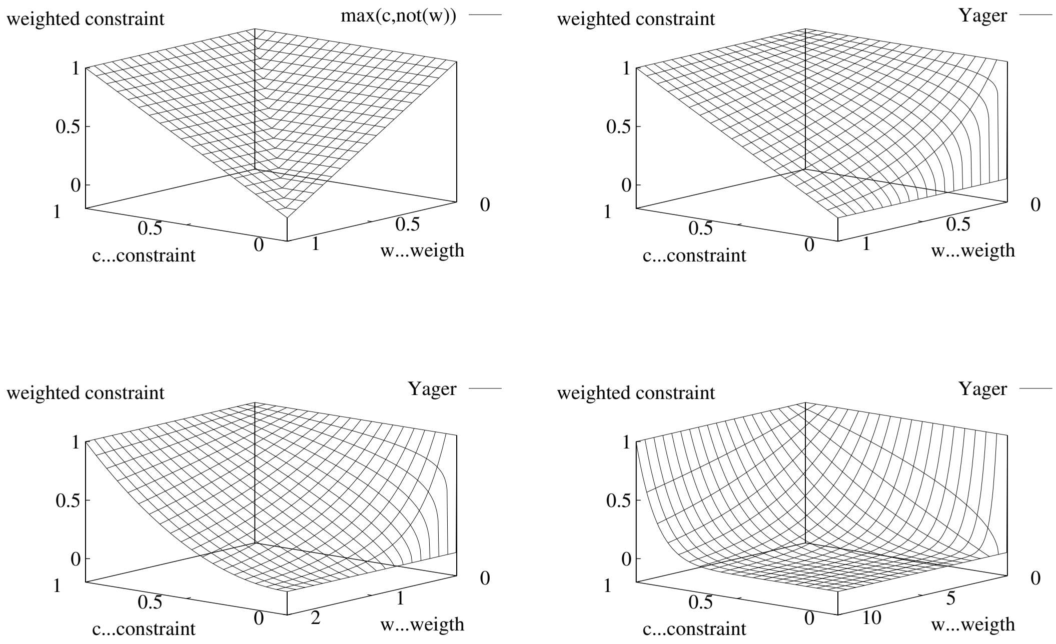


Figure 5.2: Satisfaction taking into account priority according to Definition (5.31) in the plot titled "max(c,not(w))", and according to Definition (5.40) in the plots titled "Yager", with $w \in [0,1]$, $w \in [0,2]$, and $w \in [0,10]$. The methods are respectively applied in Table 5.2 and in Table 5.4 to the tutorial example.

factor than 1 decreases the membership function of the prioritized constraint and therefore makes it more constraining for 'and'-like operator based decisions. In the next section, the μ^w_{soft} weighting scheme is applied to the tutorial example.

5.7 How to find the importance of constraints?

We demonstrated in the previous section the usefulness of importance factors to model multiple criteria optimization problems. The question nevertheless remains how to correctly ascertain the importance factors on a ratio scale. A procedure developed by Saaty [334] and restated by Ibrahim and Ayyub in [193] can be applied to compute the absolute importance factors for each constraint from paired comparisons between constraints. When the absolute importance factors for m constraints should be computed, these m constraints are pairwise compared with each other, their relative importance being written in an m-by-m matrix

$$A = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1m} \\ a_{21} & 1 & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & 1 \end{pmatrix} \tag{5.41}$$

where a decision maker judges that constraint C_i is a_{ij} times more important than constraint C_j. In order to guarantee that matrix A is 'self-consistent', only $(m-1)$ 'logically independent' pairwise comparison statements are collected to construct it. The rest of the values follows by applying

$$\forall\, i,j,k \in [1,m] :\ a_{ij} = 1/a_{ji} \ \text{ (therefore } \ a_{ii} = 1) \ \text{ and } \ a_{ij} = a_{ik}a_{kj} \tag{5.42}$$

The term 'logically independent' means that it should not be possible to compute any of the comparison statements given by the decision maker through the construction rules from (5.42) applied to the other comparison statements given by the decision maker. That is, the graph, with node-set $\{1,\ldots,m\}$ and edges (i,j) for every comparison of C_i with C_j, should be a connected tree with edgeweights a_{ij}. The full matrix A can then be defined by calculating each a_{ij} as the product of the weights along the unique path between nodes i and j in this tree.

The eigenspace $E_{\max} = r(e_1,\ldots,e_m)^\top$, with $r \in \mathbb{R}$, corresponding to the maximum eigenvalue $\lambda_{\max}$ of A, 'normalized' to $\widehat{E}_{\max} = (\widehat{e}_1,\ldots,\widehat{e}_m)^\top$ such that $\sum_{k=1}^m \widehat{e}_k = 1$, and finally multiplied with m is then the searched absolute weight vector $\vec{w}$ for the constraints:

$$\vec{w} = m\frac{E_{\max}}{\sum_{k=1}^m e_k} \tag{5.43}$$

The weighting makes sure that when all constraints are equally important, they are all weighted with 1, which means that $\mu^w_{\text{soft}} = \mu_{\text{soft}}$. A simplification following from the construction rules (5.42) of a self-consistent matrix A is that $\lambda_{\max} = m$. To calculate the corresponding eigenspace $E_{\max}$, one should solve the equation

$$AE_{\max} = \lambda_{\max}E_{\max} \tag{5.44}$$

Actually, another simplification following from the construction rules (5.42) of a self-consistent matrix A is that any column of matrix A generates $E_{\max}$. Therefore, $E_{\max}$ has not actually to be computed by any algorithm such as the Gaussian elimination procedure. This becomes totally clear when interpreting the elements of the jth-column as "how many times more important constraint C_i (of row i) is compared to C_j," as defined above, so that the jth column vector is a_{1j} times the first column vector. Therefore, the whole procedure can be simplified considerably compared to the one given in [193] since Equation (5.44) can be completely neglected. It is only necessary to construct a self-consistent matrix A as explained above and then to

relative importance	human-readable definition
1	equal importance
3	weak importance of one over another
5	essential or strong importance
7	very strong or demonstrated importance
9	absolute importance
2, 4, 6, 8	intermediate values between two adjacent scale judgments

Table 5.3: Relative importance attributes proposed by Saaty [334]. A human expert can specify weights between constraints, using the informal descriptions in the column on the right.

compute $\vec{w}$ out of e.g. the first column of A. Instead of constructing A completely, it also would suffice to identify one least important constraint C_k, and then to specify the relative importance a_{ik} of the others compared to C_k.

Saaty [334] proposes to use importance attributes as described in Table 5.3, "due to the human ability to make effective quantitative distinctions [only between] five attributes [...] Compromises between attributes can be used where greater precision is needed." Because $\vec{w}$ is the product of a normalized eigenvector of A and the number of constraints, it is relatively independent of the ad-hoc numbers given in Table 5.3.

The tutorial example illustrates the details of the procedure of Saaty [334]. Assume for instance that according to a specialist, the weights are assigned to the constraints established in (5.4) using Table 5.3 such that C_2 is absolutely (9 times) more important than C_1; C_3 is essentially (5 times) more important than C_1; and C_4 is equally (1 time 'more') important as C_1. This gives us the first column of matrix A, which is enough to set $E_{\max} = (r, 9r, 5r, r)^\top$, with $r \in \mathbb{R}$, therefore $\widehat{E}_{\max} = (1/16, 9/16, 5/16, 1/16)^\top$ and $\vec{w} = (1/4, 9/4, 5/4, 1/4)^\top$. Applied to the solutions of (5.27), the values given in (5.45) are computed using these weights and the weighting scheme from (5.40), and displayed in Table 5.4 up to two radix places. The resulting ranking corresponds well with the intuitive priorities assigned to the constraints using the values from Table 5.3. Note that when using t-norms for aggregation, further disambiguation with the non-numeric (inclusion-based and lexicographic-based) ordering methods proposed by Fargier et al. [133] may be necessary, as before. The $\mathrm{OWA}_{\mathrm{soft\_and}}$ operators immediately compute the correct ordering.

The problems that existed with the weighting scheme of (5.31), namely that the hard barrier could be crossed, that importance could not be increased, and that the ordering behavior was not smooth, are solved by the weighting scheme proposed in (5.40). Nevertheless, as the example of (5.46) illustrates, weights make sense only in combination with a previously-known aggregation operator. The weight vector $\vec{w}_{5.46}$ in Table 5.4 was computed by normalizing and then multiplying with $m = 4$

(5.45)	R_1	R_2	R_3	R_4	rank
$\vec{w}_{5.45}$	1/4	9/4	5/4	1/4	
$\vec{u}_1$	0.84	0.21	0.42	0.84	IV
$\vec{u}_2$	1	1	0.42	0.84	II
$\vec{u}_3$	1	0.21	1	1	III
$\vec{u}_4$	0.84	1	1	0.84	I
$\vec{u}_5$	0.84	0.21	0.42	0.84	IV

(5.46)	R_1	R_2	R_3	R_4	rank, operator dependent	
$\vec{w}_{5.46}$	8/11	16/11	12/11	8/11	T_M	$\text{OWA}_{\text{soft\_and\_4}}$
$\vec{u}_1$	0.60	0.36	0.47	0.60	IV	IV
$\vec{u}_2$	1	1	0.47	0.60	II	III
$\vec{u}_3$	1	0.36	1	1	III	I
$\vec{u}_4$	0.60	1	1	0.60	I	II
$\vec{u}_5$	0.60	0.36	0.47	0.60	IV	IV

Table 5.4: Some rankings of solutions with weighted constraints according to (5.40) depicted in Figure 5.2 in the plots titled "Yager" for the tutorial example. The numbers to the right of $\vec{w}_i$ are the weights associated with the relations from (5.4). The numbers to the right of the solution vectors $\vec{u}_j$ below the weights are the scores of the weighted constraints. The rankings are based on the aggregated evaluations for the corresponding instantiation vector $\vec{u}_j$ (their calculation is left as an exercise for the interested reader).

the weight vector $\vec{w}_{5.33}$ from Table 5.2, to bring it into accordance with the weighting scheme proposed by Saaty [334]. The examples (5.46) and (5.33), as well as the example (5.37) demonstrate that fine tuning of the weights for a prespecified ⟨aggregation-operator, weighting-scheme⟩ combination is absolutely necessary in order to obtain meaningful results. It is possible to completely change the ranking behavior of weights by switching to another aggregation operator or to a different weighting scheme.

5.8 A consistency test for configuration changes

An answer to the problem of making sure that fine tuning is done consistently with earlier decisions is to adopt a consistency test for configuration changes. Such configuration changes could be changes in the priorities between constraints, adopting a new aggregation operator, changing hard barriers, changing membership functions, or changing the logical structure of constraints. Basically, this change together with the test produces a new ranking for a given set of new instantiations, while observing predefined rankings for a set of old reference ranking of pairs of instantiations. The mechanism works such that, if the human expert is dissatisfied with a ranking produced by the system, the human expert can slightly change the weights of some constraints, or the exact form of some membership function (e.g. to specify that the hard barrier is actually located slightly higher), or any other parameter of the problem, such as the aggregation operator used. A consistency test will then check whether the new configuration is consistent with the rankings for a set of reference pairs of instantiations. This is done by applying the new configuration, e.g. the set of new weights, to all the old ordered pairs of instantiations, and calculating their evaluation scores with this new configuration. If for each reference pair the order between the two reference instantiations remains unchanged, this indicates that the new configuration does not invalidate any previous reference ordering. It is compatible with all decisions made in the past that became reference ranking pairs.

If one reference ranking pair is ranked in the opposite order, this means that either the new configuration is *wrong* and has to be changed again, or that some reference ranking pairs are obsolete and should therefore be removed from the reference ranking pair database. In both cases, an inconsistency among the reference rankings and the new ranking is pointed out. This inconsistency has to be resolved such that the resulting system makes rational, predictable, understandable and self-consistent decisions. The probability that the inconsistency is due to noise in the problem description and should therefore be neglected is zero, since all reference rankings have been generated with the explicit aim to change the configuration in order to give them a certain, new order. An inconsistency can point to earlier errors in configuration changes. Since each change is done under supervision, usually by a human expert, and changes are normally only adopted with the explicit goal to produce a different ordering, the inconsistency can not be attributed to noise. Whether such a decision making behavior can be termed *objective* or *subjective* depends on other factors. However, it is usually possible to lead *several* human experts to agree on a common, undisputed subset of some reference ranking pairs of instantiations, or at least to establish several different sets that correspond to configurations which can be further characterized by (and saved for later reuse under) such names as 'risky/cost-cutting', 'highest-quality', 'observe-temporal-constraints', 'standard-mix', etc., indicating their general tendency for decision making. The corresponding last configuration is saved together with these reference ranking pairs of instantiations as one knowledge base. Of course not all intermediate stages have to be stored permanently. This permits modeling with maximal flexibility the intentions of the human expert while ensuring rational and predictable behavior after changes in the configuration.

If the new configuration is adopted, the best solution *before* making the configuration change and the best solution *after* making the configuration change become an new reference ranking pair added to the new database associated to the new configuration. In the pair, the best solution *after* making the configuration change is ranked first, and the best solution *before* making the configuration change is ranked second. All data influencing the overall decision function, such as the constraints and the additional factors introduced in Section 5.9 must be stored together with the pair to be able to apply the resulting new decision function in the old context, given the new configuration. Table 5.5 lists the consistency test in procedural form.

Human expert can specify implicitly the overall configuration of the constraints by asserting a set of 'normal' reference rankings. The easiest way to apply the heuristic that establishes consistent configuration parameters for the constraints is to let the human expert do parameter changes, and to later check them out with the introduced consistency test.

Another possibility would be that a machine learning scheme in combination with a 'teacher', being either a human expert or some objective a-posteriori meta-evaluation, could automatically establish variations to the configuration parameters. This combination should find the ranking intended by the teacher using the new configuration, i.e. solve the 'inverse' problem. The difficulty is, however, that humans easily overlook some constraints, especially when the number of constraints is large and the constraints are only vaguely defined. Therefore, the subjective 'better' ranking obtained a-priori from a human expert will often objectively not be better than the instantiation found by the system because the human expert forgot some constraints, thus forcing the system to learn suboptimal decision making. Therefore, the fine tuning scenario, where human experts repeatedly change constraint parameters such as weights by hand and then compare the respective best solutions, is much better suited to establishing the best configuration for the problem. It is however an open research problem whether this fine tuning can be fully automated when a non-human error-prone but objective, a-posteriori meta-evaluation is used, such as one guided by results of quality evaluations.

5.9 Decision function and conflict identification with DynaFLIP++

To evaluate a given instantiation of a soft constraint satisfaction problem, a decision function aggregating all the constraints with their respective priorities, using an appropriate aggregation operator and a corresponding weighting scheme, must be established. The previous sections described possible techniques that help to construct the static part of such a decision function, i.e. the general constraints that will have to be evaluated several times to compute a global evaluation score for one instantiation. Whereas the representation of these general constraints is handled with the ConFLIP++ library that will be presented in Section 5.10, we present in the current Section the DynaFLIP++ library responsible for establishing efficiently a new global constraint representation for a specific instantiation of the problem. This global constraint will result in a highly structured global constraint tree for the whole schedule. The constraint evaluation function will return the weighted global satisfaction score, based on the current schedule, the value of all open variables, and this global constraint tree. Figure 5.3 outlines the general structure of

```
boolean     consistency_test(
              current_pool,
              old_configuration,
              new_configuration,
              old_reference_ranking_pair_database) {
  new_reference_ranking_pair_database ←
    new reference_ranking_pair_database;
  new_reference_ranking_pair_database.add(
    reference_ranking_pair(
      search_solution(current_pool, old_configuration),
      search_solution(current_pool, new_configuration)));
  consistency_flag ← true;
  for each reference_ranking_pair(
             best_solution_before_change,
             best_solution_after_change)
  in old_reference_ranking_pair_database {
    if satisfaction(best_solution_before_change, new_configuration) ≤
      satisfaction(best_solution_after_change, new_configuration)
    then                                   // old ordering observed
      new_reference_ranking_pair_database.add(
        reference_ranking_pair(
          best_solution_before_change,
          best_solution_after_change));
    else {                                 // old ordering violated
      outstream << "Inconsistent ranking "
                << reference_ranking_pair(
                     best_solution_before_change,
                     best_solution_after_change)
                << " (is not added to the new database)."
                << flush;
      consistency_flag ← false;
    }
  }
  new_configuration.save(
    new_reference_ranking_pair_database.save(consistency_flag));
  return consistency_flag;
}
```

Table 5.5: Pseudo-code of a consistency test for configuration changes. The program searches a database of reference ranking pairs to determine and report all those pairs in conflict the new parameters. It also builds up a new database in case the human expert wants to adopt the change, even if it is inconsistent with some old reference ranking pairs. Thus a hierarchical tree of variants and extensions of configurations, together with the corresponding database, is built up and named interactively by the human expert. If the consistency test returned true, the new configuration is an extension, otherwise it is a variant of the old one.

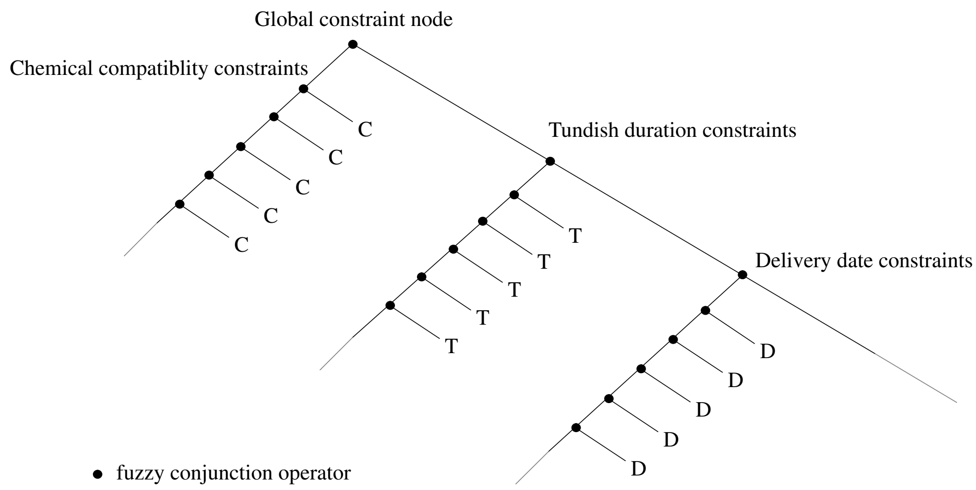


Figure 5.3: Outline of constraint tree constructed by DynaFLIP++.

this dynamically constructed tree, where the nodes are weighted aggregation operators and the leafs are ConFLIP++ objects representing individually fine-tuned static constraints. DynaFLIP++ is able to use most of the framework provided by ConFLIP++ to efficiently compute the evaluation scores for a new schedule.

When scheduling, it is often advisable to introduce an additional measure into the decision function dependent upon whether the current schedule (= instantiation of the constraint satisfaction problem) contains certain difficult jobs. If the scheduling of these jobs is not introduced as a bonus into the decision function, these jobs might never be considered for actual scheduling. There usually exists a non-empty pool of waiting jobs, and only a subset of jobs from the pool can be scheduled immediately. Therefore, the danger is that some difficult jobs will remain in the pool forever unless additional measures are taken. It is clear that this 'difficulty' or 'importance' of a job must increase over the time for which it is still reasonable to 'produce' it, to favorize its eventual scheduling. The easiest way to introduce this 'difficulty' is to formulate a corresponding constraint with an associated priority that will represent these difficult jobs. and which will therefore be represented by another branch of a certain constraint type as in Figure 5.3. Thus, the 'difficulty' of jobs will be one criteria considered when the soft constraint satisfaction problem is optimized. The same applies equally to other soft constraint satisfaction problems such as those encountered in design or planning.

To guide the search in repair based algorithms as discussed in Section 6.3, it is necessary to identify the constraint with the worst weighted evaluation, i.e. the severest conflict which can be attacked to minimize conflicts. This can be considered as a side product of evaluating the current instantiation. It corresponds to computing the evaluation using the minimum operator, and more importantly, to remember the constraint involved in the minimal weighted evaluation. This constraint represents the largest conflict for the current instantiation. Often the constraint corresponds to a general feature of the instantiation and cannot be attributed to a specific part of the instantiation. Depending on the repair operators available to the repair based constraint satisfaction algorithms, it can be helpful to find additionally the second largest and third largest conflict. Generally, the search should return the largest conflict being of a type that can be handled by an available repair operator. When DynaFLIP++ has to generate a new dynamic constraint representation for a given instantiation, it computes the individual 'leaf' constraints by calling ConFLIP++ again and again with new variable instantiations on one of the stored reference constraints, and stores the results in an intermediate form that can be used by ConFLIP++ for further aggregation. At the same time, DynaFLIP++ sorts all the computed intermediate evaluation scores, together with type information, for the later selection of 'good' repair operators.

5.10 Implementation issues and results with ConFLIP++

The reusable C++ object library ConFLIP++ is the result of the implementation efforts to realize the concepts introduced in the previous sections. It is a constraint-handling extension to FLIP++, which itself is a general purpose fuzzy logic inference process library. FLIP++ handles everything concerning fuzzification, membership functions, and linguistic variables. The user can choose between several different fuzzy inference methods, various priority schemes, different aggregation operators, and several defuzzification methods. FLIP++ also permits the graphical editing of membership functions and the easy manipulation of rule sets. ConFLIP++ itself is used as a knowledge representation tool in DéjàVu, the latter being a reusable scheduling library implementing various repair based search algorithms to compute schedules. See Section 6.4 for more information on DéjàVu. All three mentioned libraries have been developed at the Christian Doppler Laboratory for Expert Systems and are currently being further enhanced. The overall interaction model of these libraries is analogous to the layers of an onion skin, where each layer interacts only with its immediate neighbors. Additionally, all these libraries have consistent user interface methods InterFLIP++ based on the public domain wxWindows toolkit from Julian Smart of the Artificial Intelligence Applications Institute at the University of Edinburgh. XView, Motif, or Microsoft Windows surfaces can be generated by just switching compile options. The FLIP++, ConFLIP++, DynaFLIP++, InterFLIP++, and DéjàVu libraries are developed simultaneously on Sparc workstations and 386/486-PCs using Gnu-C++ as the programming language. Figure 5.4 shows

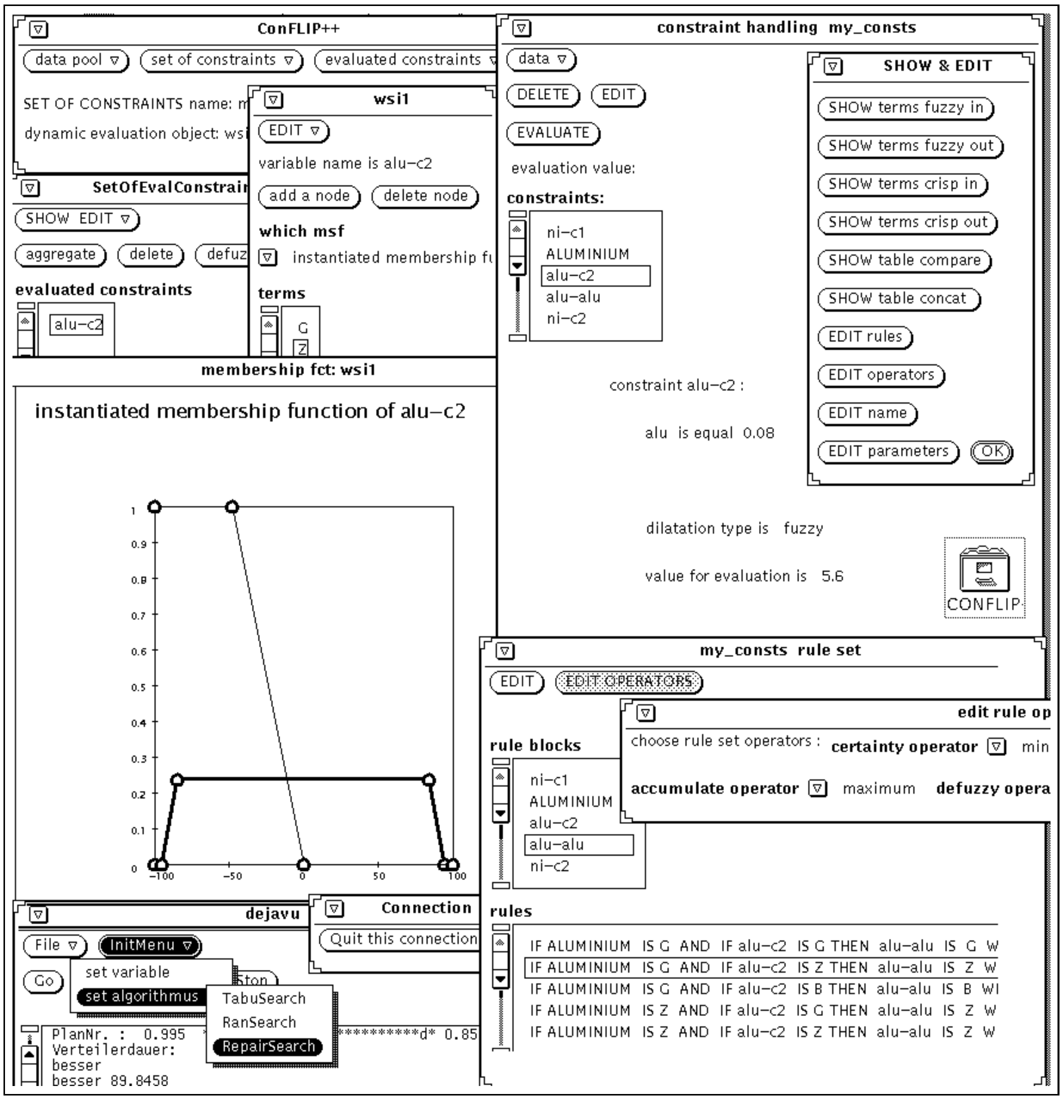


Figure 5.4: InterFLIP++ session in XView, with ConFLIP++ and DéjàVu windows.

a InterFLIP++ session in XView, with ConFLIP++ and DéjàVu windows.

ConFLIP++ allows to create, interactively edit, save and reload named sets of constraints including all parameters, and evaluate soft constraints. ConFLIP++ thus serves as a knowledge engineering tool in which domain knowledge can be stored, manipulated, and used for reasoning independently from the rest of the program using it. It is easy to add, modify, and maintain knowledge of several knowledge bases, while retaining maximal flexibility for constraints that can be adapted to model non-linear criteria used for optimization.

Using ConFLIP++, first simple constraints like

$$\text{aluminum-content} \quad \leq \quad 0.08 \tag{5.47}$$

are created and named, e.g. 'aluminum-constraint-5.47', using the objects and methods defined in ConFLIP++. The aim is to catch the vagueness in (5.47) where the $\leq$ sign is not meant to be interpreted in its strict mathematical sense, but such that 'smaller' violations are acceptable. What these 'smaller' violations could be has to be defined explicitly (and precisely) through the membership functions associated to the 'terms' of the variable as defined below. Additionally, ConFLIP++ is able to handle uncertainty about the exact value of 'aluminum-content', which is possible by propagating possibility distributions instead of defuzzified values. The operators to infer values and to aggregate several constraints are then applied to fuzzy values, which can always be represented as membership functions. This capability to model *with accuracy* vague relations and uncertain data is the major contribution of fuzzy and possibilistic logics.

In (5.47), 'aluminum-content' is a so-called *linguistic variable*, a generalization of the conventional concept of a variable. Zadeh [448] defined a linguistic variable as a quintuple:

$$\langle x, \mathcal{T}(x), U, G, \widetilde{M} \rangle \tag{5.48}$$

x is the name of the linguistic variable, e.g. the character string "aluminum-content". $\mathcal{T}(x)$ is the term set of the linguistic variable, in this case a set of several strings such as {"negative_big", "negative_medium", "negative_small", "zero", "positive_small", "positive_medium", "positive_big"}. U is the universe of discourse, here for instance the range $[0, 100]$ since 'aluminum-content' is a percentage. G is the set of syntactic rules that generate terms. These rules are mainly important when terms can be further modified by general linguistic modifiers such as 'very'. $\widetilde{M}$ is the set of semantic rules that assign meanings to the terms. These rules define the membership functions relating the instantiation of a linguistic variable to the terms in the usual fuzzy set theoretic way. For example, Figure 5.1 defines the semantics of a term such as 'About_6'. In the case of 'About_6', writing down the underlying mathematical relationship through partially linear functions would be equivalent to specifying $\widetilde{M}$ for the corresponding linguistic variable $x_3 \stackrel{\text{def}}{=} x_1 + x_2$ through the corresponding graph in Figure 5.1.

Additionally, crisp constraints are predefined as a convenience for the human expert, but are internally mapped to specially tuned soft constraints.

In the next step, several such constraints are logically combined, i.e. they are aggregated by one of the operators introduced in Section 5.5, such as for instance 'aluminum-constraint-5.47 $\wedge$ nickel-constraint', to build-up more and more complex constraints. ConFLIP++ then automatically creates a ruleset out of default or user-defined term sets for standard linguistic variables, standard rule set tables, standard membership functions for the term sets, default priority values, and various default operators using FLIP++. FLIP++ is called again later to evaluate the constraint for some instantiations of the free linguistic variables appearing in the constraint. Additionally, the system checks the scores of all constraint having a priority different from zero as well as of their constituent subconstraints before these constraints are aggregated to find out whether a hard constraint violation occurred (evaluation score equals zero) in order to invalidate instantiations that crossed the hard barrier of the corresponding constraint.

The ruleset is built for instance such that, if the first linguistic variable is compared to its term 'positive_big', and the involved inequality is 'variable $\geq$ constant', and another linguistic variable is compared to 'zero', and the constraints corresponding to the two linguistic variables are concatenated by 'or', then the resulting term for the aggregated rule is 'very_good'. The latter term comes from the predefined standard term set {"very_good", "good", "zero", "bad", "very_bad"} and its associated $\widetilde{M}$ for output linguistic variables that normally will correspond to constraint satisfaction scores.

The human expert will usually have to fine tune the automatically created ruleset and the membership functions associated to the terms. However, it is possible to store user-defined standard sets of term sets and an associated $\widetilde{M}$. Additionally, fuzzy methods are quite robust, such that the exact determination of the membership functions is not essential. The predefined triangular membership functions often perform well in a first approximation. Nevertheless, one reason that makes fine tuning necessary is that ConFLIP++ has no a-priori domain knowledge: If the constraint is 'aluminum-content $\leq$ 0.08', some generated default rules are for instance:

$$\vdots$$

IF aluminum-content *is* positive_small THEN aluminum-constraint-5.47 *is* zero
IF aluminum-content *is* positive_medium THEN aluminum-constraint-5.47 *is* bad
IF aluminum-content *is* positive_big THEN aluminum-constraint-5.47 *is* very_bad

Here the semantics $\widetilde{M}$ will be recalculated by ConFLIP++ to assign membership functions to the terms fitting the central value 0.08. However, there is no information about the point starting from which a value is very large compared to 0.08 in this context. The standard 'support' (the domain where the membership function is larger than zero) of the term 'positive_big' might for instance be preset to [0.12,100] through the standard membership function calculation. Maybe 0.09 is meant to be already very high in this context, so that even a value that is classified 'positive_small' could be very bad and therefore the constraint would not be satisfied very well and should get a 'very_bad' evaluation instead. Therefore, the standard $\widetilde{M}$, defining the membership functions associated to the terms, will often not be sufficient, and sometimes even the terms' names will have to be edited in the rule set.

The next step is the evaluation of a constraint. The evaluation happens according to the ruleset of the constraint. First, the free linguistic variables have to be given values, the latter being either defuzzified reals or possibility distributions themselves. The evaluation function returns by default a defuzzified value that describes the degree of satisfaction of the constraint with the given values.

The human expert can influence the decision making behavior of ConFLIP++ in various ways. After a constraint knowledge base has been compiled, it can be copied and the copy can then be edited. First, the human expert can select one of several aggregation, implication, and defuzzification operators. The weighting scheme can be chosen as well. Of course, the individual membership functions and priorities of the constraints can be graphically edited. For instance, it is easy to selectively edit the constraint responsible for the observation of due-dates. These changes will have immediate effect on the decision function. To ease configuration of a complete constraint knowledge base built up from scratch, the default values for all these parameters are prespecified in a way that seems to apply reasonably well to most cases. However, the human expert can later soften or harden all those constraints that have not yet been fine tuned on a individual base. In such a case, ConFLIP++ searches the complete knowledge base for membership functions of linguistic variables that have not yet been edited by the human expert. These membership functions are replaced with new standard membership functions that make the decision making behavior of the constraint knowledge base fuzzier or crisper. Additionally, the human expert can load different configurations that were constructed earlier, as described in Section 5.8. These configurations can be saved as files from ConFLIP++ together with the database for reference rankings of pairs of instantiations needed for the consistency test described in Section 5.8. The test itself is available as a method of the ConFLIP++ library, and gets automatically invoked whenever changes in the configuration are made.

5.11 Conclusion

We developed fuzzy constraints for real-world multiple criteria decision making, with a bias towards scheduling problems. We presented improved methods for compromising between antagonistic criteria, for assessing priorities among fuzzy constraints, as well as a new method for ensuring consistent and reasonable changes in configurations.

Chapter 6

Fuzzy multiple criteria optimization

Dad: *Son?*
Mooki: *What Dad?*
Dad: *I've got some advice for you.*
Mooki: *What's that Dad?*
Dad: *Do the right thing.*
Mooki: *Do the right thing?*
Dad: *Yes.*
Mooki: *That's it?*
Dad: *That's it.*
Mooki: *OK.*

Spike Lee, Do the Right Thing

A good plan today is better than a perfect plan tomorrow.

Patton's Law, Murphy's Law Complete

In this Chapter, the methods introduced in Chapter 5 are extended by iterative improvement repair based heuristics needed to deal with complex real-world multiple criteria optimization problems, similar to the one described already in Chapter 3. Here, we describe the more mature implementation of these concepts in our heuristic repair library DéjàVu which uses the DynaFLIP++ and ConFLIP++ libraries introduced in Chapter 5. The benchmark application to compare our fuzzy constraint iterative improvement repair heuristic with constructive methods based on classic constraints is a scheduling system of a continuous caster unit in a steel plant.

6.1 Introduction

The methods introduced in Chapter 5 assume implicitly that *all* possible evaluations of instantiations can be compared among one another to find the best solution. However, in real-world problems, this search space may well be very large or even infinite when variables have continuous domains. The problem of efficiently evaluating all the alternative solutions becomes quickly intractable in such a case. The problem is not solvable in its totality in any reasonable amount of time unless some type of heuristic is applied. However, there is generally no need to really find the optimal solution for real-world problems. Instead, settling for a slightly suboptimal but good solution found using a heuristic is often acceptable, supposing that it was found by using all the intelligence, computing power, and time available for that particular purpose.

In the following sections, the techniques introduced in Chapter 5 are applied to the scheduling of a steelmaking plant: Section 6.2 looks again at fuzzy constraint satisfaction problems from the point of view of a constructive algorithm. Section 6.3 explains the general repair based strategy that has been successfully applied to problems which until recently were believed to be completely out of the reach of present day computers. Section 6.4 goes on to detail the combination of such repair based techniques with the methods introduced in Section 5.2, culminating in the DéjàVu implementation for fine-grain scheduling of a steelmaking plant.

6.2 Fuzzy constraint satisfaction problems revisited

General aspects of fuzzy constraint satisfaction problems have been discussed in Section 5.3. Most of the approaches to fuzzy constraints in the literature have used methods from operations research [462] or standard backtracking algorithms [64, 120, 133, 164]. In the backtracking case, the decision tree that must be searched grows exponentially with the problem size. It is therefore necessary to prune the search tree using heuristics. Guan [164] for instance uses a coefficient λ_0 to truncate all paths for which the satisfaction falls below λ_0, corresponding to shifting the hard barrier to more constraining values. Similar methods must be used by all constructive backtracking methods. The problem is however that no compensation with partial solutions evaluating below λ_0 can be envisaged.

Matyska [273] details a similar method for logic programming approaches using fuzzy sets as implemented through a fuzzy PROLOG interpreter, where branches below a satisfaction degree of 0.5 are pruned to make the backtracking process a little bit more efficient. Matyska [273] then developed a new logic programming system CLP(FS,R) supporting unification for finite fuzzy constraints (no continuous membership functions are supported yet) to overcome some of the problems with the fuzzy PROLOG approaches.

The problems solved by applying methods from operations research are mainly limited to linear problems or must use heuristics too, in which case the remarks made for the standard backtracking algorithms apply as well.

6.3 Fuzzy repair

In contrast, repair based heuristics (sometimes called 'iterative improvement techniques') have successfully been applied to solve problems that previously seemed intractable. They have generally scored much better than constructive approaches that start from an empty instantiation and build successively more and more complete partial instantiations, such as constructive backtracking methods. Additionally, non-linearities in the constraints pose no difficulty for repair based approaches since they can easily work with approximate models of these constraints while yielding correct results. Minton et al. [277] have analysed the solution to the problem of scheduling the Hubble space telescope, a complex task by almost any measure. They have investigated the underlying heuristics, testing it on cases such as the n-queens problem and the graph-coloring problem. These problems have served for many years as classical benchmarks to study the efficiency of new heuristics and algorithms that solve constraint satisfaction problems.

The results were very promising, as the general repair based algorithm proposed by Minton et al. [277] could find solutions in less than four minutes on a Sparc workstation 1 for the *million*-queens problem, while the best constructive approach (found in an empirical study by Stone and Stone [384] to be the most-constrained backtracking algorithm) became intractable for $n > 1000$. Figure 6.1 summarizes the collected statistics averaged over 100 runs for several different n. Minton et al. even found that their repair based method exhibits linear time and space complexity for large n. The min-conflicts heuristic combined with a repair based hill climbing heuristic specifies that, starting from an initial suboptimal solution, the system attempts to minimize the number of constraint violations after each repair step. They showed convincingly that for certain problems, the use of the additional knowledge gained from operating on complete but suboptimal solutions instead of building solutions from scratch as in constructive approaches pays off well. Such repair based heuristics perform orders of magnitude better than traditional backtracking techniques. Though repair based methods can be combined with many general search strategies, they found that hill climbing methods were especially well suited for the problems they investigated. Guan [164] benchmarked constructive backtracking algorithms on the n-queens problem modeled with fuzzy constraints, the results being in line with those found by Minton et al. [277]. Only problems with small n (up to $n = 9$) were investigated.

In general, scheduling problems appear to be excellent applications for repair based methods, because:

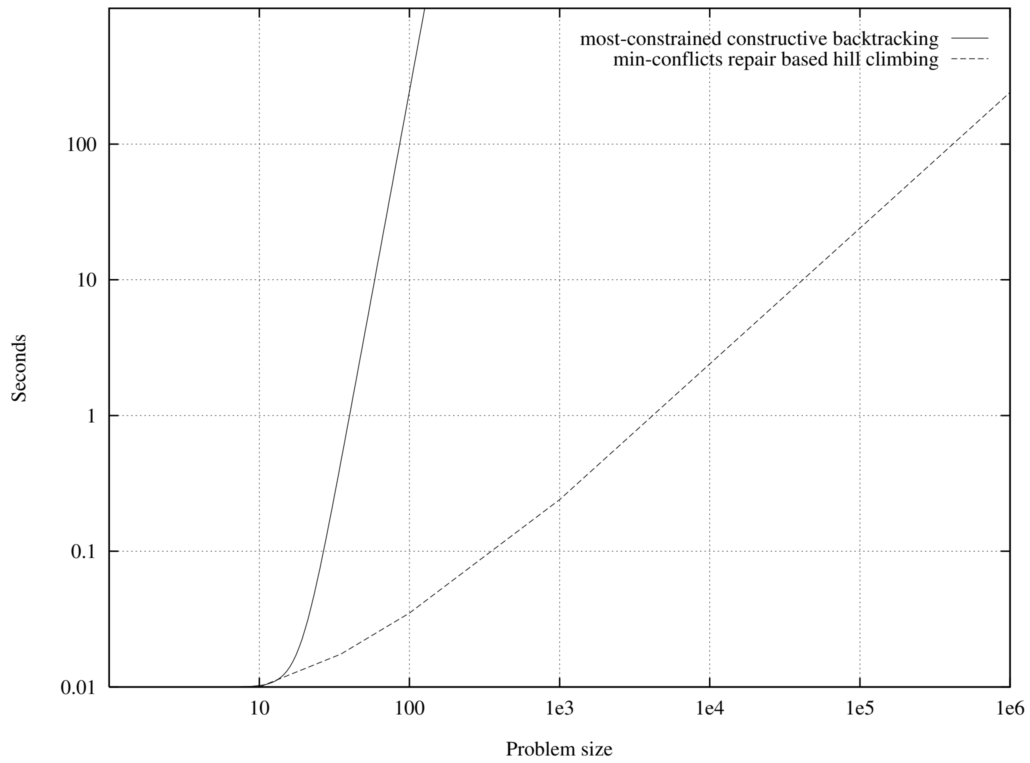


Figure 6.1: Mean solution time for min-conflicts repair based hill climbing method on n-queens problem averaged over 100 runs for several different n versus the most-constrained constructive backtracking method, the best general constructive method known (Results taken from Minton et al. [277]). Note that the repair based method exhibits linear time and space complexity for large n. The million queens problem was consistently solved in less than four minutes on a Sparc workstation 1 by the repair algorithm.

- The general algorithm is simple and therefore easy to understand and to implement.

- Repair based methods can be naturally adapted for rescheduling [465], the latter being often necessary in real-world situations because of Murphy's law, i.e. machines break down. Additionally, it normally takes much less time to change a solution than to build a new solution from scratch.

- When rescheduling occurs, the *initial* solution is normally taken to start the search for a new solution observing the additional constraints introduced by the new situation. The new solution will therefore automatically be similar to the old solution, ensuring that as few changes as possible have to be implemented. Because of this feature, the resulting overall scheduling system can be termed *robust* concerning changes. Biefeld and Cooper [28] report in this connection that human schedulers found repair based methods very natural and similar to their way of thinking.

- Repair based methods inherently posses the 'anytime' property: They return a good solution from the beginning, but can continue the search for a very good solution as long as there is time left.
- Repair based scheduling methods can easily be used interactively, sometimes termed 'cooperative scheduling', since the human expert can propose an instantiation that can be used as a starting point from which the iterative repair method resolves as many conflicts as possible.
- By combining repair based methods with concepts such as fuzzy constraints, it is easy and natural to optimize the scheduling solution while repairing the current suboptimal schedule.

Minton et al. [277] note that one limitation of the repair based hill climbing heuristic is that, as any greedy search method, it can become stuck in pathological cycles. One solution to this problem is the combination of the repair based hill climbing heuristic with the tabu list technique that was included in the DéjàVu scheduling library described in Section 6.4. The results obtained with this technique were consistently better than by using the simple repair based hill climbing heuristic alone.

More generally, Selman et al. [355] have shown that a repair based algorithm performs well on hard satisfiability problems. This indicates that the underlying technique is more useful to solve difficult problems heuristically than is generally assumed.

6.4 An application example: Scheduling a steelmaking plant with DéjàVu

A detailed description of the steelmaking scheduling problem can only be found in German language (sorry) in Dorn et al. [96], the description of the employed search algorithms and their performance in Dorn et al. [102]. There are a lot of similarities with the problem and algorithms as explained in Chapter 3, however, the details differ considerably. The main characteristics are shortly restated here to enable the reader to understand the scheduling context. Steel is produced on two continuous caster units, one being double stranded. Each day a cast sequence of about 35 charges (jobs) is scheduled for one caster over the next 24 hours. Hot pig iron is delivered from the blast furnace to the LD-converters. The steel is then poured into ladles, processed further in secondary metallurgy aggregates, and finally delivered to the casters. The casters produce continuously strands of steel that are cut into slabs of specific length. Although the steel is produced in different aggregates in a fixed flow, the casters are the bottleneck resource of the shop. The tundish, a part of the caster, has to be maintained after approximately 240 minutes. A detailed analysis of this variable's on-line determination is given in Slany [367]. The maintenance takes approximately 100 minutes, therefore a second tundish is used while the first is maintained.

The main criteria that constrain the sequence of charges are compatibility constraints between jobs. Three compatibility aspects must be considered in order to optimize production quality:

- The steel grade (the chemical analysis) of subsequent charges must be similar.
- The casting format may vary only within certain limits.
- The degassing procedure in the secondary metallurgy aggregates must be compatible between subsequent charges.

The rules are explicitly given in a crisp form, but human operators have been observed to 'break' or relax these a-priori hard rules and to implicitly make trade-offs in 20% of their decisions. Their behavior can be adequately modeled by using soft constraints with hard barriers as developed in Section 5.2.

To optimize the production, the casters should process the steel without interruption. Various operations can help when hard barriers otherwise would be exceeded. These include inserting a steel plate into the strand to separate the different qualities such that they do not mix too much, or to prematurely exchange the tundish in use with the other one. However, both operations have associated costs. If the tundish-exchange caused by quality considerations can be combined with a tundish-exchange because of tundish maintenance considerations, these costs can be minimized. The most expensive operation is to stop and then setup from scratch the whole casting unit. This takes a lot of time, but is sometimes necessary to do maintenance on the caster, or to produce certain difficult jobs.

There are soft temporal release-date and due-date constraints for certain jobs that should be delivered still hot to a subsequent plant. However, since there are (limited) warm-holding places available, the exact observance of temporal constraints is not obligatory. Additionally, the temporal constraints can be renegotiated with subsequent plants, and sometimes it is possible to exchange the temporal constraints of two jobs.

Generally speaking, it is possible to find a feasible schedule. However, criteria such as produced quality, tundish usage time, quality separating operations, number of setups and maintenance intervals of machines, release-dates, due-dates, and even such aspects as work-load of staff should all be optimized. Additionally, rescheduling has to be done quite frequently when some production parameters change due to machine breakdowns. In the past, most human errors were made in these rescheduling situations since time to think is scarce and the situation worsens rapidly (e.g. forgetting for some time a waiting aggregate, resulting in longer waiting times or worse qualities for certain jobs) if no action is taken.

In a first expert system approach, Stohl et al. [383] applied a constructive domain heuristic to the problem. Although the system found good feasible solutions, Stohl et al. believe that the solutions could be further improved, especially since constraints could only be broken through explicit user-intervention, and because the relaxing of constraints was not evaluated.

DéjàVu is a reusable library for scheduling problems implemented in C++. The name 'DéjàVu' (French for 'seen before') indicates that the methods of the library can be reused and therefore will usually be encountered more than once. So far, it has been applied to the daily fine-grain scheduling of the steelmaking plant LD3 of the VOEST Alpine Stahl AG in Linz, Austria. There are concrete plans to reuse as much as possible of the library when applying it to other problems in the future.

Several repair based algorithms were integrated in DéjàVu, namely

- a tabu list min-conflicts repair based hill climbing heuristic,
- a min-conflicts repair based iterative deepening heuristic,
- a min-conflicts repair based random search hill climbing heuristic, and
- a min-conflicts repair based genetic algorithm heuristic.

All algorithms are repair based and have several variants and different parameters. The conflict identification function explained in Section 5.9 is used together with a domain dependent repair operator library to quickly choose the repair operator that will most probably minimize conflicts for a given situation. However, the algorithms are independent of this library since the guidance provided through the conflict identification function is in all cases combined with a fallback random strategy if nothing else helps to find better instantiations [102]. Figure 6.2 shows some results for the tabu list min-conflicts repair based hill climbing heuristic versus the min-conflicts repair based random search hill climbing heuristic, for 10 samples each, with a fixed bad random initialization evaluating to 0.58139 for the chosen configuration, versus the constructive backtracking heuristic. The repair based timings were taken on a 386-PC and not on a Sparc workstation as for the constructive approach in order not to disturb the statistics by the UNIX multi-user virtual-memory management operating system.

Results using the other repair based algorithms (iterative deepening, genetic algorithm) implemented in DéjàVu so far yield intermediate results, almost equal to those found by the tabu list hill climbing heuristic. As one sees in Figure 6.2, simple random search repair is easier trapped in suboptimal solutions. Starting from a better initial instantiation (for example from the solution found with the constructive approach) does not really influence results taken after three minutes run-time. More statistics and a detailed analysis of the different algorithms can be found in Dorn et al. [102]. All repair based heuristics are much faster and yield

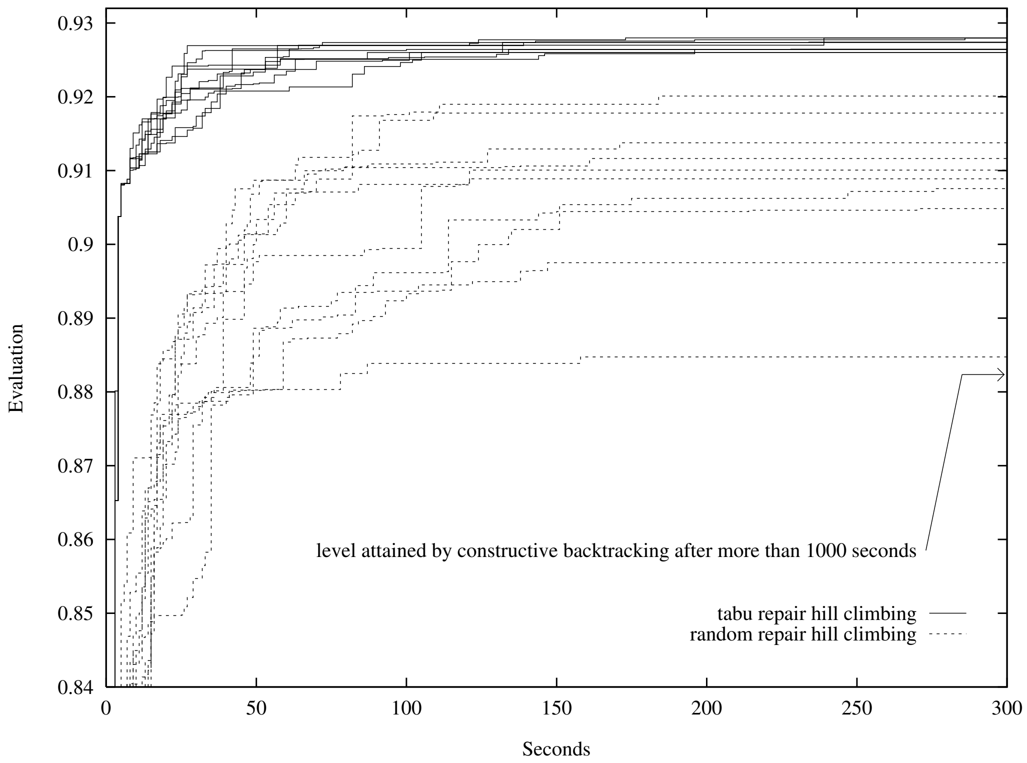


Figure 6.2: Tabu list min-conflicts repair based hill climbing heuristic versus min-conflicts repair based random search hill climbing heuristic, 10 samples each, with a fixed bad random initialization, versus constructive backtracking heuristic.

better results than the constructive approach that was evaluated using the same configuration parameters. However, the timing results are not really comparable because of different programming/run-time environments and slightly different rule sets used to search solutions in the constructive backtracking case, thus leading to slightly different optimal solutions.

6.5 Conclusion

The results obtained from the steelmaking application indicate the superiority of our approach compared to constructive non-fuzzy methods in terms of modeling expressiveness and performance.

The proposed method will be applied to other real-world scheduling problems as well as to problems of planning and design, to test the reusability, generality, and effectiveness of the implemented libraries. Furthermore, we want to refurbish the software and its documentation and distribute it as a public domain software.

We intend to further optimize the search procedure by allowing the switching of strategies when a method seems to have converged. More experiments also should be made with other aggregation operators, for instance Zimmermann's gamma operator [461].

Dubois et al. [120] introduce conditional constraints that easily fit in the ConFLIP++ framework if the need for such constraints should arise.

Chapter 7

Epilogue

The purpose of computers is insight, not numbers.
R. W. Hamming

The purpose of computers is not yet in sight.
unknown

In this Chapter we draw general conclusions about the achieved results and present interesting topics and open problems for future research.

7.1 General conclusions

We have discussed in Chapter 2 research issues and challenges in fuzzy scheduling in general, presented various approaches to the field, compared their pros and cons, and discussed some hot research topics.

In Chapters 5 and 6 we developed a combination of repair based methods and fuzzy constraints for real world multiple criteria decision making, with a bias towards scheduling problems. We presented improved methods for compromising between antagonistic criteria, for assessing priorities among fuzzy constraints, as well as a new method for ensuring consistent and reasonable changes in configurations.

The results obtained from two steelmaking applications described in Chapters 3 and 6 indicate the superiority of our approach compared to constructive non-fuzzy methods in terms of modeling expressiveness and performance. Furthermore, we have demonstrated that it is very easy to incorporate anytime and reactive scheduling features in the presented framework.

The main result of this thesis is that fuzzy constraints simplify the way reality can be modeled to represent scheduling problems in an application. As a side effect, very powerful heuristics that cut down on complexity for well behaved problems are guided by the way fuzzy constraints can be prioritized, aggregated, and finally evaluated. These methods behave in a well defined way, are mathematically sound, and intuitively easily understood.

Additionally, we have shown on the maintenance interval prediction problem described in Chapter 4 that fuzzy expert systems are a good choice regarding easiness of implementation, knowledge formulation, and in keeping the knowledge base up-to-date.

7.2 Open problems and future perspectives

One important observation is that all mechanical models can look objectively at the merits of each criterion, but it will take much more effort to build systems that mimic the human ability to reach creative agreement through negotiation. Often, invalidating a predefined problem description may yield much better overall results, as described by Fisher et al. [141]. Consider the steelmaking environment; calling the customer and renegotiating an especially difficult order was sometimes easier than to schedule that order with the original constraints, simply because the customer had different priorities than those "in the book." Another case are temporal constraints setup by other plants; talking to the responsible persons in the other plant often transforms constraints previously thought to be unmeetable but unchangeable into constraints more open to adequate solution. However, these agreements are often based on ad-hoc improvisation or human networks difficult to emulate by computer. Often, 'political' decisions play an important role in human decision making. However, it is difficult to make explicit the underlying hierarchies and dependencies, as has been empirically studied by Wagner [426]. Nevertheless, the model introduced in this thesis allows us to adequately represent all the 'objective' constraints and to make explicit any other methods that decision makers have been using sometimes unconsciously.

Annotations to the Bibliography

All the doors of this spaceship have a cheerful and sunny disposition.
It is their pleasure to open for you, and their satisfaction
to close again with the knowledge of a job well done.

Douglas Adams, The Hitch Hiker's Guide to the Galaxy

It don't mean a thing
if it ain't got that swing

Jazz lyrics by Ellington, EMI

The following annotations refer to selected items from the Bibliography starting from page 130. Order is according to the reference number in the Bibliography.

⇒Annotation [9]: Describes the language FRIL for support logic programming which is PROLOG-like. It works with support pairs and mass assignments.

⇒Annotation [12]: MASCOT and OPAL are described. OPAL uses weighted fuzzy rule combination. OPAL calculates also slack times, and this information enables the real-time level to cope with small perturbations causing delays. When significant perturbations occur, which may lead to a violation of the due dates, the OPAL system is run again, on the basis of the current workshop state and possibly new orders.

⇒Annotation [22]: Provides a careful analysis of existing literature covering scheduling under uncertainty. Berry presents her own probabilistic model permitting the computation of the likely consequence of an action. Preference Capacity Plans (PCPs) are computed using aggregate preference demand densities. Main features are

1. to focus search onto the most highly constrained parts of the problem,
2. to adjust easily to new organizational goals,
3. to give informations on the likely satisfaction of global objectives, and
4. to give information on the effect a constraint relaxation may have on the satisfaction of global objectives.

A detailed explanation can be found in Berry's Ph.D. Thesis. PCP was implemented a second time in the Distributed Asynchronous Scheduler (DAS, originally from Peter Burke and Patrick Prosser) and tested based on data from a linear flow shop manufacturing plant. The good results are due to the ability of the feedback analysis to switch the emphasis of scheduling objectives in unusual or extreme conditions.

⇒ANNOTATION [23]: Very readable analysis of what the typical real-world scheduling problems in manufacturing look-like, and how to solve them by the method proposed by the author. Berry discusses in detail the various constraints encountered in practice and the resulting difficulties. She analyses previous approaches to handle these difficulties and argues that AI approaches should be combined with techniques from decision theory and operations research. The technique proposed by the author gives the scheduling system a way to gain a global view of the situation by calculating dynamically preference capacity plans (PCP). The latter represent the probable utilization, i.e. likely demand at time t for each resource at time t given the existing preference constraints. The PCP are built by assuming every order will be completed on time and hence a predictive model based on resources of infinite capacity is constructed. Management of uncertainty is done by interpreting the probabilistic distributions as a measure of belief as strongly advocated by Cheeseman [62]. In the framework introduced by the author, preference between machines can be specified by having normalized weights for their probability distribution assignment. Preferences between constraints can be specified by altering the form of the global utility function (representing a probability distribution). In particular, kurtosis (degree to which a distribution is sharply peaked at its center) and skew (sharp abrupt borders) are given as parameters that can easily influence the outcome in either way. By analysis of the PCP, areas of high contention can be identified and adequately dealt with. A maintenance process ensures that only those parts of the PCP affected by each decision step are updated. PCP analysis can help focus the search in constraint satisfaction problems, similar to variable ordering and value ordering techniques. For the conflict case, i.e. when an operation cannot be assigned a legal start time, the scheduler can either backtrack or it can consider relaxing a preference constraint (thus changing the global utility function). The scheduling system has to balance carefully between these two options. A numerical analysis of the application of PCP in the DAS system is given.

⇒ANNOTATION [24]: The concept of emergent properties of complex systems was first observed by von Bertalanffy in the 1920s in his study of complex biological systems. He noticed that complex assemblies of entities organized in particular ways can reveal unique properties not possessed by the individual entities alone. Emergent properties cease to exist if the whole is broken into components or if the components are organized in a different way. Additionally, emergent properties cannot be understood by the study of the components in isolation.

⇒ANNOTATION [25]: Contains the first definition of computational intelligence by the editor-in-chief of the IEEE Transactions on Fuzzy Systems. In a delightful essay, he contrasts the ABC's on intelligence: artificial, biological and computational. In the strictest sense, computational intelligence "depends on numerical data supplied by manufacturers and (does) not rely on 'knowledge'." Artificial intelligence, on the other hand, uses what Bezdek calls 'knowledge tidbits'. Heuristically constructed AI such as an expert system is a typical example.

⇒ANNOTATION [37]: Contains a chapter by Hans-Peter Lipp on Fuzzy Petri Nets.

⇒ANNOTATION [38]: A well-written comparison between several plausible reasoning methods. Short introductions are given to the following probabilistic reasoning methods: Bayes's rule, modified Bayes's rule (as used in PROSPECTOR), confirmation theory (certainty factors as used in MYCIN), Bayesian belief networks, Dempster-Shafer (belief theory), evidential reasoning, and evidence space. Then, short introductions are given to the following possibilistic reasoning methods: Triangular norm based reasoning systems (Bonissone has developed himself two systems, called RUM and PRIMO, where the latter integrates monotonic rules with degrees of uncertainty as well as default values supported by nonmonotonic rules), and necessity and possibility theory. Then, short introductions are given to the following qualitative reasoning methods: Reasoned assumptions, and theory of endorsements. Bonissone then establishes a wish list for reasoning under uncertainty composed of a set of 14 requirements. The previously introduced plausible reasoning methods are then all checked against this list, and the results are given in form of a table. The article then goes on to evaluate how well these methods are suited for real-time applications.

⇒ANNOTATION [39]: Fuzzy logic controller's sensitivity to design parameters is analysed, and the fuzzy logic controller is compared with PID and optimal controllers regarding performance and robustness.

⇒ANNOTATION [40]: Refers to initial work about null-values in relational databases to represent uncertainty, then goes on to the work of Umano, Prade, and Zemankova. Interesting is an idea which leads to hyperlinks which have an associated fuzzy membership value (keyword based, with a synonym list).

⇒ANNOTATION [49]: The authors use soft constraints.

⇒ANNOTATION [51]: The authors address the problem of representing and automatically invoking error recovery sequences in response to sensed error during execution. The approach is based on the use of a fuzzy Petri net model in which sensory verification operations determine fuzzy values of tokens in the net. The outcome of a sensory verification operation changes the fuzzy values of tokens and leads to an altered firing sequence and resulting error recovery. An algorithm is described for adding sensory verification transitions and associated fuzzy transition rules which implement error recovery through retry or alternative sequence mechanisms.

⇒ANNOTATION [62]: Claim: Probability (= Bayesian) interpreted as a measure of belief is all one needs to deal with uncertainty in AI. The author explains some common misconceptions about probability, as follows.

1. 'Probability is a frequency ratio': probabilities are inherently subjective, i.e. depend on the believer's information. There is however a strong connection to frequency ratios, but only when repeatable trials are possible. The measure of belief definition of probability subsumes all others, such as frequency ratios, propensity (probability used for prediction), degree of confirmation of a hypothesis based on logical analysis, and subjective probability.

2. 'Bayesian analysis requires vast amounts of data': the available information is normally insufficient to predetermine any conditional probability, so additional assumptions have to be made. If nothing else is known, the maximum entropy assumption (such as conditional independence) represents the 'least commitment' and distributes the uncertainty as evenly as possible, thus providing a neutral background against which any non-random patterns can be observed.

3. 'Prior probabilities assume more information than given': again, Cheeseman argues that probability only represents a state of knowledge. Some prior probability assigned to a statement based on the maximum entropy assumption will change as more information is gained without inconsistency. The idea that there is a unique probability associated with a particular proposition comes from situations where all observers have the same information (e.g. physics), and so they all have the same measure of belief, assuming ideal observers. Additionally, if a problem is undefined, no theory can say anything useful about it. If probabilities are ambiguous, it is a sign that the problem is not fully defined. In practice, domain knowledge leads to non-uniform priors, even though we may be uncertain of their values (examples explained: Bertrand's paradox, ship on Atlantic).

4. 'Numbers are not necessary': using non-numerical methods such as the theory of 'endorsements', only statements of the form P_1 is-more-probable-than P_2 can be used, thus decision making is limited to choosing among alternatives, but there is no possibility to not choose at all.

5. 'More than one number is needed': the author convincingly shows that how many numbers are needed to represent uncertainty depends on the question one is trying to answer, such that approaches like Dempster-Schafer's are often an overkill, and sometimes an underkill.

6. 'Baysian does not work, so here is a new scheme!': Cheeseman argues that new theories lack a well established model theory to show how to get from data to uncertainty representation and then map it into a well defined decision theory. Certainty factors as used in MYCIN are given as a first example, and Cheeseman explains how prior probabilities were misrepresented, thus a different theory was apparently needed. The author concedes that priors are often only very subjective estimates of experts, that people are not good at providing those estimates, and therefore that real data should be used to calculate further probabilities (which then are not very sensitive to the exact value of priors). Cheeseman goes on to argue that fuzzy sets, fuzzy logic and possibilistic logic are unnecessary since a theory of probabilistic set membership can be established, capturing the vagueness idea. He further states that the motivation underlying these theories is based on the fallacy that probabilities are necessarily frequencies. He nevertheless agrees that historically it has often not been obvious how to apply prior probabilities in a particular situation, thus complicating the task of representing a problem with probabilities.

7. Some other mistakes are among others that probability and utility (importance) of an hypothesis are confused in some uncertainty theories, or that probability is mistaken as a special case of logic. Indeed, logic is only an approximation of probability, and it is necessary to threshold probability values if they are 'beyond reasonable doubt' (otherwise no logical reasoning is applicable to the real world), thus allowing logical reasoning, which is easier to use than probabilities.

Cheeseman goes on to argue that default logic and non-monotonic logic often force into a logical mold a type of reasoning that is not logical in nature: One standard example of default reasoning "all birds fly unless proved otherwise" should be in his opinion "most birds fly", which can be used as a piece of evidence in evaluating the probability of the proposition "this bird flies", along with any other evidence. Although often one line of reasoning will dominate the final result (often termed 'reason'), it should not be mistaken as a logical reason, since in probability, contradiction does not occur: all the evidence is taken to get a final probability value. AI systems should use both logical and probabilistic reasoning where applicable.

$\Rightarrow$ANNOTATION [64]: Description of how to compound conflicting scheduling objectives with fuzzy sets, using the minimum operator with nonlinear membership functions. The search procedure of a previous system called FPS uses heuristic hill-climbing. The new system BFPS described in this paper uses various heuristics to prune the backtracking search space. Objectives which are more closely looked at are 'minimizing total overdue time' and 'minimizing total number of overdue parts'. The different objectives are combined via the min-operator. Runtime results for four algorithms and four different problems are given.

⇒Annotation [68]: Fuzzy sets are used to model the subjectivity of process planners and vagueness of data. Uses fuzzy priorities for criteria and compensatory aggregation operators. The mean flow time of parts is minimized. Scheduling is done dynamically at process time, taking into account process status and predefined partial process plans.

⇒Annotation [70]: Well written introduction. AI systems design guidelines for modeling the imprecision, ambiguities, and undecidability of most real-world phenomena:

1. if you know the rules, you should use fuzzy logic,
2. if you don't know the rules, you should use neural networks, and
3. if you have lots of historical data, you should use case-based reasoning.

Discusses fuzzy neurons and fuzzy backpropagation.

⇒Annotation [91]: Mathematical-analytical methods are often insufficient for scheduling applications. This is due to three reasons: The uncertainty in the production process, combinatorial complexity of the search space, and conflicting objectives for production optimizing. Knowledge-based techniques, especially approximate reasoning and constraint relaxation, are promising candidates to solve these problems. The authors use a case study to demonstrate how knowledge-based scheduling with the desired capabilities could work. The applied knowledge representation technique covers the uncertainty inherent in the problem domain by using fuzzy set theory. Based on this knowledge representation, the importance of jobs is defined. This classification of jobs supports the straightforward generation of a schedule. The authors introduce a control strategy which comprises several types of constraints, namely organizational, spatial, and chemical ones. This strategy supports the dynamic relaxation of conflicting constraints in order to improve the schedule.

⇒Annotation [105]: Section 1.2.3 (p.15) is devoted to Fuzzy Planning and Fuzzy Scheduling.

⇒Annotation [120]: Actually composed of two reports by the authors, "Handling priority and preference in constraint satisfaction problems" and "The use of fuzzy constraints in job-shop-scheduling", the latter being also included in the working papers of the IJCAI Workshop on Knowledge-Based Production Planning, Scheduling and Control, Chambéry, 1993, pp. 101–112

⇒Annotation [128]: The paper is not badly written and even won a best-paper award at the conference. However, it is based on the wrong assumption that in fuzzy logic the rule of the excluded middle is still valid, which of course is not true since the middle is what we want to model with fuzzy logic. A very well-written refutation was given by Enrique Ruspini, plus another one by Didier Dubois and Henri Prade, and both can be found in the archives of the comp.ai.fuzzy news-groups or at the indicated ftp site. There is a plan to publish a modified version of Elkan's AAAI paper together with a number of invited commentaries in a future issue of IEEE Expert.

⇒ANNOTATION [133]: The article from the session on fuzzy constraint propagation chaired by Henri Prade deals with handling inconsistent fuzzy CSP, where complete instantiations shall be ordered according to their satisfaction degree. The so called 'drowning effect' expresses the fact that in the conjunctive combination approach only the most important violated constraint is relevant for ordering the instantiations. The authors show two different ways for selecting preferred solutions with relevance extended to all constraints. The 'inclusion-based preference' method compares satisfaction degrees of instantiations pointwise (constraint by constraint), retaining the lowest satisfaction degree among points with different satisfaction degrees. The 'cardinality-based preference' method ranks the constraint satisfaction degrees of an instantiation increasingly and then orders the instantiation by the well-known lexicographic ordering. The lexicographic criterion is more selective than the inclusion-based criterion, which is itself more selective than the conjunctive combination.

⇒ANNOTATION [138]: Deals with the issue of fuzzy nonlinear programming under fuzzy constraints.

⇒ANNOTATION [139]: The first chapter contains statistics about software production costs. For instance, 82% of the maintenance costs arise due to insufficient specifications.

⇒ANNOTATION [141]: Autors are directors of the Harward Negotiation Project at Harward University Law School. Important points are:

- Don't bargain over positions
- Separate the people from the problem
- Focus on interests, not positions
- Invent options for mutual gain
- Insist on objective criteria
- Develop your BATANA, the best alternative to a negotiated agreement.

⇒ANNOTATION [145]: Fuzzy constraint based approach to determine the arrival and starting times of trains at stations.

⇒ANNOTATION [147]: Calculating the Hamacher sum for arbitrary gamma values seems to be a very complicated process. Summarizing symmetric triangular fuzzy numbers with different width seems to be a helpless task as well.

⇒ANNOTATION [148]: Problems with traditional process control techniques in achieving control objectives are discussed and modeled. It is concluded that the difficulty in formulating a concise mathematical model for control systems is to be avoided and instead a 'mental model' with fuzzy control should be used. This is exemplified with the design of a control system for an incinerator. In another example mathematics, physics, and expert system know-how are combined in the design of control of automatic trains. As an example of the application of expert knowledge to the control of a distributed process the control of a billet foundry smelting line is treated. Expert system tools are described and speedy diagnostic theory procedures are examined.

⇒ANNOTATION [149]: This paper describes an acquisition of scheduling knowledge. The scheduling problem studied in this paper is to decide priorities of assembling products. The problem often occurs in workshops. Knowledge acquisition has been one of the most important problems in constructing expert systems. Some methods for acquiring fuzzy rules, such as fuzzy neural network and genetic algorithms, have been proposed. This paper studies the application of a Fuzzy Classifier System (FCS) to knowledge acquisition of deciding properties of assembling products. The knowledge is represented by a set of fuzzy rules in FCS. This method can handle multi-parameters and multi-constraints, and can acquire fuzzy rules easily. The feasibility of this method is verified by using practical problem in a workshop.

⇒ANNOTATION [150]: Solves a fuzzy linear programming problem which has fuzzy numbers as coefficients in its objective and constraint functions. Fuzzy numbers treated are L-fuzzy numbers. The authors introduce a partial order relation among the fuzzy numbers and show that the fuzzy linear programming problem with respect to the partial order is equivalent to the multi objective clisp linear programming problem with a certain dominated cone. Furthermore, methods to solve the induced clisp problem are given.

⇒ANNOTATION [160]: The authors describe fuzzy compromises between dispatching rules. An industrial scheduling software called SIPAPLUS has been modified in order to use such compromises, and some results are discussed.

⇒ANNOTATION [161]: Compromises may be tuned in order to provide a schedule adapted to the workshop objectives. Three methods to tune theses compromises are tested in this paper: design of experiment, fuzzy expert system, and neural network. This last approach seems to provide the best result.

⇒ANNOTATION [164]: The fuzzy constraint satisfaction problem solving techniques discussed are all based on backtracking algorithms and the minimum operator. A minimal satisfaction index λ_0 cuts-off search. Criteria have all the same importance, and fuzzy numbers are proposed for uncertain data. Chapter 5 resumes traditional constraint relaxation techniques. Actual problems considered are the N-queens problem (with $N \leq 9$) and the design of truss bridges. Small examples are compared based on the number of backtracking steps needed.

⇒ANNOTATION [172]: Optimistic and pessimistic schedules are calculated for given α-cuts. Application is the scheduling of agricultural dates. A maximal possible resource requirement analysis is done first. Uncertain time parameters are used.

⇒ANNOTATION [174]: In a cold strip mill, the shape control system is a feed back control system consisting of a shape meter and such actuators as work-rolling bending, reduction leveling and coolant equipment. The paper describes the application of fuzzy control to determine the manipulated values, which aims at smooth control of the process involving a number of nonlinear elements. In particular, the following fuzzy control is applied to determine the coolant flow rate in zones of the coolant equipment: the relationship between the amount of change in coolant flow rate and the deviation and rate of its change expressed by shape parameters as variables is represented by three rules expressed by membership functions. The amount of change in coolant flow rate is obtained from the barycenter of membership functions of the three rules. At each of the other actuators, the amount of control is determined according to its fuzzy control rules, too.

⇒ANNOTATION [175]: In recent years, the stabilization of strip shape has become increasingly important. Nippon Steel's Nagoya Works has developed a strip shape control system for its No. 3 cold strip mill. With this system, the strip shape is detected by the shape meter and feedback control is executed using the work roll bending, pressure leveling and coolant zone control. This system features the application of fuzzy control for the improvement of shape control accuracy, ensuring excellent performance not only for simple but also for complex shapes.

⇒ANNOTATION [176]: A blast furnace system based on artificial intelligence has been developed to incorporate the knowledge of expert engineers in support of the blast furnace operation. This system includes a furnace condition diagnosis system and a heat control system. The authors describe the new fuzzy method used to model vague human knowledge.

⇒ANNOTATION [177]: In the No. 5 blast furnace at the Fukayama Works, to coincide with the February 1986 reblowing-in, an expert system applying knowledge engineering was introduced to operate abnormal furnace condition forecasts and furnace heat control. The system used know-how accumulated over the years by blast furnace engineers. Using online and realtime operating, it monitors the condition of the furnace and controls it. This paper outlines the construction of the expert system and the results of its application.

⇒ANNOTATION [178]: In multi-attribute decision making, human beings influenced with various factors often change their decisions. This paper presents a new approach to identify the change in a decision making process. The new approach uses a fuzzy neural network (FNN) which has been proposed by the authors. The FNN identifies the weights to the attributes using back propagation learning. Through experiments, it is shown that the changes of subjects' decisions can be described by the changes of their weights to the attributes.

⇒ANNOTATION [179]: Well-written paper. Gives motivation why to develop fuzzy based production scheduling instead of OR or knowledge-based approach. Their methodology is divided in three parts:

1. a schedule evaluation module: based on resource utilization, estimated waiting time at a resource, early time of lots, and waiting time of jobs; min-max is used for and-or, the implication operator used is the one from Łukasiewicz; no special weights are assigned to rules, this is to be done implicitly through the design of the membership functions.
2. a scheduling policy module: finds out what is wrong with the current schedule, i.e. tries to localize 'critical' items. This results in weights assigned to priority rules, thus guiding the schedule generation.
3. a dispatching module: uses the priority rules together with the inferred weights to build a schedule

Numerical examples are given. Actual objectives were due-dates and to shorten total processing time of jobs. Additional requirements were in further strategies to utilize a certain machine as little as possible, and then to use another machine as often as possible. The authors concede that adding such objectives may influence other scheduling objectives, and that to tune up the overall objective function, the user has to adjust the membership functions of the rules.

⇒ANNOTATION [192]: The internet news-group comp.ai.fuzzy, also accessible electronically via various mailing lists and blackboards, includes a frequently-asked-questions (with answers) list, which contains pointers to the most important conferences, major journals, scientific societies, research centers, major scientific projects, book-lists, as well as names of persons-to-know and companies related to the field. The news-group is also a forum to discuss all topics related to the field, and is equipped with a searchable archive extending over several years. Recently, a hypertext (WWW) version is available too.

⇒ANNOTATION [193]: Invited review: The authors use interval analysis and logic diagram techniques to propagate uncertainty pairs (upper and lower estimates). Criteria are prioritized with the x^α method proposed by Yager, then compounded with the min operator. Assessing the αs is done by allowing $m-1$ subjective pairwise rankings between m criteria, where the ranking values can only be integers from 1 to 9, 1 meaning equal importance of one over the other, and 9 means absolute importance of one over the other. Then, a $m \times m$ matrix is constructed based on the $m-1$ rankings. The normalized eigenvector of the maximum eigenvalue of this matrix represents the αs used to weight the criteria evaluations. An example using the following criteria is given: probability of failure, magnitude of fatality, magnitude of damage, economic risk, human risk and uncertainty associated with the estimation of the consequences of failure.

⇒ANNOTATION [194]: A method of calculating a numerical solution of the inverse problem of fuzzy models is proposed. The proposed method is based on a nonlinear numerical optimization problem. For the given output value, the optimal solution, which provides the approximation of output value, is searched in the feasible region which is defined in advance. The search direction is calculated by the steepest descent method with the inequality constraint. This method provides the locally optimal solution because of the complexity of the object function. The numerical examples show the efficiency of the proposed method.

⇒ANNOTATION [195]: The machinery division of Daido Steel has developed a fuzzy controller which can be applied to various control systems for industrial furnaces. In this paper, the fuzzy controller is introduced, with the following features:

1. general purpose controller by utilizing a personal computer.
2. combination control with pid control.

The practical results of an application to temperature control for an experimental furnace are reported. It was confirmed that the developed fuzzy controller could control the temperature effectively (constant-value control) and that tuning was very easy using the fuzzy method. Using this controller, the performance of skilled operators could be attained.

⇒ANNOTATION [204]: In plate rolling, the slab sequence greatly affects product quality and productivity. A sequence obtained by procedural methods requires many corrections by the operators due to the large number of major restrictions in decision making. The expert system described in this paper can easily give operators an optimum solution by comprehending and applying their knowledge in the sequence scheduling process.

⇒ANNOTATION [211]: Definitions of alpha-cut, extension-principle (with example), fuzzy relations, decision making under fuzziness (with fuzzy constraints), and various implication operators.

⇒ANNOTATION [213]: A superior blowing control system is required for stable operation and saving manpower in LD converter processes. To investigate the possibility of future advances in blowing control systems, a prototype expert system for blowing control has been developed. This system is successful in applying the expert system to on-line real-time process control and applying fuzzy reasoning to represent the ill-structured problem. As a test result, the high blowing control ability of a skilled operator could be simulated to prove the effectiveness of applied knowledge engineering in blowing control.

⇒ANNOTATION [214]: The internet news-group comp.ai and all subnewsgroups comp.ai.* (such as genetic, vision, ...), some also accessible electronically via various mailing lists and blackboards, include several frequently-asked-questions (with answers) lists, which contain pointers to the most important conferences, major journals, scientific societies, research centers, major scientific projects, book-lists, as well as names of persons-to-know and companies related to the fields. The news-groups are also forums to discuss all topics related to the fields, and are often equipped with searchable archives extending over several years. Recently, a hypertext (WWW) version is available too.

⇒ANNOTATION [215]: Well written paper about temporal fuzzy constraint propagation. Especially the comparison with other approaches is worth reading. A profound motivation for the application of fuzzy set theory to scheduling problems is given. An actual implementation called FSS is described.

⇒ANNOTATION [224]: Well written introduction, but remains rather superficial: No concrete methods for implementing the nice features of case-based reasoning (CBR) are given. CBR is well suited for underconstrained problems like architectural design. CBR can be combined with model-based reasoning to verify the solutions proposed by the CBR system. CBR works well in domains considered as black art, i.e., there is no complete causal model of what works and why. CBR suits itself to tasks with conflicting objectives, i.e., being overconstrained. For design, usually several cases have to be combined. Large constraint satisfaction and relaxation problems can be avoided. CBR for planning works well even when several goals compete. Previously-used plans are saved and indexed by the conjunction of goals they achieve. If the conjunct of goals is repeated, the old plan that achieves them together can be recalled and repeated. CBR is well suited for reactive planning, too, since adapting and substituting semantically-similar steps for those that have failed leads to repair. Problems can be anticipated by learning from experiences. Diagnosis is a major application. Interpretation in the context of CBR means deciding whether a concept fits some fuzzy-bordered classification, which can be derived on the fly. Learning is done with successes as well as failures. CBR can be combined with rule-based reasoning, where rules are used when they match exactly. CBR is a process of 'remembering a case and adapting its solution' or 'remembering a case and evaluating the new one based on its outcome'. Major processes are case retrieval, case storage and evaluation. Two styles of case-use exist: Problem solving CBR proposes a problem, adapts, and criticizes. Interpretive CBR proposes a desired result, justifies it, and criticizes. Case retrieval is done by indexing. Choosing good indexes is one major issue in CBR. A common strategy deletes secondary components if they perform no necessary function. Behind a CBR system, a causal-model-based system ensures valid adaptations and verifies proposed solutions. CBR is useful when knowledge is incomplete and/or evidence is sparse, since descriptions of past experiences of what worked are enough to fill the knowledge base. CBR allows to build decision aiding systems that augment human memory by providing the appropriate cases while still allowing the human to reason in a natural way. People should be taught how to justify case-based suggestions and that justification or evaluation is crucial to good decision making.

⇒ANNOTATION [227]: The model of Crossing Gate Network is presented and applied to traffic planning, transportation problems, etc. The Crossing Gate Network is constructed with crossings and lines which connect crossings. Tokens move according to the transfer rules in the network and their flow is controlled by cycles, offsets, and splits on crossings. In this paper constraints are formulated and Fuzzy Linear Programming is applied to decide splits which optimize objective function. Experimental results show that the queue length decreases by the strategy.

⇒ANNOTATION [254]: In the production management of complex chemical engineering processes, the operative decisions, which have so far been made by humans, may effectively be transferred to a computer using fuzzy methods. The fuzzy concepts allow one to consider the often only linguistically expressed expert knowledge directly in the automation scheme. By the fuzzification or coarsification of process, information process models of low dimensions are obtained, enabling the effective execution of real-time decisions. A fuzzy-Petri-net scheme is presented, based on which knowledge bases for the control of complex production processes may be constructed.

⇒ANNOTATION [266]: A hot stove is a device which supplies high temperature air kept at a constant temperature to a blast furnace by using the heat regenerative function of bricks. A hot stove is required to operate at high efficiency under the condition of supplying required heat energy to a blast furnace and, in addition, protecting combustion gas flow rate. The feature of this model lies in the application of fuzzy theory. In this model, the set value of gas flow rate is calculated by means of fuzzy inference based on the evaluations of the residual heat value and brick temperature distribution. The implementation of the operation using this model resulted in an improvement in the thermal efficiency of the hot stove. The system is applied it to the No. 6 blast furnace at Chiba works.

⇒ANNOTATION [267]: In ironmaking process, Kawasaki Steel has developed a uniform burning control system in the pallet width direction of the sintering machine and a set-point control system of hot stove combustion by applying fuzzy control. In the fuzzy control system of the sintering machine, burning speed in the pallet width direction is controlled by varying filling density using 5-split sub-gates. The distribution is detected by observing the vaste gas temperature with thermometers installed in the same direction. It resulted in the improvement of uniform sintering. In the fuzzy control system of hot stove, combustion gas flow rate and calorie are controlled by observing the residual heat value and the brick temperature distribution of each stove. As a result, brick temperature dispersion has been decreased and hot stove thermal efficiency increased.

⇒ANNOTATION [268]: This editorial contains some bean counting and comparisons on neural nets, evolutionary computation, genetic algorithms, fuzzy systems and artificial life as 'Computational Intelligence' versus expert systems as 'Artificial Intelligence'.

⇒ANNOTATION [273]: CLP(FS,R) as implemented in Prolog by the author is introduced. Fuzzy Sets membership-values are implemented as singletons reals, no support for membership-functions is given. Relation to Herbrand Universe is mentioned, Fuzzy Prologs are explained (they cut-off everything below .5 to prune search subtrees when backtracking, which is still computationally inefficient and dangerous since optimal trade-off solutions might be cut-off). The author introduces FS theory in an understandable and mathematically correct way. Unification, mainly based on fuzzy sets is explained and algorithms are given for implementation. The SICStus DMCAI clone developed at Vienna University is used as a basic implementation environment.

⇒ANNOTATION [278]: Branch and bound methods are employed to solve mixed integer programming problems. However, such methods are not well suited to solve fuzzy mixed integer programming problems, because it is not specifically intended to obtain the optimum value for the objective function, but to maximize, in fuzzy programming, the degree of the satisfaction or the degree of possibility based on the formulation of the problem. In this paper, the authors propose a method based on genetic algorithm to obtain a solution which maximizes an objective function in fuzzy mixed integer programming problems.

⇒ANNOTATION [280]: As a general model of rule-based systems, the authors propose a model for a fuzzy production system having chaining rules and an inference engine associated with the model. The concept of so-called 'fuzzy Petri nets' is used to model the fuzzy production system, and the inference engine is designed to be capable of handling inexact knowledge. Fuzzy logic is adopted to represent vagueness in the rules, and the certainty factor is used to express uncertainty of each rule given by a human expert. Parallel inference schemes are devised by transforming fuzzy Petri nets to matrix formula. Further, the inference engine mechanism under Mamdani's implication method can be described by a simple algebraic formula, which makes real time inference possible.

⇒ANNOTATION [281]: At Kobe Steel, a system for blast furnace operation control which consists of an expert system to predict furnace heat and a fuzzy reasoning system which estimates furnace heat from hot-metal temperature was designed. The result of the prediction by the expert system can be evaluated and tuned by the result of the fuzzy reasoning system.

⇒ANNOTATION [282]: In this paper, the authors propose the new ranking criterions of trapezoidal fuzzy numbers (TrFNs) each of which is defined using three parameters. They show that the proposed criterions include the criterions produced by using any cutting level alpha in [0, 1] and any of the indices proposed by Dubois and

Prade. They also propose a method for solving linear programming (LP) problems with TrFN coefficients, using the ranking by the proposed ranking criterions. In this method, a decision maker (DM) can treat the constraint with more reflection of the DM's intention than in the method using the rankings by indices proposed by Dubois and Prade.

⇒ANNOTATION [284]: In this paper, the authors show some of the definitions of fuzzy inequality introduced by several researchers and consider their applications to linear programming (LP). Fuzzy linear programming (FL) problems with fuzzy coefficients in the constraint set can be transformed into conventional (nonfuzzy) LP problems using the definitions. The authors give some numerical examples and illustrate the feasible areas obtained from applying these definitions to each example in order to compare their characteristics.

⇒ANNOTATION [296]: A terrain avoidance capability is necessary for low-altitude flight. This paper presents a low altitude flight path planning algorithm based on fuzzy reasoning and its tuning method using neural network type learning. The planning algorithm repeats 3 main processes and generates the flight pass. The first process derives feature parameters from the relation between the aircraft and the terrain, the second process determines the steering based on fuzzy reasoning, and the third process updates the state variable of the aircraft dynamics. The fuzzy reasoning learns from reference paths using neural network type tuning method and masters skilled steering.

⇒ANNOTATION [297]: Interesting points: introduces the notion of bold product, residuum, and bold intersection. The exact determination of the membership function is not as important as it might seem at a first glance, since fuzzy methods are very robust. Novák points out that the usual way to interpret approximate reasoning (after Gupta and Yamakawa) is logically incorrect because the implication from logic is not symmetric but the approximate reasoning implication as usually used is symmetric. It nevertheless works in practical applications because very often the 'implication' just describes some kind of relation between input and output and is not, in fact, understood to be an implication, see also remarks made by Cheeseman [62]. He further gives an interesting and detailed analysis of the notion of a linguistic variable according to Zadeh, which captures the classic concept of variable as well.

⇒ANNOTATION [298]: Scheplan is an expert system kernel that was developed for scheduling steel-making processes. The typical constraints in such processes are a fixed sequence of production stages, no machine conflicts among products, low waiting time, continuous use of some machines, and a resting time requirement for some machines. The approach presented for designing a schedule that satisfies the constraints is not to obtain an optimal solution, but rather to obtain a feasible solution efficiently. The reason for this is that it is very difficult to define an evaluation function for the optimum, and that a combinatorial explosion may prevent a schedule from being obtained in a reasonable time. A cooperative schedule method is introduced in which the system efficiently generates a candidate schedule by a sub-scheduling and merging method, and the user evaluates and modifies the candidate schedule by interactive refinement.

⇒ANNOTATION [300]: This paper proposes a practical approach to actual scheduling problems and describes its application to creating daily schedules for steel-making processes. Cooperative scheduling is a new paradigm in which procedures, rules, and the user cooperate to make a feasible schedule efficiently. The procedures, collectively called a "scheduling engine", work as a local constraint satisfier to solve general primitive constraints. Rules that represent domain-dependent knowledge then solve the domain-specific constraints by means of a pattern-matching function. Finally, the user evaluates the schedule and modifies it via a user-friendly interface with direct-manipulation functions. The user interaction is therefore included in the system architecture as a global global constraint satisfier. The iteration of this cycle improves the schedule until it becomes feasible. Scheplan is a scheduling environment that applies this approach to scheduling steel-making processes. The system has been transferred to the Keihin plant of NKK (Nippon Kokan Co, Ltd.), one of the major steel-making industries in Japan, and is being tested and evaluated in an actual environment for operational use. According to experimental reports, the daily scheduling time is much lower than in manual scheduling. Furthermore, the quality of the schedule itself is much improved, which results in a saving of about one million dollars a year in production costs.

⇒ANNOTATION [301]: A blast furnace heat control system using neural network and fuzzy inference has been implemented in the No. 6 blast furnace in Chiba Works of Kawasaki Steel Corporation.

⇒ANNOTATION [302]: A fuzzy stochastic dynamic program results when various forms of the classical dynamic program processes such as discrete continuous, deterministic, stochastic, and adaptive ones are appropriately fuzzified.

⇒ANNOTATION [303]: There is a marked tendency directed towards a shorter period of delivery and a smaller lot of products. At NKK's Keihin Works, efforts have been concentrated on the reduction of the production scale in compliance with the medium term management plan and establishment of a flexible higher efficiency operating scheme with a single blast furnace. The new total production control system has been developed primarily to provide a balance between the problems of delivery time and cost, while securing maximum possible output.

⇒ANNOTATION [305]: It is true that intervals are frequently partially ordered and cannot be compared. Nevertheless, various definitions for ranking intervals have been proposed. In this paper, the authors propose a new definition for order relation between intervals by introducing a parameter called 'a degree between partial and total order', and apply it to the shortest path problem with arcs represented as intervals. In order to solve this problem, they modify Dijkstra's algorithm and propose a new algorithm obtaining some incomparable intervals solutions. Finally, a numerical example is shown.

⇒ANNOTATION [306]: Deals with generalized Petri nets as a flexible formal means for analysis of discrete systems. On the basis of generalized Petri nets, the authors are able to define any deterministic and non-deterministic (discrete stochastic and fuzzy) Petri nets.

⇒ANNOTATION [336]: The Japanese iron and steel industry has rationalized its production operations through the introduction of computer systems. To derive better performance from the computer systems, the industry is developing applications of artificial intelligence (AI), especially expert system and fuzzy logic. In the field of process control, AI has come to be used in various ways to supplement conventional control systems, with tangible results. This paper presents an overview of the background of development of AI and its applications in the Japanese iron and steel industry.

⇒ANNOTATION [347]: In this paper, decision making problems arising from optimal operation planning for hot parts scheduling of gas turbines of thermal power plants are formulated as multiobjective 0-1 integer programming problems. By considering the imprecise nature of human judgments, the fuzzy goals of the decision maker (DM) for each of the objective functions are introduced. The approximate solutions for the formulated problems are derived through genetic algorithms for solving general combinatorial optimization problems. In order to decrease the difficulties for the determination of not only appropriate parameter values in the genetic algorithms but also membership functions representing the fuzzy goals of the DM, simple genetic algorithms are revised and auto-tuning method of the membership functions are proposed. On the basis of the proposed methods, an interactive decision support system is developed, and the feasibility and efficiency of both the proposed methods and the corresponding decision support system are demonstrated via numerical examples.

⇒ANNOTATION [348]: In this paper, Sakawa et al. focus on large-scale multiobjective linear programming problems with block angular structure. By considering the imprecise nature of human judgements, they assume that the decision maker may have fuzzy goals for each of the objective functions. Having elicited the corresponding linear membership functions through the interaction with the decision maker, they adopt the fuzzy aggregated decision. It is shown that the formulated problem can be reduced to one master problem and a number of linear subproblems, and the satisfying solution of the decision maker can be obtained by applying the Dantzig-Wolfe decomposition method.

⇒ANNOTATION [351]: Hard constraints are represented as first-order formulas. An interpretation which satisfies all those first order formulas can be regarded as a solution. Soft constraints can be regarded as providing an order over those interpretations because soft constraints represent criteria to choose the most preferable solution. The most preferred solutions are the most preferred interpretations.

⇒ANNOTATION [359]: This paper proposes a new strategy for motion planning in robotics. When robots performs some tasks, they work according to motion plans. The plans should be effective for the robots. The proposed strategy applies a genetic algorithm (GA) to optimize the motion planning. To evaluate the planned motion, the strategy also applies fuzzy logic to a fitness function. The fitness function is referred to as Fuzzy Critic. The Fuzzy Critic evaluates populations in the GA with respect to multiple factors, while traditional fitness functions evaluate with respect to only one factor. Depending on the goals of the tasks, human operators can easily determine inference rules in the Fuzzy Critic. In this paper, the strategy determines a path for a mobile robot which moves from a starting point to a goal point while avoiding obstacles in a work space and picking up loads on the way. Simulations illustrate the effectiveness of the proposed strategy.

⇒ANNOTATION [362]: The publication contains the technical papers selected for presentation at the Winter 91/92 Information Science Proseminary hold at the Department for Information Systems, Technical University of Vienna. The goal was to give an overview about knowledge based scheduling systems and techniques. The Department for Information Systems hosts since 1989 the Christian Doppler Laboratory for Expert Systems funded by the Austrian Industries Corp. The aim of this laboratory is to do basic research in the field of expert systems while keeping close contacts to various affiliates of the Austrian Industries Corp. A strong team from this laboratory is actively doing research in the field of knowledge based scheduling. Knowledge based scheduling is considered a key to an efficient realization of Computer Integrated Manufacturing. One purpose of the proseminary was therefore allowing students to get acquainted with this subject of high industrial and economic impact. Titles of papers presented follow: Knowledge Based Scheduling - A Tutorial, PEPS: The Prototype Expert Priority Scheduler, ISIS: A Knowledge-Based System for Factory Scheduling, OPIS: An Opportunistic Factory Scheduling System, TABU Search: A Tutorial, A Rule-Based System to Schedule Production, NP-Completeness: Why Scheduling is Difficult, A Reactive Scheduling Agent, Knowledge Based Scheduling: A Survey, Fuzzy-Logic: A Tutorial (several papers), Fuzzy Scheduling: A Case Study.

⇒ANNOTATION [364]: Mathematical-analytical methods are often insufficient for planning problems. This is due to three reasons: The imprecise informations in the production process, combinatorial complexity of the search space, and conflicting objectives for production optimizing. The combination of several knowledge-based techniques, especially approximate reasoning and constraint relaxation, is a promising way to handle these problems. The authors use a case study to demonstrate how knowledge-based scheduling works with the desired capabilities. The applied knowledge representation technique covers the vagueness inherent in the problem domain by using fuzzy set theory. Based on this knowledge representation, the importance of jobs is defined. This classification of jobs is used for the straightforward generation of a schedule. The authors introduce a control strategy which comprises several types of constraints, namely organizational, spatial, and chemical ones. This strategy supports the dynamic relaxation of conflicting constraints in order to improve the schedule. To show the benefits of this strategy, the generation of a schedule for one day is explained in detail.

⇒ANNOTATION [365]: During short-term scheduling in a steel plant, one problem is to know beforehand how long equipment will be usable and when it will have to be maintained. The present system answers this question for a part in a continuous caster in order to inform another system that schedules short-term production. It uses fuzzy inference rules to process input data and to compute the life-expectancy of the part. The system performs better than humans in predicting the life-expectancy since it considers more types of influence and reasons with up-to-date informations. For these reasons, the final schedule matches reality closer, avoiding vasted raw materials and production delays as well as increasing qualities. Both systems are developed in a joint project between the Austrian Industries Holding and the Christian Doppler Laboratory for Expert Systems.

⇒ANNOTATION [366]: This paper was presented in the session on fuzzy constraint propagation chaired by Henri Prade. Mathematical-analytical methods are often insufficient for real-world applications. This is due to three reasons: The imprecise informations in the real-world applications, combinatorial complexity of the search space, and conflicting objectives for optimizing. The combination of several knowledge-based techniques, especially approximate reasoning and constraint relaxation, is a promising way to handle these problems. The paper gives an overview on existing fuzzy constraint relaxation techniques, focusing on the type of problems handled, the techniques used, with examples, advantages, and then proceed to compare these techniques with other constraint relaxation techniques. Special emphasis will be given to the industrial scheduling domain, as this is a very prominent real-world application area for constraint satisfaction methods.

⇒ANNOTATION [367]: During short-term scheduling in a steel plant, one problem is to know beforehand how long equipment will be usable and when it will have to be maintained. The proposed system answers this question for a part in a continuous caster in order to inform another system that schedules short-term production. It uses fuzzy inference rules to process input data and to compute the life-expectancy of the part. Wheras human operators tend to use pessimistic values in order to be on the safe side, the system performs better in predicting the life-expectancy since it considers more types of influence and reasons with up-to-date information. For these reasons, the final schedule matches reality closer, avoiding vasted raw materials and production delays, while at the same time increasing product quality. Both systems are currently developed in a joint project between the Austrian Industries Holding and the Christian Doppler Laboratory for Expert Systems.

⇒ANNOTATION [368]: Mathematical-analytical methods are often insufficient for real-world applications. This is due to three reasons: The imprecise informations in the real-world applications, combinatorial complexity of the search space, and conflicting objectives for optimizing. The combination of several knowledge-based techniques, especially approximate reasoning and constraint relaxation, is a promising way to handle these problems. The paper gives an overview on existing fuzzy constraint relaxation techniques, focusing on the type of problems handled, the techniques used, with examples, advantages, and then proceed to compare these techniques with other constraint relaxation techniques. Special emphasis will be given to the industrial scheduling domain, as this is a very prominent real-world application area for constraint satisfaction methods.

⇒ANNOTATION [369]: Real-world scheduling is decision making under vague constraints of different importance, often using uncertain data, where compromises between antagonistic criteria are allowed. The author explains in theory and by detailed examples a new combination of fuzzy set based constraints and repair based heuristics that help to model these scheduling problems. The authors simplifies the mathematics needed for a method of eliciting the criteria's importances from human experts. He introduces a new consistency test for configuration changes. This test also helps to evaluate the sensitivity to configuration changes. He describes the implementation of these concepts in his fuzzy constraint library ConFLIP++ and in his heuristic repair library DéjàVu. Finally, the author presents results from scheduling a continuous caster unit in a steel plant.

⇒ANNOTATION [377]: One section reviews fuzzy scheduling.

⇒ANNOTATION [385]: Automatic control strategy has been applied effectively to the steel rolling process since the 1960s to improve productivity, yield, and product quality. The hot strip mill is a typical example of the steel rolling process. In the paper, the author gives examples of automatic control systems in the steel rolling process, including functions and methods applied. The following examples are given: combustion control for the reheating furnace; thickness control of steel plate including gain scheduling and feedforward control; and tension control between stands.

⇒ANNOTATION [386]: A scheduling system for steel making process has been developed, which controls the material flow from blast furnace to continuous casters. The scheduling method consists of heuristic logic and mathematical programming algorithm. In the first part of the system, the order of the jobs in each process is decided using a heuristic procedure, and the time schedule of the refining and casting process is calculated using a linear programming method. In the second part, all the restrictions and the requirements of search for the optimum flow with minimum cost are evaluated.

⇒ANNOTATION [387]: A scheduling system which controls the scheduling flow from a blast furnace to a continuous caster in the steel making process has been developed. The scheduling procedure consists of heuristic logic and mathematical programming algorithms, and the minimum cost schedule is determined by solving linear programming and network programming problems. This scheduling program is being applied at Kobe works for daily production, and has proved useful for reducing production costs and operator labor.

⇒ANNOTATION [389]: A production scheduling expert system for the entire steel-making process is described. In order to prevent the knowledge processing time from increasing excessively due to a wide area to be covered, this system does not automatically present a single final schedule, or optimum schedule. Instead, it offers several possible schedules which meet all given conditions, along with their evaluation indexes, using a knowledge base containing accumulated know-how of skilled operators, so that the operator can select the optimum schedule from among those schedules. In the subpart expert system which controls the operation of a blast furnace, the furnace condition is estimated from the heat level and its change. To model fuzziness, numerical factors are introduced, which express the degree of certainty for the correctness of the heuristic estimation. The heat level used for estimation is obtained with a certainty factor by fuzzy reasoning from hot-metal temperature. The relationship between them is expressed by a three-dimensional membership function to reduce the number of rules. A rule learning capability is provided also.

⇒ANNOTATION [390]: Some studies have been made on expert system applied to knowledge engineering for ill-structured problems. As a result, expert systems to control blast furnace thermal conditions and schedule steel-making processes have been developed for practical use.

⇒ANNOTATION [391]: In the raw material yard at Fukuyama works, AI technology is adopted to a field which formerly depended upon the heuristic knowledge of a highly experienced operator, and systemization has been realized. It has enabled the high level standardization and inheritance of the technology as well as quick reactions to operational changes. In this paper, first the storage yard scheduling expert system, second the cooperative function aimed at automation of the expert system, conveyor, and transfer machine, and at last the cooperative function among the expert systems which is aimed at the total optimization of all those operations, are discussed.

⇒ANNOTATION [392]: Due to the difficulty in direct measurement of the inner condition of a blast furnace, it is difficult to establish a mathematical furnace model and automatic furnace control. To solve this problem, an expert system for blast furnace heat level control has been developed. This system has the following two functions: Heat-level estimation using fuzzy logic inference, and heat level control using the production system. The system is effectively employed at Kakogawa works no. 1 blast furnace and Kobe works no. 3 blast furnace as a guide for furnace operators.

⇒ANNOTATION [396]: At Mizushima Works, ore yard control systems of electric equipment were renewed. This was the second step of the iron making department's information system. New belt conveyor control for energy saving and staking control for quality improvement have been applied. Control technologies such as knowledge engineering, fuzzy control, and self-tuning control have been applied. The systems have been achieving good results that were difficult to obtain by conventional ways, and are working without problem since April 1987.

⇒ANNOTATION [400]: A multi-objective decision making model of fuzzy preference relations is presented. A type of dominance of alternative is presented whose purpose is the choice of an optimal alternative in uncertainty in imprecise environments. This system is based on the expected utility decision making in which the probabilities and the utilities are expressed as membership functions. The expected fuzzy utility is calculated on the basis of these probabilities and utilities in each node. Next, the truth value of the fuzzy preference relation is calculated in the decision node. The relationship among multiple interdependent objectives is determined by introducing a fuzzified Hamming distance calculated through the membership functions of utilities and the ordering of alternatives for each objective. The authors employ dome techniques for knowledge representation based on fuzzy production rules and basic concepts from the theory of approximate reasoning.

⇒ANNOTATION [401]: Ambiguity of both linguistic rules and input data affects the reasoning results of expert systems. Fuzzy reasoning and Delphi's method are used in a system for rock identification to reduce this ambiguity. It is suggested that reasoning directivity and rigor factors affect not only searching time to find the routes to a goal, but also help to find the optimal search route itself.

⇒ANNOTATION [410]: Constructive fuzzy constraint satisfaction system based on KEE (frames, lisp-style) using α-cuts, t-norms, and backtracking, with constraint hierarchies (no other way to represent importance and no way to compensate are used). Interesting ideas: machine learning used to reduce number of fuzzy rules; a consistency check (cf Slany [369]) should be made before proceeding with the addition of any new rule base; in other papers application to planning and design problems is discussed; use not point valued membership functions but type 2 fuzziness, i.e. interval valued representation of membership functions; a method similar to the one in Slany [364] is used to describe current system states by linguistic terms; exhaustive search versus tree search method are compared based on dimensionality of the memory space required; knowledge acquisition is main concern; FLAR: fuzzy logic based approximate reasoning toolbox; 120 real life jobs are evaluated and compared with results taken from the ISIS implementation: FLAR is said to produce slightly better schedules, and is termed user-friendly because of the use of linguistic terms.

⇒ANNOTATION [412]: Thickness control in a cold rolling mill is performed by adjusting the reduction force (rolling load) applied by the rolls, and strip tension. This thickness control consists of 'setup control' which determines initial values and 'ACG' which dynamically controls the strip thickness during rolling. For the setup control a mathematical model derived from rolling theory has been used. As the model has some unmeasurable parameters, the values of those parameters have to be estimated. Since inaccurate estimates can cause significant errors, in actual control operation a skilled operator corrects the calculations on the basis of judgement of the rolling condition. This know-how of skilled operators has been quantitatively expressed by a fuzzy model to permit rolling load to be estimated with a fair degree of accuracy. The coefficients of the fuzzy model are obtained by the least square method whereby the actual operating data are weighted by a truth value in the premise, and also modified by sequential learning for each roll so that the model can respond to operational changes. The model is found to have higher estimation accuracy than the preset model of the mill.

⇒ANNOTATION [413]: Urquhart gives a historic overview about Łukasiewicz's logics, Post's many-valued systems, Bochvar's work on paradoxes, Kleene's system and explains them. He then goes on to link them to current activities. Urquhart is however quite pessimistic as he asserts that "the logic of uncertainty, the logic of probability and the logic of error are all non-truth-functional." For instance, probability calculus (including subjective probabilities) is non-truth-functional, "because the probability of a conjunction is not a function of the value of the conjuncts, because the conjuncts may or may not be stochastically independent." Concerning fuzzy logic, in addition to the artificial nature of the precise numerical values assigned to sentences like '73 is a large number' or 'Picasso's *Guernica* is beautiful' he points out that similar problems as involved with interpreting Łukasiewicz's logics arise concerning the correct interpretation since the operations do not correspond to the usual connectives.

⇒ANNOTATION [426]: Analyses inter-human relations and problems related to the task of scheduling rooms and persons having to do with surgery in a large hospital.

⇒ANNOTATION [435]: Well-written but dense article. To optimize the productivity and yield of a modern high-speed continuous casting operation, it is desirable to minimize the number of metallurgical grades that have to be melted to satisfy a collection of customer orders. This problem has been addressed through the development of an expert system for selecting the set of all potential grades for each order and an optimal selection algorithm for determining the actual grades that would be required to produce all orders. As a further refinement, a fuzzy formulation with a membership function based on the likelihood of a grade meeting a customer's specifications without difficulty has been added. The membership functions are weighted because, although mechanical, chemistry and desirability components are considered in the decision-making process, they are clearly not of equal importance. This method enables the plate mill to trade off minimizing the number of grades used against maximizing the likelihood that the customer specifications will be met without difficulty.

⇒ANNOTATION [437]: Refers to a lot of interesting literature concerned with multiobjective decision making under uncertainty (initial contributions by Bellmann and Zadeh [16], Zimmermann extended it to continuous variables, Yager to model objectives of different importance). The author considers only problems where all possible decisions can be compared, i.e. their number is small enough to be manageable and thus the problem complexity is low. Yager argues that establishing assessments of objectives can be achieved by: linear orderings of objectives; intervals; relative ratio; absolute ratings. These are progressively more difficult to obtain from human experts, and forcing the latter to provide such information may yield incorrect answers if the human cannot give the information. Furthermore, as the assessment scale becomes more refined it becomes more sensitive to 'noise' and, consequently, more error prone. An example is given (selecting a car, based on cost, gas mileage, comfort, repair frequency), and trade-offs between the various objectives are acceptable. He describes the approach of Bellmann and Zadeh, which combine all objective evaluations via the minimum-operator, thus always reaching a so-called Pareto optimal solution (a proof is given at the end of the paper). It requires only linear ordering of objectives, but does not permit to distinguish between the importance of objectives. Yager then goes on to explain his extension which allows the assignment of importance to the objectives by performing the exponent of the normal evaluation additionally to the minimum-operator. The importance measures can be expressed on the unit interval. Yager quite correctly points out that many different sets of and-or-implication operations have been proposed so far. In the present paper, Yager proposes a new methodology allowing him to include the differing importance factors while still only requiring an ordinal scale for preference information. It works as follows: For a particular objective the negation of its importance acts as a barrier such that all ratings of alternatives that are below that barrier become equal to the value of the barrier. The motivation is that implication $a \to b$ can be seen as $\neg a \vee b$. That is, he disregards all distinctions below the barrier while keeping distinctions above the barrier. A detailed example about selecting a car is given. Methods for adjudicating ties are given, disregarding successively identical scores. For the case of continuous importance scales, the implication can similarly be replaced by b^a.

⇒ANNOTATION [439]: Coefficients are not associated directly with a particular attribute, but rather with an ordered position. This gives the operators lying between 'and' and 'or' nice features such as symmetry. Yager also gives a good introduction to triangular-, so called t-norms (a class of 'and' operators, since in multicriteria decision making, no compensation for one bad satisfaction is possible with them) and the associated t-conorms (a class of 'or' operators, since they allow no distraction for one good satisfaction). Yager then goes on to propose the OWA operators lying between those two extremes while being indifferent to the individual criteria. He also links them to the notion of quantifiers and shows their usefulness. Yager points to the literature for methods related to the combination of the OWA operators with unequal importances of the different criteria.

⇒ANNOTATION [442]: The basic oxygen furnace (BOF) is a refining process in which oxygen is blown into hot metal produced by the blast furnace. The purpose of blowing control is to control molten steel composition (carbon, phosphorus, etc.) and temperature endpoint. The authors present a system in which existing mathematical models are complemented by expert systems and fuzzy control incorporating empirical rules of skilled operators to improve the accuracy of control. The system consists of three reasoning functions, which operate in the following phases:

1. Reasoning before blowing. A blowing scheme is determined by calculation.
2. Reasoning during blowing. The temperature of the molten steel is measured, and necessary adaptations are made according to the knowledge of experienced melters, which has been captured in the system.
3. Reasoning after blowing. For estimation of end-point composition, a fuzzy control similar to the one in (2.) is used.

After three months of operation, the system performed better than a skilled melter with five years of experience.

⇒ANNOTATION [451]: Short introduction to the articles [40, 297, 462, 9, 211]. Zadeh points out that the term fuzzy logic is used in a narrower and in a broader sense. He lists several successful applications to the conception and the design of intelligent systems of fuzzy logic in the broader sense.

⇒ANNOTATION [461]: Section 13.3 is dedicated to fuzzy set models in production control and scheduling. Section 13.3.1 describes the work of von Altrock [3] who uses fuzzy evaluations for due date satisfaction using the γ-operator. Section 13.3.2 describes the work of Bensana [18] on OPAL. Section 13.3.3 describes a method by Hintz and Zimmermann [190], enriched with examples. Section 13.3.4 presents results by Rinks [323] on production scheduling using linguistic variables. Section 13.3.5 describes fuzzy mathematical programming by Holtz and Desonki [191] for maintenance scheduling, where several aggregation operators were investigated and parametrized membership function were used. Finally, Section 13.3.6 describes the classroom scheduling application of Prade [315].

⇒ANNOTATION [462]: Zimmermann describes how to handle vague linear constraints by allowing smaller violations in inequalities. He deals with over-constrained problems where not all constraints must be absolutely satisfied. Additionally, the decision maker might not want to maximize the objective function but instead try to arrive at an acceptable aspiration level. The author does not distinguish between constraints and objectives, arguing that it models the behavior of decision makers quite well. The solutions can be found by using standard (crisp) linear programming method with only one more variable and one more constraint, which makes this approach computationally very efficient. Extensions for nonlinear membership functions and operators other than minimum (gamma-operator) are given.

Bibliography

NOTE: Order is by name of first author and year. References having annotations are marked with ⇒ANNOTATION. The annotations are listed in Section "Annotations to the Bibliography" starting from page 105. Order of the annotations is according to the reference number in the Bibliography.

An ASCII-version of fuzzy-scheduling related entries in BibTeX-format is located online at
URL: "ftp://mira.dbai.tuwien.ac.at/pub/slany/fuzzy-scheduling.bib.Z".
I will be happy to insert any updates sent to me by electronic mail to wsi@vexpert.dbai.tuwien.ac.at .

[1] J.M. Adamo. Fuzzy decision trees. *Fuzzy Sets and Systems*, 4:207–219, 1980.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):823–843, 1983.

[3] Constantin von Altrock. Konzipierung eines Lösungsverfahrens zur Produktionsplanung und -steuerung in der chemischen Industrie. Master's thesis, Institute for OR, RWTH Aachen, Germany, 1990. In German.

[4] Stéphane Amarger, Didier Dubois, and Henri Prade. Constraint propagation with imprecise conditional probabilities. In *Uncertainty in Artificial Intelligence*, pages 26–34, 1991.

[5] Plamen P. Angelov. A parameterized generalization of fuzzy mathematical programming problems. In *5th IFSA*, pages 612–615, 1993.

[6] Plamen Angelov. A generalized approach to fuzzy optimization. *Int. Journal of Intelligent Systems*, To appear, 1994.

[7] Z.A. Azmi. New fuzzy approaches by using statistical and mathematical methodologies in operations research. *The Journal of Fuzzy Mathematics*, 1(1):69–87, 1993.

[8] K.R. Baker. *Introduction to Sequencing and Scheduling.* John Wiley & Sons, Inc., 1974.

[9] J. F. Baldwin. Fuzzy and probabilistic uncertainties. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 528–537. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[10] F. Barachini and N. Theuretzbacher. The challenge of real-time process control for production systems. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 705–709, 1988.

[11] Gérard Bel, Didier Dubois, Henri Farreny, and Henri Prade. Towards the use of fuzzy rule-based systems in the monitoring of manufacturing systems. In J.P. Crestin and J.F. McWaters, editors, *Software for Discrete Manufacturing (IFIP)*, pages 525–535. Elsevier Science Publishers, 1986.

[12] G. Bel, E. Bensana, D. Dubois, J. Erschler, and P. Esquirol. A knowledge-based approach to industrial job-shop scheduling. In Andrew Kusiak, editor, *Knowledge-Based Systems in Manufacturing*, chapter 10, pages 207–246. Taylor & Francis, 1989. ⇒ANNOTATION.

[13] G. Bel, E. Bensana, D. Dubois, and J.L. Koning. Handling fuzzy priority rules in a job-shop scheduling system. In *3rd IFSA*, Seattle, Washington, USA. August 1989.

[14] Gérard Bel, Eric Bensana, Pierre Berlandier, Philippe David, Hélène Fargier, Christine Gaspin, Khaled Ghedira, Philippe Janssen, Philippe Jégou, Tibor Kökeny, Jérôme Lang, David Lesaint, Roger Martin-Clouaire, Bertrand Neveu, Jean-Pierre Rellier, Thomas Schiex, Brigitte Trousse, Gérard Verfaillie, and Marie-Catherine Vilarem (Projet CSPFlex.). Représentation et traitement pratique de la flexibilité dans les problèmes sous contraintes. In *Actes des journées nationales du PRC GDR Intelligence Artificielle*, Marseille, France. October 1992. In French.

[15] G. Bel, E. Bensana, and J.L. Koning. Gestion de points de vue antagonistes à l'aide de règles de décisions floues: application à l'ordonnancement d'atelier. In *Les applications des ensembles flous*, Nîmes, France. November 1992. In French.

[16] R. E. Bellman and L. A. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 17-B(4):141–164, December 1970. Republished in [438].

[17] E. Bensana, M. Carrege, G. Bel, and D. Dubois. An expert system approach to industrial job shop scheduling. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1645–1650, San Francisco, CA. April 1986.

[18] E. Bensana, G. Bel, and D. Dubois. OPAL, a multi knowledge-based system for industrial job-shop scheduling. *International Journal of Production Research*, 26(5):795–815, 1988.

[19] E. Bensana, G. Bel, and D. Dubois. A knowledge based approach to industrial job shop scheduling. In I. B. Türksen, editor, *Computer Integrated Manufacturing, Current Status and Changes*, pages 295–330. Springer Verlag, 1988.

[20] Eric Bensana. Advanced job shop scheduling in aeronautical manufacturing. In *Scheduling of Production Processes (ECAI'92 Workshop)*, pages 61–70. ECCAI, August 1992.

[21] Pauline M. Berry. Scheduling and uncertainty reasoning. In *Scheduling of Production Processes (ECAI'92 Workshop)*, pages 16–25. ECCAI, August 1992.

[22] Pauline M. Berry. Scheduling: A problem of decision-making under uncertainty. In Bernd Neumann, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI-92)*, pages 638–642, August 1992. ⇒ANNOTATION.

[23] Pauline M. Berry. The PCP: A predictive model for satisfying conflicting objectives in scheduling problems. *AI in Engineering*, 7:227–242, 1992. ⇒ANNOTATION.

[24] L. von Bertalanffy. The organism considered as a physical system. In L. von Bertalanffy, editor, *General system theory.* Braziller, New York, 1968. Republished from 1940. ⇒ANNOTATION.

[25] James C. Bezdek. On the relationship between neural networks, pattern recognition and intelligence. *The International Journal of Approximate Reasoning*, 6:85–107, 1992. ⇒ANNOTATION.

[26] V. Bhattacharya, J.R. Rao, and R.N. Tiwari. Fuzzy multi-criteria facility location problem. *Fuzzy Sets and Systems*, 51:277–287, 1992.

[27] V. Bhattacharya, J.R. Rao, and R.N. Tiwari. Bi-criteria multi facility location problem in fuzzy environment. *Fuzzy Sets and Systems*, 56:145–153, 1993.

[28] E. Biefeld and L. Cooper. Bottleneck identification using process chronologies. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'91)*, Sidney, Australia. 1991.

[29] C. Bing-Yuan. Fuzzy geometric programming (i). *Fuzzy Sets and Systems*, 53:135–153, 1993.

[30] C. Bing-Yuan. Extended fuzzy geometric programming. *The Journal of Fuzzy Mathematics*, 1(2):285–293, 1993.

[31] M.P. Biswal. Fuzzy programming technique to solve multi-objective geometric programming problems. *Fuzzy Sets and Systems*, 51:67–71, 1992.

[32] A.K. Bit, M.P. Biswal, and S.S. Alam. Fuzzy programming approach to multicriteria decision making transportation problem. *Fuzzy Sets and Systems*, 50:135–141, 1992.

[33] A.K. Bit, M.P. Biswal, and S.S. Alam. Fuzzy programming approach to multiobjective solid transportation problem. *Fuzzy Sets and Systems*, 57:183–194, 1993.

[34] A.K. Bit, M.P. Biswal, and S.S. Alam. Fuzzy programming technique for multiobjective capacitated transportation problem. *The Journal of Fuzzy Mathematics*, 1(2):367–376, 1993.

[35] G.R. Bitran. Linear multiple objective problems with interval coefficients. *Management Science*, 26(7):694–706, 1980.

[36] Arthur Bloch, editor. *Murphy's Law Complete: All the reasons why everything goes wrong!* Methuen London Ltd, 1986.

[37] S.F. Bocklisch, editor. *Prozeßanalyse mit unscharfen Verfahren.* Verlag Technik, Berlin, 1987. In German. ⇒ANNOTATION.

[38] Piero P. Bonissone. Reasoning, plausible. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 2, pages 1307–1322. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[39] Serge Boverie, Bernard Demaya, and André Titli. Fuzzy logic controller compared with other automatic control approaches. In *Proceedings of the 30th IEEE-CDC Conference on Decision and Control*, Brighton (U.K.). December 1991. ⇒ANNOTATION.

[40] Bill P. Buckles and Frederick E. Petry. Fuzzy databases. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 508–515. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[41] J.J. Buckley. Fuzzy programming and the pareto optimal set. *Fuzzy Sets and Systems*, 10:57–63, 1983.

[42] J.J. Buckley. Possibility and necessity in optimization. *Fuzzy Sets and Systems*, 25:1–13, 1988.

[43] J.J. Buckley. Possibilistic linear programming with triangular fuzzy numbers. *Fuzzy Sets and Systems*, 26:135–138, 1988.

[44] J.J. Buckley. Solving possibilistic linear programming problems. *Fuzzy Sets and Systems*, 31:329–341, 1989.

[45] J.J. Buckley. Stochastic versus possibilistic programming. *Fuzzy Sets and Systems*, 34:173–177, 1990.

[46] L. Campos. Fuzzy linear programming models to solve fuzzy matrix games. *Fuzzy Sets and Systems*, 32:271–289, 1989.

[47] L. Campos and J.L. Verdegay. Linear programming problems and ranking of fuzzy numbers. *Fuzzy Sets and Systems*, 32:1–11, 1989.

[48] L. Campos, A. Gonzalez, and M.A. Vila. On the use of the ranking function approach to solve fuzzy matrix games in a direct way. *Fuzzy Sets and Systems*, 49:1–11, 1992.

[49] Ulrico Canzi and Giovanni Guida. The temporal model of CRONOS-III: A knowledge-based system for production scheduling. In Jürgen Dorn and Karl A. Froeschl, editors, *Scheduling of Production Processes*, Ellis Horwood Series in Artificial Intelligence, chapter 11, pages 113–129. Ellis Horwood Limited, 1993. ⇒ANNOTATION.

[50] T. Cao and A.C. Sanderson. Task sequence planning using fuzzy Petri nets. In *Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics. Decision Aiding for Complex Systems*, volume 1, pages 349–354. IEEE, 1991.

[51] T. Cao and A.C. Sanderson. Sensor-based error recovery for robotic task sequences using fuzzy Petri nets. In *Proceedings. 1992 IEEE International Conference on Robotics And Automation*, volume 2, pages 1063–1069, Los Alamitos, CA, USA. IEEE Computer Society Press, 1992. ⇒ANNOTATION.

[52] J. Cardoso, Robert Valette, and Didier Dubois. Petri nets with uncertain markings. In G. Rozenberg, editor, *Advances in Petri Nets (10th Int. Conf. on Applications and Theory of Petri Nets, Bonn, West Germany, 28-30 June 1989)*, volume 483 of *Lecture Notes in Computer Science*, pages 64–78, Berlin, Heidelberg, New York. Springer-Verlag, 1990. URL: ftp://laas.laas.fr/pub/robert/petri89.tex.

[53] C. Carlsson. Fuzzy multiobjective programming with composite compromises. In M. Grauer, A. Lewandowski, and A.P. Wierzbicki, editors, *Multiobjective and Stochastic Optimization, CP-82-S12*. IIASA, Laxemburg, Austria, 1982.

[54] Christer Carlsson and Pekka Korhonen. A parametric approach to fuzzy linear programming. *Fuzzy Sets and Systems*, 20:17–30, 1986.

[55] D. Chakraborty, J.R. Rao, and R.N. Tiwari. Multiobjective imprecise-chance constrained programming problem. *The Journal of Fuzzy Mathematics*, 1(2):377–387, 1993.

[56] S. Chanas and J. Kamburowski. The use of fuzzy variables in PERT. *Fuzzy Sets and Systems*, 5(11):11–19, 1981.

[57] S. Chanas. Parametric programming in fuzzy linear programming. *Fuzzy Sets and Systems*, 11:243–251, 1983.

[58] S. Chanas, W. Kolodziejczyk, and A. Machaj. A fuzzy approach to the transportation problem. *Fuzzy Sets and Systems*, 13:211–222, 1984.

[59] S. Chanas. Fuzzy programming in multiobjective linear programming – a parametric approach. *Fuzzy Sets and Systems*, 29:303–313, 1989.

[60] S. Chanas. Parametric techniques in fuzzy linear programming problems. In J.L. Verdegay and M. Delgado, editors, *The Interface between Artificial Intelligence and Operations Research in Fuzzy Environment*, ISR Series no. 95, pages 105–116. TÜV Rheinland, 1989.

[61] S. Chanas, M. Delgado, J.L. Verdegay, and M.A. Vila. Interval and fuzzy extensions of classical transportation problems. *Transportation Planning and Technology*, 17(2):203–218, 1993. Special Issue: Application of Fuzzy Set Theory to Transportation.

[62] Peter Cheeseman. In defense of probability. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 1002–1009, Los Angeles, Ca. Morgan Kaufmann Publishers, Inc., August 1985. ⇒ANNOTATION.

[63] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the *really* hard problems are. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 331–337, 1991.

[64] Dahai Chen, Yuhuan Pan, and Jinsong Xue. A fuzzy production system with backtracking control strategy for multiobjective scheduling to a one-machine-n-parts problem. In Andrew Kusiak, editor, *Modern Production Management Systems*, pages 135–145. Elsevier, 1987. ⇒ANNOTATION.

[65] Brigitte Pradin-Chezalviel and Robert Valette. Petri nets as a candidate for a semantical interpretation of linear logic. In *GWAI'92 (16. Fachtagung für Künstliche Intelligenz), Workshop Logic and Change, Bonn, Germany*, September 1992. URL: ftp://laas.laas.fr/pub/robert/logichange.ps.

[66] Brigitte Pradin-Chezalviel and Robert Valette. Reasoning about resources by combining Petri nets and linear logic. Technical Note LAAS 92444, LAAS, December 1992. URL: ftp://laas.laas.fr/pub/robert/chicago.ps.

[67] W. Y. Chiang and M. S. Fox. Protection against uncertainty in a deterministic schedule. In *Proceedings 4th International Conference on Expert Systems and the Leading Edge in Production and Operations Management*, pages 184–197, Hilton Head Island. 1990.

[68] F. Choobineh and M. Shivani. A dynamic process planning and scheduling algorithm. In Ming Leu, editor, *Proceedings of the JAPAN/USA Symposium on Flexible Automation (A Pacific Rim Conference)*, volume 1, pages 429–432, San Francisco, California. July 1992. ⇒ANNOTATION.

[69] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling.* Addison Wesley, 1967.

[70] Earl Cox. Integrating fuzzy logic into neural nets. *AI Expert*, pages 43–47, June 1992. ⇒ANNOTATION.

[71] L. Custòdio, C. Bispo, and J. Sentieiro. Fuzzy logic applied to production scheduling. In *Proceedings of the CIM-Europe Workshop* CIM in the Process Industry, Athens, Greece. June 1991.

[72] Luis Marques Custòdio and Carlos Pinto-Ferreira. Production planning and scheduling using fuzzy logic. In Roger Kerr, editor, *Workshop Notes of the Workshop on Fuzzy Scheduling Systems*, Linz, Austria. University of Linz, Department of Mathematics, June 1993.

[73] Piotr Czyżak and Roman Słowiński. Multiobjective diet optimization problem under fuzziness. In J.L. Verdegay and M. Delgado, editors, *The Interface between Artificial Intelligence and Operations Research in Fuzzy Environment*, ISR Series no. 95, pages 85–103. TÜV Rheinland, 1989.

[74] P. Czyżak and R. Słowiński. FLIP – multiobjective linear programming package. Collaborative Paper CP-91-008, IIASA, Laxenburg, 1991.

[75] Piotr Czyżak and Roman Słowiński. A fuzzy MOLP method with graphical display of fuzziness. *Annales Univ. Sci. Budapest, Sect. Comp.*, 12:59–67, 1991.

[76] Piotr Czyżak and Roman Słowiński. A visual interactive method for MOLP problems with fuzzy coefficients. In R. Lowen and M. Roubens, editors, *Fuzzy Logic*, pages 321–332. Kluwer Academic Publishers, 1993.

[77] E.L. De Porter and K.P. Ellis. Optimization of project networks with goal programming and fuzzy linear programming. *Computers and Industrial Engineering*, 19:500–504, 1990.

[78] R. Dechter and I. Meiri. Experimental evaluation of preprocessing techniques in constraint satisfaction problems. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 271–277, 1989.

[79] M. Delgado. A resolution method for multiobjective problems. *European Journal of Operational Research*, 13:165–172, 1983.

[80] M. Delgado, J.L. Verdegay, and M.A. Vila. Solving the biobjective linear programming problem: A fuzzy approach. In M.M. Gupta et al., editor, *Approximate Reasoning in Expert Systems*, pages 317–322. North-Holland, 1985.

[81] M. Delgado, J.L. Verdegay, and M.A. Vila. Fuzzy transportation problems: A general analysis. In J. Kacprzyk and S.A. Orlovski, editors, *Optimization Models using Fuzzy Sets and Possibility Theory*, pages 342–258. D. Reidel Publishing Co., 1987.

[82] M. Delgado, J.L. Verdegay, and M.A. Vila. Imprecise costs in mathematical programming problems. *Control and Cybernetics*, 16(2):113–121, 1987.

[83] M. Delgado, J.L. Verdegay, and M.A. Vila. On fuzzy linear programming models. In *Preprints of II IFSA Congress*, pages 715–718, Tokyo. 1987.

[84] M. Delgado, J.L. Verdegay, and M.A. Vila. A general model for fuzzy linear programming. *Fuzzy Sets and Systems*, 29:21–29, 1989.

[85] M. Delgado, J.L. Verdegay, and M.A. Vila. Relating different approaches to solve linear programming problems with imprecise costs. *Fuzzy Sets and Systems*, 37(1):33–42, 1990.

[86] M. Delgado, J.L. Verdegay, and M.A. Vila. A possibilistic approach for multiobjective programming problems efficiency of solutions. In R. Słowiński and J. Teghem Jr., editors, *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, pages 229–248. Kluwer Academic Publishers, 1990.

[87] M. Delgado, F. Herrera, J.L. Verdegay, and M.A. Vila. Post-optimality analysis on the membership functions of a fuzzy linear programming problem. *Fuzzy Sets and Systems*, 53:289–297, 1993.

[88] M. Delgado, J. Kacprzyk, J.L. Verdegay, and M.A. Vila, editors. *Fuzzy Optimization. Recent Advances.* Physica-Verlag, 1994.

[89] Jürgen Dorn and Reza Shams. An expert system for scheduling in a steelmaking plant. In *Proceedings of the 1st World Congress on Expert Systems*, pages 395–404, Orlando Fla. Pergamon Press, December 1991.

[90] Jürgen Dorn, Wolfgang Slany, and Christian Stary. Uncertainty management by relaxation of conflicting constraints in production scheduling. In Marc Drummond, Mark Fox, Austin Tate, and Monte Zweben, editors, *Practical Approaches to Scheduling and Planning*, Working Notes AAAI Spring Symposium Series, pages 62–66, Stanford, CA. Published by the American Association of Artificial Intelligence, March 1992.

[91] Jürgen Dorn, Wolfgang Slany, and Christian Stary. Uncertainty management by relaxation of conflicting constraints in production process scheduling. CD-Technical Report 92/33, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, January 1992. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/cd-tr9233.ps.Z. ⇒ANNOTATION.

[92] Jürgen Dorn, Wolfgang Slany, and Christian Stary. Uncertainty management by relaxation of conflicting constraints in production scheduling. In Marc Drummond, Mark Fox, Austin Tate, and Monte Zweben, editors, *Practical Approaches to Scheduling and Planning*, Working Notes from the 1992 AAAI Spring Symposium Series, republished as NASA Technical Report FIA-92-17, pages 62–66, Stanford, CA. NASA Ames Research Center, Artificial Intelligence Branch, May 1992.

[93] Jürgen Dorn and Wolfgang Slany, editors. Scheduling von Produktionsprozessen - Von linearen Integermodellen zu symbolischen AI-Modellen. CD-Studie 93/9, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, July 1993. In German.

[94] Jürgen Dorn and Wolfgang Slany. A flow shop with compatibility constraints in a steelmaking plant. CD-Technical Report 93/56, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993.

[95] Jürgen Dorn, Mario Girsch, and Wolfgang Slany. Reparatur von Plänen durch fallbasiertes Schließen. In A. Bockmayr and F. J. Radermacher, editors, *Forschungsbericht des Max-Planck-Instituts für Informatik zum Workshop Künstliche Intelligenz und Operations Research*, September 1993. In German.

[96] Jürgen Dorn, Wolfgang Slany, Wolfgang Snopek, Christian Stary, Wolfgang Steindl, and Klaus Stohl. Aufgabenananlyse der Dispatcher im Stahlwerk LD3. CD-Studie 93/11, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993. In German.

[97] Jürgen Dorn, Roger Kerr, and Gabi Thalhammer. Reactive scheduling in a fuzzy-temporal framework. CD-Technical Report 93/55, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993. Appeared also as Preprints of the IFIP TC5/WG5.7 Workshop on Knowledge-based Reactive Scheduling, October 1993, Athens.

[98] Jürgen Dorn. Supporting scheduling with temporal logic. In *IJCAI 93 Workshop on Knowledge-Based Production Planning, Scheduling, and Control*, pages 113–124, Chambery. August 1993.

[99] Jürgen Dorn, Wolfgang Slany, Christian Stary, Wolfgang Steindl, Wolfgang Snopek, and Klaus Stohl. Aufgabenbasierte Spezifikation des Bildschirmarbeitsplatzes für Dispatcher im Stahlwerk LD3. CD-Studie 93/12, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993. In German.

[100] Jürgen Dorn and Karl A. Froeschl, editors. *Scheduling of Production Processes*. Ellis Horwood Series in Artificial Intelligence. Ellis Horwood Limited, 1993.

[101] Jürgen Dorn and Wolfgang Slany. A flow shop with compatibility constraints in a steelmaking plant. In Mark Fox and Monte Zweben, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.

[102] Jürgen Dorn, Mario Girsch, Günther Skele, and Wolfgang Slany. Comparison of iterative improvement techniques for schedule optimization. CD-Technical Report 94/61, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994.

[103] Jürgen Dorn and Roger Kerr. Co-operating scheduling systems communicating through fuzzy sets. To be presented at the IFAC-IMS Workshop in Vienna, Austria., 1994.

[104] Jürgen Dorn. Interaction with a scheduling system by soft constraints. To be presented at the SIGMAN Workshop on Scheduling in New Orleans, U.S.A., 1994.

[105] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer Verlag, Heidelberg, 1993. ⇒ANNOTATION.

[106] Didier Dubois and Henri Prade. Algorithmes de plus court chemins pour traiter des données floues. *R.A.I.R.O. Recherche opérationalle/Opérations Research*, 12(2):213–277, May 1978. In French.

[107] D. Dubois and H. Prade. The advantages of fuzzy approach in OR/MS demonstrated on two examples of resource allocation problems. In *5th Int. Congress on Cybernetics and Systems Research*, Vienna. 1980.

[108] D. Dubois, G. Bel, H. Farreny, and H. Prade. Toward the use of fuzzy rule-based systems in the monitoring of manufacturing systems. In J. P. Crestin, editor, *Software for Discrete Manufacturing*. North-Holland, Amsterdam, 1986.

[109] D. Dubois and H. Prade. Decision evaluation methods under uncertainty and imprecision. In Janusz Kacprzyk and Mario Fedrizzi, editors, *Combining Fuzzy Imprecision with Probabilistic Uncertainty in Decision Making*, volume 310 of *Lecture Notes in Economical and Mathematical Systems*, pages 48–65. Springer Verlag, 1986.

[110] Didier Dubois. Linear programming with fuzzy data. In James C. Bezdek, editor, *Analysis of Fuzzy Information*, volume III of *Applications in Engineering and Science*, pages 241–263. CRC Press, 1987. This paper was also presented at the XXVIth meeting of The Institute of Managment Sciences, Copenhagen, June 1984, as well as the Int. Conf. on Fuzzy Information Processing (FIP-84), Kauai, Hawaiian Islands, July 1984.

[111] Didier Dubois and Henri Prade. Fuzzy numbers: An overview. In James C. Bezdek, editor, *Analysis of Fuzzy Information: Artificial Intelligence and Decision Systems*, volume I, pages 3–39. CRC Press, Inc., Boca Raton, FL (USA), 1987.

[112] Didier Dubois. An application of fuzzy arithmetic to the optimization of industrial machining process. *Mathematic Modelling*, 9(6):461–475, 1987.

[113] Didier Dubois and Henri Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, N.Y., 1988.

[114] Didier Dubois. Fuzzy knowledge in an artificial intelligence system for job-shop scheduling. In Gerald W. Evans, Waldemar Karwowski, and Mickey R. Wilhelm, editors, *Applications of Fuzzy Set Methodologies in Industrial Engineering*, volume 8 of *Advances in Industrial Engineering*, pages 73–89. Elsevier, 1989.

[115] Didier Dubois and Henri Prade. Processing fuzzy temporal knowledge. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4):729–744, August 1989.

[116] Didier Dubois and Jean-Luc Koning. DEBORA: A decision engine based on rational aggregation. In *The 2nd Annual IEEE Conference on AI, Simulation and Planning in High Autonomy Systems*, pages 68–77, Cocoa Beach, Florida. April 1991.

[117] Didier Dubois, Hélène Fargier, and Henri Prade. Propagation and satisfaction of flexible constraints. Rapport IRIT 92-59-R, Institut de Recherche en Informatique de Toulouse (IRIT), November 1992. To appear in "Fuzzy Sets, Neural Networks and Soft Computing", edited by R. R. Yager and L. A. Zadeh, Kluwer Academic Publ., 1993.

[118] Didier Dubois, Hélène Fargier, and Henri Prade. Handling flexibility and uncertainty in job-shop scheduling. In Roger Kerr, editor, *Workshop Notes of the Workshop on Fuzzy Scheduling Systems*, Linz, Austria. University of Linz, Department of Mathematics, June 1993.

[119] Didier Dubois, Henri Prade, and Ronald R. Yager, editors. *Readings in Fuzzy Sets for Intelligent Systems*. Morgan Kaufmann, 1993.

[120] Didier Dubois, Hélène Fargier, and Henri Prade. Handling flexibility and uncertainty in job-shop scheduling. Rapport IRIT 93-30-R, Institut de Recherche en Informatique de Toulouse (IRIT), November 1993. ⇒ANNOTATION.

[121] Didier Dubois, Hélène Fargier, and Henri Prade. Propagation and satisfaction of flexible constraints. In Yager R.R. and Zadeh L.A., editors, *Fuzzy Sets, Neural Networks and Soft Computing*. Kluwer Acad., 1993.

[122] Didier Dubois, Hélène Fargier, and Henri Prade. The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction. In *Proc. 2^{nd} IEEE Conference on Fuzzy Sets*, San Francisco, CA. March 1993.

[123] Didier Dubois, Hélène Fargier, and Henri Prade. The use of fuzzy constraints in job-shop scheduling. In *Proc. of IJCAI-93/SIGMAN Workshop on Knowledge-based Production Planning, Scheduling and Control*, Chambery, France. August 1993.

[124] D. Dutta, J.R. Rao, and R.N. Tiwari. Sensitivity analysis in fuzzy linear fractional programming problem. *Fuzzy Sets and Systems*, 48:211–216, 1992.

[125] D. Dutta, R.N. Tiwari, and J.R. Rao. Multiple objective linear fractional programming – a fuzzy set theoretic approach. *Fuzzy Sets and Systems*, 52:39–45, 1992.

[126] D. Dutta, J.R. Rao, and R.N. Tiwari. Fuzzy approach for multiple criteria linear fractional optimization: A comment. *Fuzzy Sets and Systems*, 54:347–349, 1993.

[127] Ziad A. Eldukair and Bilal M. Ayyub. Multi-attribute fuzzy decisions in construction strategies. *Fuzzy Sets and Systems*, 46:155–165, 1992.

[128] Charles Elkan. The paradoxical success of fuzzy logic. In *Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI'93*, pages 698–703, Washington D.C. July 1993. URL: ftp://cs.ucsd.edu/pub/elkan/paradoxicalsuccess.ps.Z; Responses: URL: ftp://ftp.cs.cmu.edu/user/ai/areas/fuzzy/doc/elkan/response.txt. ⇒ANNOTATION.

[129] W. Eversheim, Hans-Peter Lipp, and F. Lehmann. Getting trouble in assembly under control. fuzzy-Petri-networks for replanning. *VDI-Zeitschrift*, 134(2):93–96, February 1992.

[130] C. Fabian and M. Stoica. Fuzzy integer programming. In H.J. Zimmermann, L.A. Zadeh, and B.R. Gaines, editors, *Fuzzy Sets and Decision Analysis*, pages 123–132. North-Holland, Amsterdam, 1984.

[131] Hélène Fargier. Problèmes de satisfaction de contraintes floues. Technical Report 92-29-R, IRIT-UPS, Toulouse, France, September 1992. In French.

[132] Hélène Fargier, Roger Martin-Clouaire, and Thomas Schiex. Satisfaction de contraintes souples. In *Journées nationales "Les applications des ensembles flous"*, pages 183–191, Nîmes, France. October 1992. In French.

[133] Hélène Fargier, Jérôme Lang, and Thomas Schiex. Selecting preferred solutions in fuzzy constraint satisfaction problems. In Hans-Jürgen Zimmermann, editor, *EUFIT'93, First European Congress on Fuzzy and Intelligent Technologies*, pages 1128–1134, Aachen, Germany. Augustinus Buchhandlung, September 1993. ⇒ANNOTATION.

[134] M. Fedrizzi, J. Kacprzyk, and M. Roubens, editors. *Interactive Fuzzy Optimization.* Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1991.

[135] M. Fedrizzi, J. Kacprzyk, and J.L. Verdegay. A survey of fuzzy optimization and mathematical programming. In M. Fedrizzi, J. Kacprzyk, and M. Roubens, editors, *Interactive Fuzzy Optimization*, Lecture Notes in Economics and Mathematical Systems, pages 15–28. Springer-Verlag, 1991.

[136] M. Fedrizzi and R. Fuller. Stability in possibilistic linear programming with continuous fuzzy number parameters. *Fuzzy Sets and Systems*, 47(2):187–191, 1992.

[137] Y.J. Feng. A method using fuzzy mathematics to solve the vectormaximum problem. *Fuzzy Sets and Systems*, 9:129–136, 1983.

[138] Dimitar Filev. Fuzzy optimal control. *Fuzzy Sets and Systems*, 47:151–156, 1992. ⇒ANNOTATION.

[139] Clive Finkelstein. *An introduction to information engineering : from strategic planning to information systems.* Addison-Wesley Publishing Company, Inc., 1989. ⇒ANNOTATION.

[140] B. Fischer. Ein Werkzeug zur auftragsbezogenen Generierung von Fuzzy Petri–Netz Modellen einer Fertigungssteuerung. *VDI–Berichte*, 1035:91–108, 1993. In German.

[141] Roger Fisher, William Ury, and Bruce Patton. *Getting to Yes: Negotiating an agreement without giving in.* Business Books Limited, 2nd enlarged and revised edition, 1991. ⇒ANNOTATION.

[142] Mark S. Fox. *Constraint-Directed Search: A Case Study of Job-Shop Scheduling.* Pitman, London, 1987.

[143] Mark S. Fox and Norman Sadeh. Why is scheduling difficult? A CSP perspective. In *Proceedings of the European Conference on Artificial Intelligence*, pages 754–767, 1990.

[144] Eugene C. Freuder and Richard J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58:21–70, 1992.

[145] Koji Fukumori. Fundamental scheme for train scheduling. MIT AI Memo 596, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, September 1980. ⇒ANNOTATION.

[146] R. Fuller. On stability in fuzzy linear programming problems. *Fuzzy Sets and Systems*, 30:339–344, 1989.

[147] Robert Fullér. On Hamacher sum of triangular fuzzy numbers. *Fuzzy Sets and Systems*, 42(2):205–212, July 1991. ⇒ANNOTATION.

[148] M. Funabashi and S. Masui. Expert systems in system control. *Transactions of the Information Processing Society of Japan*, 28(2):197–206, 1987. In Japanese. ⇒ANNOTATION.

[149] Takeshi Furuhashi et al. An acquisition of scheduling knowledge using fuzzy classifier system. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[150] Nagata Furukawa et al. Fuzzy linear programming with a partial order relation among fuzzy number. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[151] M.C. Garcia-Aguado and J.L. Verdegay. On the sensitivity of membership functions for fuzzy linear programming problems. *Fuzzy Sets and Systems*, 56(1):47–49, 1993.

[152] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and Co., 1979.

[153] J. Geidel. Zeitplanung von Projekten mit unscharfen Daten. Diplomarbeit, Univ. Karlsruhe, 1989. In German.

[154] J. Geidel. Project scheduling with fuzzy data. *Methods of Operations Research*, 62:339–347, 1990.

[155] Mario Girsch. Optimierung von Schedules mit Genetischen Algorithmen und Iterativer Vertiefung. Diplomarbeit (≃ master's thesis), Technical University of Vienna, 1994. In German.

[156] Fred Glover. Tabu search–part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[157] Fred Glover. Tabu search–part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[158] Bernard Grabot. Fuzzy knowledge modelling in complex process control. In *NAFIPS'90: Quater Century of Fuzziness*, pages 200–203, Toronto, Canada. June 1990.

[159] Bernard Grabot. A decision support system for variable routings management in manufacturing systems. *Fuzzy Sets and Systems*, 58:87–104, 1993.

[160] Bernard Grabot and Laurant Geneste. Dispatching rules in scheduling: a fuzzy approach. *International Journal of Production Research*, to be published, 1994. ⇒ANNOTATION.

[161] Bernard Grabot, Laurant Geneste, and Arnaud Dupeux. Multi-heuristic scheduling: three approaches to tune compromises. *Journal of Intelligent Manufacturing*, to be published, 1994. ⇒ANNOTATION.

[162] Qi Guan and Gerhard Friedrich. Extending constraint satisfaction problem solving in structural design. In *Proceedings of the 5th International Conference IEA/AIE*, Paderborn, Germany. June 1992.

[163] Qi Guan and Gerhard Friedrich. Fuzzy control over constraint satisfaction problem solving in structural design. In *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, San Francisco. March 1993.

[164] Qi Guan. *Extending Constraint Satisfaction Problem Solving with Fuzzy Set Theory.* PhD thesis, Technical University of Vienna, 1994. ⇒ANNOTATION.

[165] Rolf Günther and Hans-Peter Lipp. A fuzzy Petri net concept for complex decision making processes in production control. In Hans-Jürgen Zimmermann, editor, *EUFIT'93, First European Congress on Fuzzy and Intelligent Technologies*, pages 290–295, Aachen, Germany. Augustinus Buchhandlung, September 1993.

[166] R. Günther and J. Toggweiler. Ein Fuzzy Ansatz für die operative Führung eines Montagebandes im Fahrzeugkranbau. *VDI–Berichte*, 1035:73–89, 1993. In German.

[167] H. Hamacher, H. Leberling, and H.J. Zimmermann. Sensitivity analysis in fuzzy linear programming. *Fuzzy Sets and Systems*, 1(4):269–281, 1978.

[168] E.L. Hannan. Linear programming with multiple fuzzy goals. *Fuzzy Sets and Systems*, 6:235–248, 1981.

[169] E.L. Hannan. On fuzzy goal programming. *Decision Science*, 12:522–531, 1981.

[170] E.L. Hannan. Fuzzy programming with multiple fuzzy goals. *Fuzzy Sets and Systems*, 6:235–248, 1981.

[171] Maciej Hapke. System PROFAD for fuzzy project scheduling. Technical Report 11/91, Inst. of Computing Science, Technical Univ. of Poznan, 1991.

[172] Maciej Hapke and Roman Słowiński. A DSS for resource-constrained project scheduling under uncertainty. *Journal of Decision Systems*, 2(2):111–128, 1993. ⇒ANNOTATION.

[173] Maciej Hapke, Andrzej Jaszkiewicz, and Roman Słowiński. Fuzzy project scheduling system for software development. *Fuzzy Sets and Systems*, Accepted for publication, 1994.

[174] A. Hasegawa, F. Taki, and N. Yoshitani. Development of shape control system for cold strip mill. *CAMP-ISIJ*, 2:463ff, 1989. In Japanese. ⇒ANNOTATION.

[175] Akihiko Hasegawa and Fumio Taki. Development of shape fuzzy control system for cold strip mill. *Seitetsu Kenkyu (Nippon Steel Research Journal)*, Special Issue on Instrument and Control Engineering(339):53–56, October 1990. In Japanese (ISSN: 0370-9566). ⇒ANNOTATION.

[176] K. Hashimoto, S. Matsuoka, and M. Toyoshima. Blast furnace artificial intelligence system based on ESHELL for NKK corporation. *Fujitsu (Japan)*, 38(5):422–428, 1987. In Japanese (ISSN: 0016-2515). ⇒ANNOTATION.

[177] K. Hashimoto. Expert system development for blast furnace operation control at Fukuyama works. In *Proceedings of the Australian Computer Conference 1987*, pages 793–810, Melbourne, Vic., Australia. Darlinghurst, NSW, Australia: Australian Computer Society, September 1987. ⇒ANNOTATION.

[178] Tomonori Hashiyama et al. A study on varying degrees of attention in multi-attribute decision making processes. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[179] Itsuo Hatono, Tetsuya Suzuka, Motohide Umano, and Hiroyuki Tamura. Towards intelligent scheduling for flexible manufacturing: Application of fuzzy inference to realize high-variety of objectives. In Ming Leu, editor, *Proceedings of the JAPAN/USA Symposium on Flexible Automation (A Pacific Rim Conference)*, volume 1, pages 433–440, San Francisco, California. July 1992. ⇒ANNOTATION.

[180] D. Heckerman and A. Mamdani, editors. *Uncertainty in Artificial Intelligence (Proceedings of the 9th Conference)*. Morgan Kaufmann Publishers, 1993.

[181] Ch. Helfrich. Die Auswirkung der Fuzzy-Technologie auf die Logistik. In German. Fachhochschule München. Handout from presentation at the Wirtschaftsforum der Führungskräfte, Vienna, Austria, October 1992.

[182] F. Herrera and J.L. Verdegay. Approaching fuzzy integer linear programming problems. In M. Fedrizzi, J. Kacprzyk, and M. Roubens, editors, *Interactive Fuzzy Optimization*, Lecture Notes in Economics and Mathematical Systems, pages 78–91. Springer-Verlag, 1991.

[183] F. Herrera, J.L. Castro, and J.L. Verdegay. Solving linear boolean programming problems with imprecise costs. In *Proceeding of the IEEE International Conference on Fuzzy Systems*, pages 1025–1032, San Diego, USA. 1992.

[184] F. Herrera, M. Kovacs, and J.L. Verdegay. Fuzzy linear programming problems with homogeneous linear fuzzy functions. In *Proc. of the Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'92)*, pages 361–364, Palma, Spain. 1992.

[185] F. Herrera, M. Kovacs, and J.L. Verdegay. An optimum concept for fuzzified linear programming problems: A parametric approach. In B. Riecan and M. Duchon, editors, *Proc. of the Int. Conf. on Fuzzy Sets Theory and App.*, volume 1, pages 57–64. Tatra Mountains Math. Pub., 1992.

[186] F. Herrera, M. Kovacs, and J.L. Verdegay. Optimality for fuzzified mathematical programming problems: A parametric approach. *Fuzzy Sets and Systems*, 54(3):279–285, 1993.

[187] F. Herrera, M. Kovacs, and J.L. Verdegay. A parametric approach for (g,p)-fuzzified linear programming problems. *The Journal of Fuzzy Mathematics*, 1(3):699–713, 1993.

[188] F. Herrera and J.L. Verdegay. Three models of fuzzy integer programming. *To appear in European Journal of Operations Research*, 1993.

[189] F. Herrera, J.L. Verdegay, and H.J. Zimmermann. Boolean programming problems with fuzzy constraints. *Fuzzy Sets and Systems*, 55(3):285–293, 1993.

[190] G.W. Hintz and H.-J. Zimmermann. A method to control flexible manufacturing systems. *European Journal of Operational Research*, 41:21–334, 1989. In German.

[191] M. Holtz and Dr. Desonki. Fuzzy-Modell für Instandhaltung. *Unscharfe Modellbildung und Steuerung*, IV:54–62, 1981. In German.

[192] Erik Horstkotte, Cliff Joslyn, and Mark Kantrowitz. comp.ai.fuzzy faq: Fuzzy logic and fuzzy expert systems 1/1 [monthly posting]. URL: ftp://ftp.cs.cmu.edu/afs/cs.cmu.edu/project/-ai-repository/ai/pubs/faqs/fuzzy/fuzzy.faq, January 1994. ⇒ANNOTATION.

[193] Ahmed Ibrahim and Bilal M. Ayyub. Multi-criteria ranking of components according to their priority for inspection. *Fuzzy Sets and Systems*, 48(1):1–14, May 1992. ⇒ANNOTATION.

[194] Norikazu Ikoma et al. Numerical solution of inverse problem of fuzzy model. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[195] Tadashi Imaizumi, Koh-ichi Sugimoto, and Yoshihiro Umeda. Development of fuzzy controller. *Denki Seiko (Electric Furnace Steel)*, 61(1):61–68, January 1990. In Japanese (ISSN: 0011-8389). ⇒ANNOTATION.

[196] M. Inuiguchi, H. Ichihashi, and Y. Kume. Aspect configuration of fuzzy mathematical programming problems. *Trans. System Control and Inf. Soc.*, 2:326–351, 1989.

[197] M. Inuiguchi, H. Ichihashi, and H. Tanaka. Possibilistic linear programming with measurable value function. *ORSA J. on Computing*, 1(3):146–158, 1989.

[198] M. Inuiguchi and H. Ichihashi. Relative modalities and their use in possibilistic linear programming. *Fuzzy Sets and Systems*, 35:303–323, 1990.

[199] M. Inuiguchi, H. Ichihashi, and Y. Kume. A solution algorithm for fuzzy linear programming with piecewise linear membership functions. *Fuzzy Sets and Systems*, 34:15–31, 1990.

[200] M. Inuiguchi, H. Ichihashi, and Y. Kume. Relationships between modality constrained programming problems and various fuzzy mathematical programming problems. *Fuzzy Sets and Systems*, 49:243–259, 1992.

[201] M. Inuiguchi, H. Ichihashi, and Y. Kume. Modality constrained programming problems: A unified approach to fuzzy mathematical programming problems in the setting of possibility theory. *Information Sciences*, 67:93–126, 1993.

[202] H. Ishibuchi and H. Tanaka. Multiobjective programming in optimization of the interval objective function. *European Journal of Operational Research*, 48:219–225, 1990.

[203] Hiroaki Ishii, Minoru Tada, and Teruo Masuda. Two scheduling problems with fuzzy due-dates. *Fuzzy Sets and Systems*, 46(3):339–347, March 1992.

[204] Tatsuo Ishikawa, Noriyoshi Arizono, Yoshihiro Noguchi, Toshiro Saeki, and Shingo Kusumoto. Expert system for scheduling rolling sequence in plate mill. *R&D Kobe Steel Engineering Reports*, 40(3):14–16, July 1990. In Japanese. ⇒ANNOTATION.

[205] C. Isik and S. Ammar. Fuzzy optimal search methods. *Fuzzy Sets and Systems*, 46:331–337, 1992.

[206] S-S. Jang, S-H. Wong, and C-K. Liau. On-line/off-line optimization of complex processes using a linguistic self-organized optimizing control scheme. *Fuzzy Sets and Systems*, 47:23–33, 1992.

[207] G. Kabbara. New utilization of fuzzy optimization method. In M.M. Gupta and E. Sanchez, editors, *Fuzzy Information and Decision Processes*, pages 239–246. North-Holland, Amsterdam, 1982.

[208] J. Kacprzyk and J.W. Owsinski. Nonstandard mathematical programming models including imprecision as a planning tool in an agricultural enterprise operating in varying conditions. In *Proc. Conf. on Organization of Agricultural Enterprises*, Kolobrzeg. 1984.

[209] J. Kacprzyk and S. A. Orlovski, editors. *Optimization Models using Fuzzy Sets and Possibility Theory*. D. Reidel Publishing Co., 1987.

[210] J. Kacprzyk and M. Fedrizzi. *Multiperson Decision Making Using Fuzzy Sets and Possibility Theory*. Kluwer Academic Pub. Co, 1990.

[211] Janusz Kacprzyk. Fuzzy sets and fuzzy logic. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 537–542. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[212] A. Kandel, editor. *Fuzzy Expert Systems.* CRC Press, Boca Raton, Ca., 1992.

[213] Michitaka Kanemoto, Hiroshi Yamane, Tooru Yoshida, and Hideo Tottori. An application of expert system to LD converter processes. *Journal of the Iron and Steel Institute of Japan, ISIJ International*, 30(2):128–135, 1990. ⇒ANNOTATION.

[214] Mark Kantrowitz. comp.ai faqs: 6 parts [monthly posting]. URL: ftp://ftp.cs.cmu.edu/-afs/cs.cmu.edu/project/ai-repository/ai/pubs/faqs/ai/ai_[16].faq and other directories at the same site, January 1994. ⇒ANNOTATION.

[215] Roger M. Kerr and R. N. Walker. A job shop scheduling system based on fuzzy arithmetic. In *Proceedings of the 2nd International Conference on Expert Systems and Leading Edge in Production and Operations Management*, pages 433–450, Hilton Head Island, S.C. Management Science Dept., University of S. Carolina, May 1989. ⇒ANNOTATION.

[216] Roger M. Kerr. Implementation of expert systems for production scheduling - three case studies. In *Proceedings of the 1st World Congress on Expert Systems*, pages 405–414. Pergamon Press, December 1991.

[217] Roger M. Kerr. Research issues and challenges in fuzzy scheduling. In Roger Kerr, editor, *Workshop Notes of the Workshop on Fuzzy Scheduling Systems*, Linz, Austria. University of Linz, Department of Mathematics, June 1993.

[218] Roger M. Kerr. Scheduling in the steel industry. CD-Studie 94/16, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994.

[219] Roger M. Kerr and Wolfgang Slany. Research issues and challenges in fuzzy scheduling. CD-Technical Report 94/68, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/riacifs.ps.Z.

[220] M. King and A. Mercer. The optimum markup when bidding with uncertain costs. *European Journal of Operational Research*, 47:348–363, 1990.

[221] Erich Peter Klement and Wolfgang Slany, editors. *Fuzzy Logic in Artificial Intelligence. Proceedings of the 8th Austrian Artificial Intelligence Conference, FLAI'93, Linz, Austria, June 1993*, volume 695 of *Lecture Notes in Artificial Intelligence*. Springer Verlag Berlin Heidelberg, 1993. A conference report by Thalhammer [399] is available on the net: URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/flai.txt.

[222] Erich P. Klement and Wolfgang Slany. Fuzzy logic in artificial intelligence. CD-Technical Report 94/67, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994.

[223] L.T. Koczy. Fuzzy graphs in the evaluation and optimization of networks. *Fuzzy Sets and Systems*, 46:307–319, 1992.

[224] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3–34, 1992. ⇒ANNOTATION.

[225] W. Kolodziejczyk. On equivalence of two optimization methods for fuzzy discrete programming problems. *European Journal of Operational Research*, 36:85–91, 1988.

[226] H. Kominami, S. Naitoh, N. Kamada, Ch. Hamaguchi, T. Tanaka, and H. Endoh. Breakout forecasting system by neural network for continuous casting process. *Seitetsu Kenkyu (Nippon Steel Research Journal), Special Issue on Instrument and Control Engineering*, 339:31–34, 1990. In Japanese.

[227] Ken-ichi Kosaka et al. The application of fuzzy linear programming to flow control of crossing gate network. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[228] Bart Kosko. *Neural Networks and Fuzzy Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[229] M. Kovacs. A concept of optimality for fuzzified linear programming problems. In M. Fedrizzi, J. Kacprzyk, and M. Roubens, editors, *Interactive Fuzzy Optimization*, Lecture Notes in Economics and Mathematical Systems, pages 36–44. Springer-Verlag, 1991.

[230] M. Kovacs. Stable embedding of ill-posed linear equality and inequality systems into fuzzified systems. *Fuzzy Sets and Systems*, 45:305–312, 1992.

[231] R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge Based Systems – Numerical Methods.* Springer Verlag, 1991.

[232] Rudolf Kruse, Jörg Gebhardt, and Frank Klawonn. *Fuzzy-Systeme*. B.G. Teubner Stuttgart, 1993. In German.

[233] Thomas Länger. Fuzzy and possibilistic approaches towards scheduling applications. Diplomarbeit ($\simeq$ master's thesis), Technical University of Vienna, 1994. In English.

[234] Y-J. Lai and C-L. Hwang. Interactive fuzzy linear programming. *Fuzzy Sets and Systems*, 45:169–183, 1992.

[235] Y-J. Lai and C-L. Hwang. A new approach to some possibilistic linear programming problems. *Fuzzy Sets and Systems*, 49:121–133, 1992.

[236] Y-J. Lai and C-L. Hwang. *Fuzzy Mathematical Programming. Methods and Applications*. Number 394 in Lecture Notes in Economics and Mathematical Systems. Springer Verlag, 1992.

[237] Y-J. Lai and C-L. Hwang. Possibilistic linear programming for managing interest rate risk. *Fuzzy Sets and Systems*, 54:135–146, 1993.

[238] Jérôme Lang. Possibilistic logic as a logical framework for min-max discrete optimisation problems and prioritized constraints. In *FAIR'91*, 1991.

[239] H. Leberling. On finding compromise solutions in multi-criteria problems using the fuzzy min-operator. *Fuzzy Sets and Systems*, 6:105–118, 1981.

[240] E. Stanley Lee and R.J. Li. Fuzzy multiple objective programming and compromise programming with Pareto optimum. *Fuzzy Sets and Systems*, 53:275–288, 1993.

[241] A. Lehtimäki. A fuzzy decision aid for production control. In *Second IFSA Congress*, pages 628–632, Tokyo, Japan. July 1987.

[242] R.J. Li and E.S. Lee. Fuzzy approaches to multicriteria De Novo programs. *J. Math. Anal. Appl.*, 153:97–111, 1990.

[243] R.J. Li and E.S. Lee. An exponential membership function form fuzzy multiple objective linear programming. *Comp. Math. App.*, 22(12):55–60, 1991.

[244] R.J. Li and E.S. Lee. De Novo programming with fuzzy coefficients and multiple fuzzy goals. *J. of Mathematical Analysis and Applications*, 172:212–220, 1993.

[245] Hans-Peter Lipp. *Die Anwendung der unscharfen Mengentheorie für ein Steuerungskonzept zur operativen Führung komplexer Systeme*. Dissertation A, TH Karl-Marx-Stadt (Chemnitz), 1980. In German.

[246] Hans-Peter Lipp. Anwendung eines Fuzzy-Petri-Netzes zur Beschreibung von Koordinierungssteuerungen in komplexen Produktionssystemen. *Wissenschaftliche Zeitschrift der TU Chemnitz*, 24(5):633–639, 1982. In German.

[247] Hans-Peter Lipp. *Ein Konzept eines unscharfen Petri-Netzes als Grundlage für operative Entscheidungsprozesse in komplexen Produktionssystemen*. Dissertation B (Habilitationsschrift, TU Chemnitz), TU Karl-Marx-Stadt (Chemnitz), 1989. In German.

[248] Hans-Peter Lipp, Rolf Günther, and P. Sonntag. Unscharfe Petri-Netze – Ein Basiskonzept für computerunterstützte Entscheidungsprozesse in komplexen Systemen. Wissenschaftliche Schriftenreihe 7, TU Chemnitz, 1989. In German.

[249] Hans-Peter Lipp. Flexible Fertigungssysteme nach stückflußabhängigen und zeitabhängigen Kriterien steuern. *ZWF CIM*, 12:625–655, 1990. In German.

[250] Hans-Peter Lipp and B. Fischer. Entwurf von Fuzzy-Petri-Netzen für die operative Führung flexibler Fertigungssysteme. *Wissenschaftliche Zeitschrift der TU Chemnitz*, 33(6):825–833, 1991. In German.

[251] Hans-Peter Lipp. Die operative Montagesteuerung auf der Grundlage von zeitbewerteten Fuzzy-Petri-Netzen. *Wissenschaftliche Zeitschrift der TU Chemnitz*, 33(6):835–842, 1991. In German.

[252] Hans-Peter Lipp. Application of timed fuzzy Petri nets in expert systems for operative management of complex production systems. In G. Hommel, editor, *Prozessrechensysteme '91. (Process Computer Systems '91) Proceedings*, pages 103–112, Berlin, Heidelberg, New York. Springer-Verlag, 1991.

[253] Hans-Peter Lipp. Erweiterung des Fuzzy-Control-Konzeptes für Steuerungsaufgaben in der Verfahrensindustrie. In *Anwendersymposium zu Fuzzy-Technologien, 8.-9.9.1992, Proceedings*, Korneliuscenter, Promenade 9, D-52076 Aachen, Germany. MIT GmbH, 1992. In German.

[254] Hans-Peter Lipp. Einsatz von Fuzzy-Konzepten für das operative Produktionsmanagement. *Automatisierungstechnische Praxis*, 34(12), 1992. In German. ⇒ANNOTATION.

[255] H.-P. Lipp. Ergebnis- und zeitabhängige Simulation flexibler Fertigungssysteme mittels Fuzzy Petri-Netzen. *VDI–Berichte*, 989:181–198, 1992. In German.

[256] H.-P. Lipp. Einsatz von Fuzzy Konzepten für das operative Produktionsmanagement. *atp-Automatisierungstechnische Praxis*, 34(12):668–675, 1992. In German.

[257] H.-P. Lipp and W. Ringelband. Einsatz eines Modells auf Basis unscharfer Petri–Netze zur Disposition des Pfanneneinsatzes im Stahlwerk. In *Fuzzy Logic, IIR-Konferenz*, München. May 1992. In German.

[258] H.-P. Lipp. Flexible Produktionsführung durch unscharfe Produktionsprozessregelung. *Spektrum der Wissenschaft*, 3, 1993. In German.

[259] H.-P. Lipp. Mit Unschärfe mehr Flexibilität. *Superelectronics Jahrbuch*, 1993. In German.

[260] J. Llena. On fuzzy linear programming. *European Journal of Operational Research*, 22:216–223, 1985.

[261] W.A. Lodwick. Analysis of structure in fuzzy linear programs. *Fuzzy Sets and Systems*, 38:15–26, 1990.

[262] M.K. Luhandjula. Linear programming under randomness and fuzziness. *Fuzzy Sets and Systems*, 10:57–63, 1983.

[263] M.K. Luhandjula. Linear programming with a possibilistic objective function. *European Journal of Operational Research*, 31:110–117, 1987.

[264] M.K. Luhandjula. Fuzzy optimization: An appraisal. *Fuzzy Sets and Systems*, 30:257–282, 1989.

[265] M.K. Luhandjula, H. Ichihashi, and M. Inuiguchi. Fuzzy and semi-infinite mathematical programming. *Information Sciences*, 61:233–250, 1992.

[266] Yunosuke Maki, Yasuo Masuda, T. Sawada, T. Matsumoto, Hiroshi Obata, and N. Takashima. Application of fuzzy theory for automatic control of hot stove combustion gas flow. In *Proceedings of the 6th IFAC Symposium on Automation in Mining, Mineral and Metal Processing (IFAC MMM 1989)*, pages 278–284, Buenos Aires, Argentina. September, 4–8 1989. ⇒ANNOTATION.

[267] Yunosuke Maki, Nobuhiro Takashima, Hiroshi Obata, Osamu Iida, Kazuma Nakashima, and Toshiro Sawada. Application of fuzzy theory to ironmaking process control. *Kawasaki Steel Technical Report*, 22(3):54–60, 1990. In Japanese. ⇒ANNOTATION.

[268] Robert J. Marks II. Intelligence: Computational versus artificial. *IEEE Transactions on Neural Nets*, September 1993. ⇒ANNOTATION.

[269] R.E. Marsten and T.L. Morin. Parametric integer programming: The right-hand-side case. *Annals of Discrete Math.*, 1:375–390, 1977.

[270] Roger Martin-Clouaire. Dealing with soft constraints in a constraint satisfaction problem. In *Proceedings of the 4th International Conference on Information Processing of Uncertainty in Knowledge-Based Systems*, pages 37–40, Palma de Mallorca, Spain. July 1992.

[271] Roger Martin-Clouaire. CSP techniques with fuzzy linear contraints: practical issues. In *Proc. of the First European Congress on Fuzzy and Intelligent Technologies (EUFIT)*, Aachen, Germany. September 1993.

[272] N. Mathé. Prise en compte de l'imprécision des délais dans la construction d'ordonnancements prévisionnels. Master's thesis, Univ. P. Sabatier, DEA d'automatique, ENSAE, Toulouse, June 1987. In French.

[273] Luděk Matyska. Logic programming with fuzzy sets. Technical Report TCU/CS/1993/4, Dept. of Computer Science, City Univ., London, UK, December 1993. URL: ftp://ftp.cs.city.ac.uk/papers/93/cs93-4.ps.Z. ⇒ANNOTATION.

[274] C. McCahon and E.S. Lee. Fuzzy job sequencing for a flow shop. *European Journal of Operational Research*, 62:294–301, 1992.

[275] Daniel McNeill and Paul Freiberger. *Fuzzy Logic.* Simon & Schuster, 1993.

[276] S. Minton, M. Johnston, A. Philips, and P. Laird. Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 17–24, Menlo Parc. 1989.

[277] Steven Minton, Mark D. Johnston, Andrew B. Philips, and Philip Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:161–205, 1992.

[278] Hiroto Mizunuma et al. Solution using genetic algorithm for fuzzy mixed integer programming problems. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[279] S.V. Mohandas, T.A. Phelps, and K.M. Ragsdell. Structural optimization using a fuzzy goal programming approach. *Computers & Structures*, 37(1):1–8, 1990.

[280] Myung Geun Chun and Zeung Nam Bien. A model with an inference engine for a fuzzy production system using fuzzy Petri nets. *Journal of the Korean Institute of Telematics and Electronics*, 29B(7):508–519, July 1992. In Korean. ⇒ANNOTATION.

[281] N. Nagai et al. Application of expert systems to forecasting heat level at Kobe no. 3 blast furnace. *CAMP-ISIJ*, 2:10–13, 1989. In Japanese. ⇒ANNOTATION.

[282] Yozo Nakahara et al. New ranking trapezoidal fuzzy numbers and its application to linear programming problems. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[283] K. Nakamura. Some extension of fuzzy linear programming. *Fuzzy Sets and Systems*, 14:211–229, 1984.

[284] Setsuko Nakamura et al. Several definitions of fuzzy inequality and their applications to linear programming. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[285] R. Narasimhan. Goal programming in a fuzzy environment. *Decision Sci.*, 11:325–336, 1980.

[286] V. Narsimha, R.N. Tiwari, and K.S. Sastri. Solution of optimal control problems with lumped parameters having single or multiple goal objectives in fuzzy environment. *Fuzzy Sets and Systems*, 48:173–184, 1992.

[287] V. Narsimha, R.N. Tiwari, and K.S. Sastri. Dynamic programming approach to multiple objective control problem having deterministic or fuzzy goal. *Fuzzy Sets and Systems*, 57:195–202, 1993.

[288] R. Narsimhan. On fuzzy goal programming – some comments. *Decision Sci.*, 12:532–538, 1981.

[289] R.M. Nauss. *Parametric Integer Programming.* University of Missouri Press, 1979.

[290] C.V. Negoita and M. Sularia. On fuzzy programming and tolerances in planning. *Econom. Comp. Econom. Cybernet. Stud. Res.*, 1:3–15, 1976.

[291] C.V. Negoita, P. Flondor, and M. Sularia. On fuzzy environment in optimization problems. In J. Rose and C. Bilciu, editors, *Modern Trends in Cybernetic and Systems.* Springer-Verlag, 1977.

[292] C.V. Negoita and D. Ralescu. On fuzzy optimization. *Kybernetes*, 6:193–195, 1977.

[293] C.V. Negoita. The current interest in fuzzy optimization. *Fuzzy Sets and Systems*, 6:261–269, 1981.

[294] Constantin V. Negoiţă. *Expert Systems and Fuzzy Systems.* Benjamin/Cummings, 1985.

[295] I. Nishizaki and M. Sakcuva. Two-person zero-sum games with multiple fuzzy goals. *Japanese Journal of Fuzzy Theory and Systems*, 4(3):289–300, 1992.

[296] Kohei Nomoto et al. Terrain avoidance flight path planning based on learning fuzzy reasoning. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[297] Vilém Novák. Fuzzy logic: Application to natural language. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 515–521. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[298] Masayuki Numao and Shin-ichi Morishita. Scheplan — a scheduling expert for steel-making process. In *Proceedings of the International Workshop on Artificial Intelligence for Industrial Applications*, pages 467–472, Hitachi City, Japan. IEEE AI Cat. No.88CH2529–6; New York, USA, 25–27 May 1988. ⇒ANNOTATION.

[299] Masayuki Numao and Shin-ichi Morishita. A scheduling environment for steel-making processes. In *Proceedings of the 5th Conference on Artificial Intelligence Applications*, pages 279–286, Miami, FL, USA. IEEE Comput. Soc. Press IEEE Cat. No.89CH2712–8, Washington, DC, USA, 6–10 March 1989. Identical to [300].

[300] Masayuki Numao and Shin-ichi Morishita. Cooperative scheduling and its applications to steel-making processes. *IEEE Transactions on Industrial Electronics*, 1990. ⇒ANNOTATION.

[301] Hiroshi Obata, Makoto Nomura, Yunosuke Maki, Yuu Kobayashi, and Yoshiaki Itoh. Development of blast furnace heat control system using neural network and fuzzy inference. *CAMP-ISIJ*, 9, 1990. In Japanese. ⇒ANNOTATION.

[302] Tosio Odanaka et al. On a fuzzy stochastic dynamic programming. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[303] Yasutsugu Ogura and Hiroyuki Hamada. Total production control system of BF–MILL with expert system in NKK. In Georg Gottlob and Wolfgang Nejdl, editors, *Proceedings of the International Workshop in Vienna Expert Systems in Engineering: Principles and Applications*, September 1990. ⇒ANNOTATION.

[304] M. Oh'Eigeartaigh. A fuzzy transportation algorithm. *Fuzzy Sets and Systems*, 8:235–245, 1982.

[305] Shinkoh Okada et al. Order relation between intervals and its application to shortest path problem. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[306] V. Oley, J. Strelec, and J. Chmurny. Analysis of deterministic, stochastic and fuzzy discrete systems by generalized Petri net. In T. Terano, M. Sugeno, M. Mukaidono, and K. Shigemasu, editors, *Fuzzy Engineering Toward Human Friendly Systems*, pages 230–238, Amsterdam, Netherlands. IOS Press, 1991. ⇒ANNOTATION.

[307] S.A. Orlovski. On programming with fuzzy constraint sets. *Kybernetes*, 6:197–201, 1977.

[308] S.A. Orlovski. On formulization of a general fuzzy mathematical problem. *Fuzzy Sets and Systems*, 3:311–321, 1980.

[309] S.A. Orlovski. Multiobjective programming problems with fuzzy parameters. *Control and Cybernetics*, 4:175–184, 1984.

[310] W. Ostasiewicz. A new approach to fuzzy programming. *Fuzzy Sets and Systems*, 7:139–152, 1982.

[311] R. Ostermark. Fuzzy linear constraints in the capital asset pricing model. *Fuzzy Sets and Systems*, 30:93–102, 1989.

[312] J.W. Owsinski, S. Zadrozny, and J. Kacprzyk. Analysis of water use and needs in agricultural through a fuzzy programming model. In J. Kacprzyk and S.A. Orlovski, editors, *Optimization Models using Fuzzy Sets and Possibility Theory*, pages 377–395. D. Reidel Publishing Co., 1987.

[313] S. S. Panwalkar and W. Iskander. A survey of scheduling rules. *Operations Research*, 25(1):45–61, 1977.

[314] J.B. Pickens. Fuzzy goal programming in forestry: An application with special solution problems. *Fuzzy Sets and Systems*, 39:239–246, 1991.

[315] Henri Prade. Using fuzzy set theory in a scheduling problem: A case study. *Fuzzy Sets and Systems*, 2(2):153–165, 1979.

[316] Henri M. Prade. Operations research with fuzzy data. In P.P. Wang and S.K. Chang, editors, *Fuzzy Sets. Theory and Applications to Policy Analysis and Information Systems*, pages 155–169. Plenum Press, 1980.

[317] Patrick Prosser. A reactive scheduling agent. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1004–1009, 1989.

[318] Patrick Prosser. Scheduling as a constraint satisfaction problem: Theory and practice. In *Scheduling of Production Processes (ECAI'92 Workshop)*, pages 7–15. ECCAI, August 1992.

[319] Christian Rabetge. *Fuzzy Sets in der Netzplantechnik*. PhD thesis, Göttingen, 1990. Also appeared as a book from Dt. Univ.-Verl., Wiesbaden, 1991.

[320] J. Ramik and H. Rommelfanger. A single- and a multi-valued order on fuzzy numbers and its use in linear programming with fuzzy coefficients. *Fuzzy Sets and Systems*, 57:203–208, 1993.

[321] J.R. Rao, R.N. Tiwari, and B.K. Mohanty. A preference structure on aspiration levels in a goal programming problem – a fuzzy approach. *Fuzzy Sets and Systems*, 25:175–182, 1988.

[322] J.R. Rao and K. Saraswati. Comments on: Facility location problem on a network under multiple criteria-fuzzy set theoretic approach. *Int. J. of Systems Sci.*, 21(11):2387–2391, 1990.

[323] D.B. Rinks. A heuristic approach to aggregate production scheduling using linguistic variables. In R.R. Yager, editor, *Fuzzy Set and Possibility Theory: Recent Developments*. Pergamon Press, New York, 1982.

[324] W. Rodder and H.J. Zimmermann. Duality in fuzzy programming. In *Int. Symp. on Extremal Methods and Systems Analysis*, University of Texas, Austin, TX. 1977.

[325] W. Rodder and H.J. Zimmermann. Duality in fuzzy linear programming. In A.V. Fiacco and K.O. Kortanek, editors, *Extremal Methods and Systems Analyses*, pages 415–429. Springer Verlag Berlin, Heidelberg, New York., 1980.

[326] H. Rommelfanger, R. Hanuscheck, and J. Wolf. Linear programming with fuzzy objective functions. In *Proc. of the I IFSA Congress*, Palma (Spain). 1985.

[327] M. Rommelfanger. Interactive decision making in fuzzy linear optimization problems. *European Journal of Operational Research*, 41:210–217, 1989.

[328] M. Rommelfanger, R. Hanuscheck, and J. Wolf. Linear programming with fuzzy objectives. *Fuzzy Sets and Systems*, 29:31–48, 1989.

[329] H. Rommelfanger. FULPAL: An interactive method for solving (multiobjective) fuzzy linear programming problems. In R. Słowiński and J. Teghem Jr., editors, *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, chapter 5, pages 279–299. Kluwer Academic Publishers, Dordrecht, 1990.

[330] Azriel Rosenfeld, Robert A. Hummel, and Steven W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6):173–184, 1976.

[331] M. Roubens. Inequality constraints between fuzzy numbers and their use in mathematical programming. In R. Słowiński and J. Teghem Jr., editors, *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, chapter 7, pages 321–330. Kluwer Academic Publishers, Dordrecht, 1990.

[332] Marc Roubens and Jacques Teghem Jr. Comparison of methodologies for fuzzy and stochastic multi-objective programming. *Fuzzy Sets and Systems*, 42:119–132, 1991.

[333] P.A. Rubin and R. Narasimhan. Fuzzy goal programming with nested priorities. *Fuzzy Sets and Systems*, 14:115–129, 1984.

[334] T. L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, 1980.

[335] Norman Sadeh and Mark S. Fox. Variable and value ordering heuristics for job shop scheduling. In *Proceedings 4th International Conference on Expert Systems and the Leading Edge in Production and Operations Management*, pages 134–144, Hilton Head Island. 1990.

[336] T. Saito. Application of artificial intelligence in the Japanese steel industry. In *Proceedings of the 6th IFAC Symposium on Automation in Mining, Mineral and Metal Processing (IFAC MMM)*, pages 30–38, Buenos Aires, Argentina. September 1989. ⇒ANNOTATION.

[337] M. Sakawa and H. Yano. An interactive fuzzy satisficing method for multiobjective linear fractional programming problems. *Fuzzy Sets and Systems*, 28:129–144, 1988.

[338] M. Sakawa. Interactive decision making for multiobjective nonlinear programming problens with fuzzy parameters. *Fuzzy Sets and Systems*, 29:315–326, 1989.

[339] M. Sakawa and H. Yano. An interactive fuzzy satisficing method for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 30:221–238, 1989.

[340] M. Sakawa and H. Yano. An interactive fuzzy satisficing multiobjective linear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 35:125–142, 1990.

[341] M. Sakawa, H. Yano, and J. Takahashi. Interactive decision making for multiobjective linear programming problems with fuzzy parameters based on solution concept incorporating fuzzy goals. *Journal of Japan Society for Fuzzy Theory and Systems*, 2(4, 5):66–577, 1990.

[342] M. Sakawa and H. Yano. Feasibility and Pareto optimality for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 43:1–15, 1991.

[343] M. Sakawa and H. Yano. Interactive decision making for multiobjective linear fractional programming problems with fuzzy parameters based on solution concepts incorporating fuzzy goals. *Japanese Journal of Fuzzy Theory and Systems*, 3(1):45–62, 1991.

[344] M. Sakawa and H. Yano. New solution concepts for multi-objective nonlinear programming problems with fuzzy coefficients taking account of fuzzy goals. *Japanese Journal of Fuzzy Theory and Systems*, 3(2):133–148, 1991.

[345] M. Sakawa, H. Yano, and J. Takahashi. Pareto optimality for multiobjective linear fractional programming problems with fuzzy parameters. *Information Sciences*, 63:33–53, 1992.

[346] M. Sakawa. *Fuzzy Sets and Interactive Multiobjective Optimization.* Applied Information Technology. Plenum Press, 1993.

[347] M. Sakawa et al. Hot parts operating schedule by genetic algorithms and fuzzy modeling. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[348] Masatoshi Sakawa et al. Fuzzy programming for large-scale multiobjective linear programming with block angular structure. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[349] Y. Sasabe, S. Kubota, A. Koyama, and H. Miki. Real-time expert system applied to mold bath level control of continuous caster. *ISIJ International*, 30(2):136–141, 1990.

[350] M. Sasaki, M. Gen, and K. Lda. Interactive sequential fuzzy goal programming. *Computers and Industrial Engineering*, 19:567–571, 1990.

[351] Ken Satoh. Formalizing soft constraints by interpretion ordering. In *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI 90)*, pages 585–590, 1990. ⇒ANNOTATION.

[352] Thomas Schiex. Possibilistic constraint satisfaction problems or "How to handle soft constraints ?". In *8^{th} International Conference on Uncertainty in Artificial Intelligence*, pages 268–275, Stanford, CA. July 1992.

[353] Günter Schmidt. How to apply fuzzy logic to reactive production scheduling. In Roger M. Kerr and Elizabeth Szelke, editors, *Preprints of the IFIP TC 5 / WG5.7 Int. Workshop on Knowledge Based Reactive Scheduling*, pages 56–66, Athens, Greece. October 1993.

[354] Brian Schott and Thomas Whalen. Symbolic structures, duality, and fuzzy mathematics in scheduling decision support. In *Proceedings of the '84 IEEE Int'l Conference on Systems, Man, & Cybernetics*, pages 73–77, 1984.

[355] B. Selman, H. J. Levesque, and D. Mitchell. A new method for solving hard satisfyability problems. In *Proceedings of the AAAI'92 Conference*, San Jose, Ca. 1992.

[356] B. Shafai and G. Sotirov. Uniqueness of solution in FLP under parameter perturbations. *Fuzzy Sets and Systems*, 34:179–186, 1990.

[357] M. J. Shah, R. Damian, and J. Silverman. Knowledge based dynamic scheduling in a steel plant. In *Proceedings of the 6th International Conference on Artificial Intelligence for Industrial Applications*, pages 108–113, St. Barbara. 1990.

[358] Stuart C. Shapiro, editor. *Encyclopedia of Artificial Intelligence.* John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992.

[359] Takanori Shibata et al. Robotic motion planning by genetic algorithm with fuzzy critic. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[360] Günther Skele. Repair Based Scheduling mit Tabu Search. Diplomarbeit ($\simeq$ master's thesis), Technical University of Vienna, 1994. In German.

[361] Wolfgang Slany[1]. Artificial Intelligence Trends in Japan 1991. CD-Studie 91/5, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, September 1991. In German.

[362] Wolfgang Slany, editor. Knowledge based scheduling. CD-Studie 92/6, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, January 1992. ⇒ANNOTATION.

[363] Wolfgang Slany, Christian Stary, and Jürgen Dorn. Uncertainty management in production scheduling applied to high-grade steelmaking. In *6. Workshop Planen und Konfigurieren*, FORWISS-Report, pages 51–60, Erlangen München Passau. Bayrisches Forschungszentrum für Wissensbasierte Systeme, March 1992.

[364] Wolfgang Slany, Christian Stary, and Jürgen Dorn. Vague data management in production process scheduling applied to high-grade steelmaking. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, pages 214–221, University of Maryland. Morgan Kaufmann Publishers, Inc., June 1992. ⇒ANNOTATION.

[365] Wolfgang Slany. Fuzzy expert system for maintenance interval prediction. In Franz Moser, Hans Schnitzer, and Hansjörg Bart, editors, *European Symposium on Computer Aided Process Engineering-3. ESCAPE-3, 25th European Symposium of the Working Party on Computer Aided Process Engineering, 494th Event of the European Federation of Chemical Engineering (EFChE)*, Supplement to Computers & Chemical Engineering, pages S155–S159, Graz, Austria. Pergamon Press Ltd, July 1993. ⇒ANNOTATION.

[366] Wolfgang Slany. Fuzzy constraint relaxation techniques for knowledge-based scheduling. In Hans-Jürgen Zimmermann, editor, *EUFIT'93, First European Congress on Fuzzy and Intelligent Technologies*, pages 1124–1127, Aachen, Germany. Augustinus Buchhandlung, September 1993. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/eufit93.ps.Z. ⇒ANNOTATION.

[367] Wolfgang Slany. Fuzzy expert system to predict maintenance intervals in a continuous caster. In Duk-Hyon Baik, editor, *Preprints of the international conference CPC-93, Computerized Production Control in Steel Plants*, pages 291–296, Seoul, Korea. The Korean Institute of Metals and Materials, The Institute of Materials, UK., November 1993. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/fespmicc.ps.Z. ⇒ANNOTATION.

[368] Wolfgang Slany. Fuzzy constraint relaxation techniques for knowledge-based scheduling. In Roger Kerr, editor, *Workshop Notes of the Workshop on Fuzzy Scheduling Systems*, Linz, Austria. University of Linz, Department of Mathematics, June 1993. ⇒ANNOTATION.

[369] Wolfgang Slany. Scheduling as a fuzzy multiple criteria optimization problem. CD-Technical Report 94/62, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/cd-tr9462.ps.Z. ⇒ANNOTATION.

[370] Wolfgang Slany. Scheduling with fuzzy decision making methods. In Mario Fedrizzi, Erich P./ Klement, Aldo Ventre, and Alessandro Zorat, editors, *Proceedings of CIFT'94: Current Issues in Fuzzy Technologies: Decision Models and Systems*, Trento, Italy. June 1994.

[371] Wolfgang Slany. Fuzzy scheduling. CD-Technical Report 94/66, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994. This is the thesis you are reading. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/cd-tr9466.ps.Z.

[372] Roman Słowiński. A multicriteria fuzzy linear programming method for water supply system development planning. *Fuzzy Sets and Systems*, 19:217–237, 1986.

[373] R. Słowiński and J. Teghem Jr. Fuzzy versus stochastic approaches to multicriteria linear programming under uncertainty. *Naval Research Logistics*, 35:673–695, 1988.

[374] R. Słowiński and J. Teghem Jr., editors. *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty.* Kluwer Academic Publishers, Dordrecht, 1990.

[1] Publications by Slany not related to the thesis's topic can be found in the Curriculum Vitae from page 160.

[375] R. Słowiński. FLIP: An interactive method for multiobjective linear programming with fuzzy coefficients. In R. Słowiński and J. Teghem Jr., editors, *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, chapter 3, pages 249–262. Kluwer Academic Publishers, Dordrecht, 1990.

[376] Stephen F. Smith, Mark S. Fox, and Peng Si Ow. Constructing and maintaining detailed construction plans: Investigations into the development of knowledge-based factory scheduling systems. *AI Magazine*, 7(4):45–61, Fall 1986.

[377] Stephen F. Smith. Knowledge-based production management: approaches, results and prospects. Technical Report CMU-RI-TR-91-21, Carnegie-Mellon Univ. Robotics Institute, December 1991. ⇒ANNOTATION.

[378] D. G. Smith and E. Poulard. Fuzzy logic in vehicular routing. In Roger Kerr, editor, *Workshop Notes of the Workshop on Fuzzy Scheduling Systems*, Linz, Austria. University of Linz, Department of Mathematics, June 1993.

[379] Paul Snow and Eugene C. Freuder. Improved relaxation and search methods for approximate constraint satisfaction with a maximin criterion. In Peter F. Patel-Schneider, editor, *Proc. of the 8^{th} biennal conference of the Canadian society for computational studies of intelligence*, pages 227–230, Ottawa, Ontario, Canada. May 1990.

[380] G. Sommer and M.A. Pollatschek. A fuzzy programming approach to an air pollution regulation problem. *European Journal of Operational Research*, 10:303–313, 1978.

[381] P. Srinivasan and D. Gracanin. Approximate reasoning with fuzzy Petri nets. In *Second IEEE International Conference on Fuzzy Systems, San Francisco, California, 28.3.–1.4.1993.* IEEE, 1993.

[382] R.E. Steuer. Algorithms for linear programming problems with interval objective function coefficients. *Mathematics of Operations Research*, 6(3):333–348, 1981.

[383] K. Stohl, W. Snopek, T. Weigert, and T. Moritz. Development of a scheduling expert system for a steelplant. In *Proceedings of the IFIP Conference on Experts Systems in Mineral and Metallurgy*, Finland. 1991.

[384] H. S. Stone and J. M. Stone. Efficient search techniques – an empirical study of the n-queens problem. *IBM Journal of Research and Development*, 31:464–474, 1987.

[385] R. Takahashi. Control systems in the steel rolling process. *Journal of the Society of Instrumental and Control Engineering (Japan)*, 27(4):325–332, April 1988. In Japanese. ⇒ANNOTATION.

[386] T. Takahashi, M. Konishi, S. Tamura, H. Hanaoka, and H. Nakagawa. Scheduling for steel making process using mathematical programming method. In *Proceedings of the 6th IFAC Symposium on Automation in Mining, Mineral and Metal Processing (IFAC MMM 1989)*, pages 133–138, Buenos Aires, Argentina. September 1989. ⇒ANNOTATION.

[387] Tetsuya Takahashi, Kiyoshi Ebina, Masayoshi Saegusa, Hiroyuki Nakagawa, and Hirotaka Hanaoka. Scheduling system for steelmaking process using mathematical programming method. *R&D Kobe Steel Engineering Reports*, 40(3):29–32, July 1990. In Japanese. ⇒ANNOTATION.

[388] E. Takeda and N.T. Nishida. Multiple criteria decision making with fuzzy domination structures. *Fuzzy Sets and Systems*, 3:123–136, 1980.

[389] A. Takekoshi, T. Aoki, I. Takahashi, Hakaru Tomishima, and Masato Yoshino. Application of knowledge engineering for iron and steel making works. *NKK Technical Report*, 125:2–9, January 1989. In Japanese. ⇒ANNOTATION.

[390] A. Takekoshi, T. Aoki, I. Takahashi, H. Tomishima, and M. Yoshino. Application of knowledge engineering for iron and steel making works. *NKK Technical Review*, 56:1–9, 1989. ⇒ANNOTATION.

[391] Atsuhisa Takekoshi, Mamoru Inaba, Yukinori Satoh, Shoji Hamaya, Kazuhiro Kikuchi, and Hiroki Furukawa. Raw material yard expert system at Fukuyama works. *NKK Technical Review*, 133:1–8, 1990. In Japanese. ⇒ANNOTATION.

[392] Naoki Tamura, Kouichi Matsuda, Yoshiyuki Matsui, Hiroaki Nakano, Shinji Kitano, and Korehito Kadoguchi. Expert system for blast furnace heat level control. *R&D Kobe Steel Engineering Reports*, 40(3):9–13, July 1990. In Japanese. ⇒ANNOTATION.

[393] H. Tanaka, T. Okuda, and K. Asai. On fuzzy mathematical programming. *Journal of Cybernetics*, 3:37–46, 1974.

[394] H. Tanaka and K. Asai. Fuzzy linear programming problems with fuzzy numbers. *Fuzzy Sets and Systems*, 13:1–10, 1984.

[395] H. Tanaka, H. Ichihashi, and K. Asai. A formulation of fuzzy linear programming problems based on comparison of fuzzy numbers. *Control and Cybernetics*, 13:185–194, 1984.

[396] Shuichi Taniyoshi, Tsunehiro Kaneda, Teruo Tamura, Daisuke Onada, Yasuhiro Miyazaki, and Tsuyoshi Ikeda. Computer control system for ore yard operation at Mizushima works. *Kawasaki Steel Giho*, 22(1):6–11, 1990. In Japanese. ⇒ANNOTATION.

[397] C.G. Tapia and B.A. Murtagh. Interactive fuzzy programming with preference criteria in multiobjective decision-making. *Computers and Operational Research*, 18:307–316, 1991.

[398] C. Tapia and B.A. Murtayes. Interactive group decision-making using fuzzy programming with preference criteria. *Fuzzy Sets and Systems*, 45:13–23, 1992.

[399] Gabi Thalhammer. Conference report: FLAI'93: 8th Austrian conference on fuzzy logic in artificial intelligence. *AI Communications: the European Journal on Artificial Intelligence*, 7(1):60–64, March 1994. The proceedings of the conference can be found in [221]. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/flai.txt.

[400] Akihiko Tokumaru et al. Heuristic approach to multi-objective decision making problems in fuzzy environment. In *Proceedings of the 9th Fuzzy System Symposium*, Sapporo Japan. May 1993. In Japanese. ⇒ANNOTATION.

[401] Yuusaku Tominaga and Ryoichi Kouda. Delphi approach using fuzzy sets reasoning to assist rock identification. In *Proceedings of ISME-AI'90*, pages 97, 202–206. Branch of Exploration Engineering (BEE), Mining and Materials Processing Institute of Japan (MMIJ), October 1990. ⇒ANNOTATION.

[402] Y. Tsujimura, S.H. Park, I.S. Chang, and M. Gen. An effective method for solving flow shop scheduling problems with fuzzy processing times. *Computers and Industrial Engineering*, 25(1-4):239–242, 1993.

[403] Alan M. Turing. Intelligent machinery. In D. C. Ince, editor, *Mechanical Intelligence*, Collected Works of A. M. Turing. North-Holland, 1992. Original paper appeared in B. Meltzer and D. Michie (Editors), *Machine Intelligence,* 5:3–23, 1969, Edinburgh University Press, but was actually written as early as in 1948.

[404] I. B. Türksen. Fuzzy sets and their application in production research. In H.J. Bullinger and H.J. Warnecke, editors, *Proc. ICPR, Toward the Factory of the Future*, pages 649–656, 1985.

[405] I. B. Türksen. Approximate reasoning for production planning. *Fuzzy Sets and Systems*, 26:1–15, 1988.

[406] I. B. Türksen, D. Ulguray, and Q. Wang. A qualitative model for hierarchical job shop scheduling. In *NAFIPS'90: Quater Century of Fuzziness*, Toronto, Canada. June 1990.

[407] I. B. Türksen. Fuzzy logic-based expert systems for operations management. In C.Y. Suen and R. Shinghal, editors, *Operational Expert System Applications in Canada*, pages 170–183. Pergamon Press Ltd/Headington Hill Hall/Oxford OX3 0BW/United Kingdom, 1991.

[408] I. B. Türksen and Y. Tian. A fuzzy expert system for a spare-parts service centre. In *The World Congress on ES Proceedings*, pages 835–847, 1991.

[409] I. B. Türksen and M. Berg. An expert system prototype for inventory capacity planning: An approximate reasoning approach. *Int. J. Approximate Reasoning*, 5:223–250, 1991.

[410] I. B. Türksen, D. Ulguray, and Q. Wang. Hierarchical scheduling based on approximate reasoning — a comparison with ISIS. *Fuzzy Sets and Systems*, 46:349–371, 1992. ⇒ANNOTATION.

[411] I. B. Türksen. Fuzzy expert systems for IE/OR/MS. *Fuzzy Sets and Systems*, 51:1–27, 1992.

[412] K. Ueyama, K. Ezaki, S. Hirayama, and T. Niidome. Application of fuzzy model to preset of tandem cold mill. In *Preprints of the 2nd IFSA Congress (1987)*, pages 358–361, Tokyo. July 1987. ⇒ANNOTATION.

[413] Alasdair Urquhart. Many-valued logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume III (Alternatives to Classical Logic), pages 71–116. Kluwer Academic Publishers, 1986. ⇒ANNOTATION.

[414] Robert Valette, J. Cardoso, and Didier Dubois. Monitoring manufacturing systems by means of Petri nets with imprecise markings. In *IEEE International Symposium on Intelligent Control 1989, 25-26 Sept., Albany N.Y., USA*, pages 233–238, 1989. URL: ftp://laas.laas.fr/pub/robert/albany.tex.

[415] Robert Valette and M. Courvoisier. Petri nets and artificial intelligence. In *IEEE/SICE International Workshop an Emerging Technologies for Factory Automation, Palm Cove, Cairns, North Queensland, Australia, August 17-19 1992*, pages 218–238, 1992. URL: ftp://laas.laas.fr/pub/robert/rp_ia92.ps.

[416] J.L. Verdegay. Fuzzy mathematical programming. In M.M. Gupta and E. Sanchez, editors, *Fuzzy Information and Decision Processes*, pages 231–237. North-Holland, 1982.

[417] J.L. Verdegay. Problemas de transporte con parametros difusos. *Rev. Acad. Ciencias Mat. Fis. Quim y Nat. de Granada*, 2:47–56, 1983. In Spanish.

[418] J.L. Verdegay. Solving the mathematical programming problem with a new formulation of fuzzy objective. *BUSEFAL*, 15:127–133, 1983.

[419] J.L. Verdegay. Problemas de decision en ambiente difuso. *Trabajos de Estadistica y de Investigacion Operativa*, 34(3):68–78, 1983. In Spanish.

[420] J.L. Verdegay. Duality in fuzzy linear programming. *BUSEFAL*, 16:91–95, 1983.

[421] J.L. Verdegay. A dual approach to solve the fuzzy linear programming problem. *Fuzzy Sets and Systems*, 14:131–141, 1984.

[422] J.L. Verdegay. Applications of fuzzy optimization in operational research. *Control and Cybernetics*, 13(3):229–239, 1984.

[423] J.L. Verdegay. Fuzzy mathematical programming problem: Resolution. In M.G. Singh, editor, *Systems and Control Encyclopedia. Theory, Technology, Applications*, pages 1816–1819. Pergamon Press, 1987.

[424] J.L. Verdegay and M. Delgado, editors. *The Interface between Artificial Intelligence and Operations Research in Fuzzy Environment.* ISR Series no. 95. Verlag TÜV Rheinland, 1989.

[425] J.L. Verdegay and M. Delgado. *Approximate Reasoning Tools for Artificial Intelligence.* ISR Series no. 96. Verlag TÜV Rheinland, 1990.

[426] Ina Wagner. A web of fuzzy problems: Confronting the ethical issues. *Communications of the ACM*, pages 94–101, June 1993. ⇒ANNOTATION.

[427] P.Z. Wang. Latticized linear programming and fuzzy relation inequalities. *J. Math. Anal. Appl.*, 159(1):72–87, 1991.

[428] T. Watanabe. Job-shop scheduling using fuzzy logic in a computer integrated manufacturing environment. In *Proceedings of the 5th International Conference on System Research, Informatics and Cybernetics*, pages 150–158, Baden-Baden, Germany. August 1990.

[429] Tohru Watanabe, Hidekatsu Tokumare, Yusuke Nakajima, and Yasunori Hashimoto. Job-shop scheduling using fuzzy inference to take profit into account. In Ming Leu, editor, *Proceedings of the JAPAN/USA Symposium on Flexible Automation (A Pacific Rim Conference)*, volume 1, pages 423–427, San Francisco, California. July 1992.

[430] B. Werners. Interactive multiple objective programming subject to flexible constraints. *European Journal of Operational Research*, 31:324–349, 1987.

[431] Thomas Whalen and Brian Schott. Interactive task assignment with composite fuzzy preference relations: An application to faculty scheduling. In James C. Bezdek, editor, *Analysis of Fuzzy Information: Artificial Intelligence and Decision Systems*, volume II, pages 131–144. CRC Press, Inc., Boca Raton, FL (USA), 1987.

[432] G. Wiedey and H.J. Zimmermann. Media selection and fuzzy linear programming. *J. Op. Res. Soc.*, 29:1071–1084, 1978.

[433] M. R. Wilhelm and H. R. Parsaei. A fuzzy linguistic approach to implementing a strategy for computer integrated manufacturing. *Fuzzy Sets and Systems*, 42, 1991.

[434] Patrick Henry Winston. *Artificial Intelligence*. Addison-Wesley Publishing Company, Inc., 1977.

[435] L. Richard Woodyatt, Kenneth L. Stott, Floyd E. Wolf, and Francis J. Vasko. Using fuzzy sets to assign metallurgical grades to steel. *Journal of Metallurgy*, pages 28–31, February 1992. ⇒ANNOTATION.

[436] L.R. Woodyatt, K.L. Stott, F.E. Wolf, and F.J. Vasko. An application combinig set covering and fuzzy sets to optimally assign metallurgical grades to customer orders. *Fuzzy Sets and Systems*, 53:15–25, 1993.

[437] Ronald R. Yager. A new methodology for ordinal multiobjective decisions based on fuzzy sets. *Decision Sciences*, 12:589–600, 1981. Republished in Dubois et al. [119, pp. 751–756]. ⇒ANNOTATION.

[438] R. R. Yager, S. Ovchinnikov, R. M. Tong, and H. T. Nguyen, editors. *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh.* John Wiley & Sons, 1987.

[439] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, Cybernetics*, 18:183–190, 1988. Republished in Dubois et al. [119, pp. 80–87]. ⇒ANNOTATION.

[440] R.R. Yager. A mathematical programming approach to inference with the capability of implementing default rules. *Int. J. of Man-Machine Studies*, 29:685–714, 1988.

[441] T. Yamaguchi and Y. Kono. Application of fuzzy multiobjective linear programming to greenhouse cultivation planning. *Japanese Journal of Fuzzy Theory and Systems*, 4(6):701–708, 1992.

[442] H. Yamane, H. Tottori, and T. Yoshida. Development of BOF blowing expert system. *CAMP-ISIJ*, 2:218ff, 1989. In Japanese. ⇒ANNOTATION.

[443] Y. Yang. A new approach to uncertain parameter linear programming. *European Journal of Operational Research*, 54:95–114, 1991.

[444] T. Yang, J.P. Ignizio, and H-J. Kim. Fuzzy programming with nonlinear membership functions: Piecewise linear approximation. *Fuzzy Sets and Systems*, 41:39–53, 1991.

[445] H. Yano and M. Sakawa. Interactive fuzzy decision making for generalized multiobjective linear fractional programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 32:245–261, 1989.

[446] A.V. Yazenin. Fuzzy and stochastic programming. *Fuzzy Sets and Systems*, 22:171–180, 1987.

[447] Lotfi A. Zadeh. Fuzzy sets. *Information and Control, New York: Academic Press.*, 8:338–353, 1965. Republished in [438].

[448] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning (parts 1 and 2). *Information Sciences*, 8:199–249, 301–357, 1975. Republished in [438].

[449] L.A. Zadeh. Calculus of fuzzy restrictions. In L.A. Zadeh, K.S. Fu, K. Tanaka, and M. Shimura, editors, *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, pages 1–40. Academic Press, 1979.

[450] Lotfi A. Zadeh. Knowledge representation in fuzzy logic. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):89–100, March 1989.

[451] Lotfi A. Zadeh. Fuzzy sets and fuzzy logic: An overview. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 507–508. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[452] Maria Zemankova-Leech and Abraham Kandel. *Fuzzy relational data bases – a key to expert systems.* Verlag TÜV Rheinland, 1984.

[453] R. Zhao and R. Govind. The complete decision set of the generalized symmetrical fuzzy linear programming problem. *Fuzzy Sets and Systems*, 51:53–65, 1992.

[454] H.J. Zimmermann. Optimization in fuzzy environment. In *Proceedings of the XXI Int. TIMS and 46th ORSA Conference*, San Juan (Puerto Rico). 1974.

[455] H.J. Zimmermann. Description and optimization of fuzzy systems. *International Journal of General Systems*, 2:209–215, 1975.

[456] H.J. Zimmermann. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1(1):45–55, 1978.

[457] H.J. Zimmermann and M.A. Pollatschek. Fuzzy 0-1 linear programs. In H.J. Zimmermann, L.A. Zadeh, and B.R. Gaines, editors, *Fuzzy Sets and Decision Analysis*, pages 133–145. North-Holland, 1984.

[458] Hans-Jürgen Zimmermann. Applications of fuzzy set theory to mathematical programming. *Information Sciences*, 36:29–58, 1985. Republished in Dubois et al. [119, pp. 795–809].

[459] H.J. Zimmermann. Applications of fuzzy sets theory to mathematical programming. *Information Sciences*, 36:29–58, 1985.

[460] H.J. Zimmermann. *Fuzzy Sets, Decision Making and Expert Systems*. Kluwer Academic Publishers, 1987.

[461] Hans-Jürgen Zimmermann. *Fuzzy Set Theory — and Its Applications*. Kluwer Academic Publishers, 2nd, revised edition, 1991. ⇒ANNOTATION.

[462] Hans-Jürgen Zimmermann. Fuzzy mathematical programming. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 521–528. John Wiley & Sons, Inc., 2nd enlarged and revised edition, 1992. ⇒ANNOTATION.

[463] Hans-Jürgen Zimmermann. Approximate reasoning in manufacturing. In A. Kusiak, editor, *Intelligent Design and Manufacturing*, pages 701–722. John Wiley & Sons, Inc., 1992.

[464] Steven Zoraster and Ronald Sawey. Optimization-based vs. rule-based. *Communications of the ACM*, 35(6), June 1992.

[465] Monte Zweben, M. Deale, and R. Gargan. Anytime rescheduling. In *Proceedings of the DARPA Workshop on Innovative Approaches to Planning and Scheduling*, 1990.

[466] Monte Zweben, Eugene Davis, Brian Daun, Ellen Drascher, Micheal Deale, and Megan Eskey. Learning to improve constraint-based scheduling. *Artificial Intelligence*, 58:271–296, 1992.

Acknowledgements

I would like to thank Romana Baier, Markus Bonner, Friedrich Döll, Gerhilde Egghart, Ali Gharakani, Mario Girsch, Wolfgang Graff, Witold Hendrysiak, Volker Lainer, Thomas Länger, Stefan Mayer, Manfred Mitterholzer, Mehrdad Rohani-Amiri, Ulrich Santa, Günther Skele, Rainhard Steindl, Wolfgang Steindl, Chadi Suleiman, Gabi Thalhammer, and Eva Valsky for their excellent implementation work, their help in general, and many valuable discussions we had together.

I would like to thank the many people whose help and comments about my ideas and preliminary versions of this thesis in particular have been been so important, including: Harvey Abramson, Klaus-Peter Adlassnig, Shun-ichi Amari, Plamen Angelov, Yonnel Arrouas, Bernard De Baets, Ranan Banerji, Pauline Berry, Sandford Bessler, Hermann and Gerda Bodenseher, Rainer Born, Tim Boykett, Christer Carlsson, Brahma Deo, Didier Dubois, Thomas Eiter, Gerald Ehritz, Irina Ezhkova, José Ezquerra, Hélène Fargier, Dimitar Filev, Barry Flachsbart, Eugene Freuder, Gerhard Friedrich, Robert Fullér, Hans Gamper, Joachim Geidel, Christian Geiger, Bernard Grabot, Qi Guan, Volkmar Haase, Brigitte Haberstroh, Petr Hajek, Maciej Hapke, Alois Haselböck, Itsuo Hatono, Marcus Herzog, Masao Iri, Peter Kotauczek, Richard Kowalczyk, Rudolf Kruse, Johannes Kuntner, Franz Lackinger, Jérôme Lang, Roger Martin-Clouaire, Heidi Milos, Steve Minton, Bernhard Moser, Alois Niedermayr, Erika Nowak, Max Ott, Jan Overbeck, Helmut Pinger, Henri Prade, Patrick Prosser, Johannes and Monika Retti, Elie Sanchez, Josef Scheidl, Thomas Schiex, Werner Schimanovich, Norbert Schindler, Brian Schott, Andreas-Geyer Schulz, Katrin Seyr, Peter Skalicky, Emmerich Simoncsics, Kyōko Slany, Wolfgang Snopek, Christian Stary, Klaus Stohl, Markus Stumptner, Hideyuki Takagi, Bill Taylor, Herbert Toth, Robert Trappl, Burhan Türksen, Motohide Umano, José-Luis Verdegay, Roman Weissgärber, Helmar Weseslindtner, Lotfi Zadeh, Hans-Jürgen Zimmermann, and Monte Zweben.

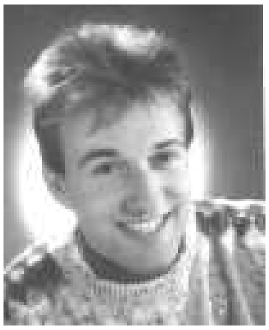
I especially wish to thank Jürgen Dorn, Roger Kerr, and Peter Klement for their assistance and advice. Working together was always a learning experience for myself, as well as a constant source of motivation. I hope that the contents of this dissertation will meet the quality of our scientific cooperation. Nevertheless, all misinterpretations and mistakes are mine.

Finally, I whish to express my special thanks to Georg Gottlob. He gave sustained support and motivation, coupled with the right degree of freedom, while always keeping me focused on what's really important.

This research was facilitated by the generosity and courage of the Austrian Industries Holding that provides founding for the basic research going on at the Christian Doppler Laboratory for Expert Systems. I always enjoyed the rich and scientifically challenging environment at the laboratory.

Vienna, Austria — June, 1994 W. S.

Curriculum Vitae



Personal Data:

Name:	Dipl.-Ing. Wolfgang SLANY.
Born:	November 14, 1966, in Vienna, Austria.
Parents:	Prof. Dr. Jörg and Dr. Edda SLANY.
Sisters:	Astrid and Nicole SLANY.
Nationality:	Austria.
Marital status:	Married to Kyōko SLANY (born JINSEI) since February 23, 1990 (Tokyo, Japan), March 2, 1990 (Petaling Jaya, Malaysia), March 25, 1990 (Toba, Japan), August 12, 1990 (Maria Ellend, Austria), and April 7, 1994 (Schönbrunn, Austria).
Languages:	German, French, English, Japanese.
Hobbies:	Japanese; Future Telecommunication; Reading; Learning; Neurophysiology; Quantum-Cogno-Dynamics; Altruism; Time Management; Travelling; Squash, Unicycle, Boomerang, Skiing, Juggling, Tennis, Dancing, Tae-Kwon-Do, Rock'n Roll Acrobatics; Electronics; Organizing parties ...
Address:	Mariannengasse 21/5, A-1090 Wien, Austria.
Email:	wsi@vexpert.dbai.tuwien.ac.at
URLs:	http://www.dbai.tuwien.ac.at:8080/staff/slany.html ftp://mira.dbai.tuwien.ac.at/pub/slany

Education:

1973–1985:	Lycée Français de Vienne, Austria, Bac-C; passed with distinction.
1983–1984:	build my own computer with 256 Bytes RAM (sic!) and thermo-printer, which could do such things as adaptive MORSE-code recognition.
1985–1989:	Student of Mechanical Engineering and Computer Science at the Technical University of Vienna.

Summer 1989: Graduation as a Diplomingenieur of Information/Computer Science (Master's Thesis on Algebraic Optimization of Database Queries, see [XVI][2]); passed with distinction more than one year ahead of officially scheduled five years.

1989–1991: Research Student of Mathematical Engineering and Information Physics at the University of Tokyo under Prof. M. Iri, Japan, with a scholarship from the Mombushō, the Japanese Ministry of Education.

since Fall 1989: Ph.D. student in Information/Computer Science at the Technical University of Vienna under Prof. G. Gottlob, student of Mechanical Engineering at the Technical University of Vienna.

1985–1994: various seminars on Fuzzy Logic, Artificial Intelligence, Management Methodics, Trade and Economics, Personality Training, and Foreign Languages.

Work Experience:

1982–1984: Freelance computer-game programmer.

Summer 1984: Probationer (computer program development section) at the Computer Center of the Municipal Electric Power Stations of Vienna.

Summer 1987: Probationer (computer program development section) at Hewlett Packard Austria [IX].

1987–1988: Tutor (teaching programming in Modula-2) at the Department of Practical Information Science under Prof. M. Brockhaus, Technical University of Vienna.

1987–1989: System manager for the Key Station (Computer Assisted Japanese Liasion Office) under Prof. E. Simoncsics, Technical University of Vienna.

1988–1989: Student-assistant (research, implementation) at the Department of Applied Computer Science under Prof. G. Gottlob, Technical University of Vienna, working on the ARTHUR project [XVI].

1988–1989: Half-time programmer at Profisoft (now Vienna Software Publishers), Austria, to implement the algebraic simplifier for the OS/2 toolset N/joy! [XIII].

Summer 1989: Probationer (computer program development section; AI section; Sales section) at Kōzō Keikaku, Tokyo, Japan [XIV], [XV].

1990: Freelance Japanese–English translator.

1990–1991: Freelance far-east collaborator of the Christian Doppler Laboratory for Expert Systems c/o Department of Information

[2] Please refer to the last Section of this Curriculum Vitae for the [roman-number] references.

Systems, Technical University of Vienna, which resulted in a study about Artificial Intelligence Trends in Japan 1991, see [XX].

1991–1993: Research and teaching assistant at the Christian Doppler Laboratory for Expert Systems c/o Department of Information Systems under Prof. G. Gottlob, Technical University of Vienna.

1993–1994: Research and teaching assistant at the Department of Information Systems under Prof. G. Gottlob, Technical University of Vienna.

since April 1994: University assistant at the Department of Information Systems under Prof. G. Gottlob, Technical University of Vienna.

Scientific Activities and Teaching Experience:

1986–now: Various publications, for details see next Section. The references given in this Section can also be found there. In the following, courses and talks I gave at various occasions, plus other work related to scientific activities are listed.

1987–1988: Teaching programming in Modula-2 to younger students as a tutor at the Department of Practical Information Science, Technical University of Vienna.

1987–1989: ÖGAI (Austrian Society for Artificial Intelligence) membership coordinator.

1987: Talk in German language about ‘Connectionism’ at the Secessionsgespräche [VII].

1987: Talk in German language about ‘Clinical observations about the asymmetry of the brain: a historical review’ at the Department for Logistics of the University of Vienna [V].

1987: Talk in German language about ‘The Origin of Consciousness in the Breakdown of the Bicameral Mind’ at the Secessionsgespräche [VIII].

1988: Talk in German language about ‘Competitive Learning’ at the Department for Medical Cybernetics and Artificial Intelligence of the University of Vienna [X].

1988: Talk in German language about a ‘Pattern Oriented Functional Programming Language’ at the Department of Practical Information Science of the Technical University of Vienna [XII].

1990: Talk in Japanese language about ‘Telecommunication in the Future’ at the Tokyo University [XVIII].

1990: Won the Examiner’s Prize to the 1990 Tokyo International Student Communication Essay Contest organized by the Daily Yomiuri (largest Japanese newspaper) with a paper in Japanese language about ‘Telecommunication in the Future Metropolis’ [XVII].

1991: Talks in German language about ‘Artificial Intelligence Trends in Japan’ in Linz (Austrian Industries) and at the Austrian Research Institute for Artificial Intelligence, University of Vienna [XX].

1991–1992: Various interviews in German language for national radio stations and newspapers regarding ‘Fuzzy Logic in Japan’ (some of it can be found in [XIX], [XXV]).

1991–1992: Organization of a students seminar about ‘Knowledge Based Scheduling’ [XXIX].

1992: Talk in German language about ‘Fuzzy Control applied to Microbiology’ at the Department for Applied Microbiology, Universität für Bodenkultur, Vienna.

1992: Talk about ‘Uncertainty Management by Relaxation of Conflicting Constraints in Production Process Scheduling’ at Stanford University, Ca., at the AAAI Spring Symposium about ‘Practical Approaches to Scheduling and Planning’ [XXIV].

1992: Talk about ‘Uncertainty Management in Production Process Scheduling applied to High-Grade Steelmaking’ at FORWISS, München, at the 6. Planungs und Konfigurierungs Workshop ‘Planen unter Unsicherheit’ [XXX].

1992: Talk in German language about ‘Neural Nets and Fuzzy Logic’ at the University of Linz.

1992: Talks about ‘Vague Data Management in Production Process Scheduling applied to High-Grade Steelmaking’ at the University of Maryland at the First International Conference on Artificial Intelligence Plannning Systems and at the Department of Information and Software Systems Engineering, George Mason University, Va. [XXXI].

1992: Organized an international mailing-list for ‘Fuzzy Logic’ (fuzzy-mail@vexpert.dbai.tuwien.ac.at).

1992–1993: Chairperson together with Peter Klement from the University of Linz, of the 8. Austrian Artificial Intelligence Conference about ‘Fuzzy Logic in Artificial Intelligence’ [XXXVIII].

1992–1993: Organization of a students seminar about ‘Scheduling of Production Processes - From linear Integer-Models to symbolic AI-Models’ together with Jürgen Dorn [XXXIV].

1993: Talk in German language about ‘Fuzzy Scheduling’ at the Department for Medical Computer Sciences at the University of Vienna.

1993: Talk about ‘Fuzzy Constraint Relaxation Techniques for Knowledge-Based Scheduling’ in Linz at the Fuzzy Scheduling Workshop organized by Roger Kerr at the 8. Austrian Artificial Intelligence Conference about ‘Fuzzy Logic in Artificial Intelligence’ [XL].

1993: Talk about 'Fuzzy-Mail-List and related services' in Linz at the Workshop for Doctoral Students in Fuzzy-Based Systems, organized by Hans Gamper and Bernhard Moser at the 8. Austrian Artificial Intelligence Conference about 'Fuzzy Logic in Artificial Intelligence'.

1993: Talk about 'Fuzzy Expert System for Maintenance Interval Prediction' in Graz, Austria, at the European Symposium on Computer Aided Process Engineering-3, ESCAPE-3, 25th European Symposium of the Working Party on Computer Aided Process Engineering, 494th Event of the European Federation of Chemical Engineering (EFChE) [XLI].

1993: Organized an international mailing-list for 'Knowledge Based Scheduling' together with Sandford Bessler (sched-l@vexpert.dbai.tuwien.ac.at).

1993: Talk about 'Fuzzy Constraint Relaxation Techniques for Knowledge-Based Scheduling' in Aachen, Germany, Session on Fuzzy Constraint Propopagation chaired by Henri Prade, EUFIT'93, First European Congress on Fuzzy and Intelligent Technologies [XXXIX].

1993: Talk about 'Fuzzy Expert System to Predict Maintenance Intervals in a Continuous Caster' in Seoul, RO Korea, at the Computerized Production Control in Steel Plant Conference, CPC-93 [XLII].

1993: Talk in German language about '3 Repair-Algorithms for the Fine-Planning in the LD3-Works in Linz' in Linz (VOEST Alpine Industrial Engineering Plant).

1994: Representative of the Department of Information Systems, Technical University of Vienna as an institutional affiliate of the Berkeley Initiative for Soft Computing (BISC) headed by Prof. L. Zadeh.

1994: Organized an national mailing-list for 'Austrian Computer Professionals for Social Responsibility' together with Peter Purgathofer (eCE@vexpert.dbai.tuwien.ac.at).

1994: Invitation to and talk at the CIFT'94 workshop on Current Issues in Fuzzy Technologies: Decision Models and Systems organized by Prof. M. Fedrizzi [LI].

1991–now: supervised the thesis and practica of so far 30 students at the Department of Information Systems, Technical University of Vienna.

1991–now: visited research centers in Austria, Czech Republic, France, Germany, Italy, Japan, and the United States of America.

1985–now: participated at numerous national and international conferences and workshops; helped organizing ECAI'92 and EDBT'92, organizing committee member at FUBEST'94. Helped organize lectures by Prof. Raymond Reiter, Prof. Yves Kodratoff, Prof. Harvey Abramson, Prof. Lotfi Zadeh, Prof. Rudolf Kruse, Prof. Hans-Jürgen Zimmermann, Prof. Brahma Deo, and Prof. Ranan Banerji.

1989–now: served as reviewer and program committee member for several conferences and journals related to artificial intelligence (ECAI, IJCAI, IEEE NN, ...) and fuzzy logic (FLAI, FUBEST).

Scientific Affiliations: Member of AAAI (American Association for Artificial Intelligence), ACM (Association of Computing Machinery), eCE (Austrian Computer Professionals for Social Responsibility), IEEE Computer (Institute of Electrical and Eletronics Engineers, Computer Society), OCG (Austrian Computer Society), ÖGAI (Austrian Society for Artificial Intelligence), and NIPPON-ÖJG (Austro-Japanese Society).

List of Publications:

The following list contains most of the author's publications since 1986, some of them very old, some in German, some in Japanese, but most in English. The order is by year, if equal according to the authors, therefore items related to the present thesis should be searched at the end of the list. The Bibliography Section of this thesis starts on page 130, the corresponding selected annotations start on page 105.

[I] Anton Ertl, Martin Laubach, Wolfgang Slany, and Stefan Thurow. Natürlichsprachige Ausgabe. Arbeitsgemeinschaft Vergleich von Programmiersprachen PSAG T12-14, TU Wien, 1986. In German.

[II] Wolfgang Slany. Buchbesprechung: Into the heart of the mind. *ÖGAI Journal*, 5(4):44–45, 1986. In German.

[III] Yonnel Arrouas, Martin Laubach, Ramesh Misra, and Wolfgang Slany. Auswirkungen des Robotereinsatzes in Österreich. Gesellschaftswissenschaftliche Grundlagen der Informatik AG 3, Institut für Praktische Informatik, TU Wien, 1987. In German.

[IV] Johannes Blach, Anton Ertl, Martin Laubach, and Wolfgang Slany. Some thoughts about structured system design. Softwaretechnology Report 4, Institut für Praktische Informatik, TU Wien, 1987. In German.

[V] Anton Ertl and Wolfgang Slany. Klinische Beobachtungen zur Asymmetrie des Gehirns: Ein geschichtlicher Überblick. Seminar aus Neuroinformatik, Institut für Logistik der Uni Wien, 1987. In German.

[VI] Thomas Hadek, Manfred Lipp, Alexander Ruzicka, Markus Ruzicka, and Wolfgang Slany. Bericht über ein Programmierabenteuer mit dem

LIMAT RT280. LU Report 4, Institut für Industrieroboter und Handhabungsgeräte, TU Wien, 1987. In German.

[VII] Wolfgang Slany. Connectionism. In German. Manuskript zu meinem Vortrag im Rahmen der Secessionsgespräche am 29. April, 1987.

[VIII] Wolfgang Slany. The origin of consciousness in the breakdown of the bicameral mind. In German. Manuskript zu meinem Vortrag im Rahmen der Secessionsgespräche, 1987.

[IX] Wolfgang Slany and Matthias Mitterauer. HP-Software Support Service Configurator. Ergebnisbericht zu Ferialpraxisprojekt, Hewlett Packard Austria, 1987. In German.

[X] Wolfgang Slany. Competitive learning. Institutsbericht 1988-3, Institut für Medizinische Kybernetik und Artificial Intelligence der Uni Wien, 1988. In German.

[XI] Wolfgang Slany. HEXI. Ergebnisbericht N. 31, Institut für Statistik und Informatik der Uni Wien, January 1988. In German.

[XII] Wolfgang Slany. Pattern oriented functional programming language. In Robert Glück, editor, *Sprachen zur funktionalen Programmierung.* Institut für Praktische Informatik der Technischen Universität Wien, 1988. In German.

[XIII] Wolfgang Slany. Querystrukturen. Informatik Praktikum II, Institut für Angewandte Informatik und Systemanalyse der TU Wien, 1988. In German.

[XIV] Wolfgang Slany. Concurrency control in distributed database systems. Report, Kōzō Keikaku, Tokyo, Japan, 1989.

[XV] Wolfgang Slany. The itty-bitty information center about postage in Japan. Report, Kōzō Keikaku, Tokyo, Japan, 1989. Partly in Japanese.

[XVI] Wolfgang Slany. Optimierung relationaler Anfragen am Beispiel der ARTHUR Implementierung. Diplomarbeit ($\simeq$ master's thesis), Technical University of Vienna, 1989. In German.

[XVII] Wolfgang Slany. Telepolis. In Japanese. Examiner's Prize Entry to the 1990 Tokyo International Student Communication Essay Contest organized by the Daily Yomiuri (largest Japanese newspaper), 1990.

[XVIII] Wolfgang Slany and Kyōko Slany. Mirai no komyunikeeshon. In Japanese. Unpublished essay about the future ways to communicate, 1990.

[XIX] Wolfgang Slany. 2000 Patente für Fuzzy-Logik. *a3-Volt*, 13(12), Dezember 1991. In German.

[XX] Wolfgang Slany. Artificial Intelligence Trends in Japan 1991. CD-Studie 91/5, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, September 1991. In German.

[XXI] Wolfgang Slany. Dairokkan. In Japanese. Entry to the 1991 Japanese Design the Future Essay Contest, 1991.

[XXII] Jürgen Dorn, Wolfgang Slany, and Christian Stary. Uncertainty management by relaxation of conflicting constraints in production scheduling. In Marc Drummond, Mark Fox, Austin Tate, and Monte Zweben, editors, *Practical Approaches to Scheduling and Planning*, Working Notes AAAI Spring Symposium Series, pages 62–66, Stanford, CA, March 1992. Published by the American Association of Artificial Intelligence.

[XXIII] Jürgen Dorn, Wolfgang Slany, and Christian Stary. Uncertainty management by relaxation of conflicting constraints in production process scheduling. CD-Technical Report 92/33, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, January 1992. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/cd-tr9233.ps.Z.

[XXIV] Jürgen Dorn, Wolfgang Slany, and Christian Stary. Uncertainty management by relaxation of conflicting constraints in production scheduling. In Marc Drummond, Mark Fox, Austin Tate, and Monte Zweben, editors, *Practical Approaches to Scheduling and Planning*, Working Notes from the 1992 AAAI Spring Symposium Series, republished as NASA Technical Report FIA-92-17, pages 62–66, Stanford, CA, May 1992. NASA Ames Research Center, Artificial Intelligence Branch.

[XXV] Wolfgang Slany. Anstoß gab die Schrift. *a3-Volt*, 14(1-2), Jänner/Februar 1992. In German.

[XXVI] Wolfgang Slany. Conference report: Workshop on the design and analysis of fuzzy controllers, workshop on fuzzy logic R&D in Germany. *AI Communications: the European Journal on Artificial Intelligence*, 5(2):92–95, June 1992.

[XXVII] Wolfgang Slany. Verantstaltungsbericht: Realize Concrete Applications of Fuzzy Logic in Industry. *ÖGAI Journal*, 11(1):7, June 1992.

[XXVIII] Wolfgang Slany. Verantstaltungsbericht: Workshop on the Design and Analysis of Fuzzy Controllers, Workshop on Fuzzy Logic R&D in Germany. *ÖGAI Journal*, 11(1):3–6, June 1992.

[XXIX] Wolfgang Slany, editor. Knowledge based scheduling. CD-Studie 92/6, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, January 1992.

[XXX] Wolfgang Slany, Christian Stary, and Jürgen Dorn. Uncertainty management in production scheduling applied to high-grade steelmaking. In *6. Workshop Planen und Konfigurieren*, FORWISS-Report, pages 51–60, Erlangen München Passau, March 1992. Bayrisches Forschungszentrum für Wissensbasierte Systeme.

[XXXI] Wolfgang Slany, Christian Stary, and Jürgen Dorn. Vague data management in production process scheduling applied to high-grade steelmaking. In *Proceedings of the First International Conference on*

Artificial Intelligence Planning Systems, pages 214–221, University of Maryland, June 1992. Morgan Kaufmann Publishers, Inc.

[XXXII] Jürgen Dorn, Mario Girsch, and Wolfgang Slany. Reparatur von Plänen durch fallbasiertes Schließen. In A. Bockmayr and F. J. Radermacher, editors, *Forschungsbericht des Max-Planck-Instituts für Informatik zum Workshop Künstliche Intelligenz und Operations Research*, September 1993. In German.

[XXXIII] Jürgen Dorn and Wolfgang Slany. A flow shop with compatibility constraints in a steelmaking plant. CD-Technical Report 93/56, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993.

[XXXIV] Jürgen Dorn and Wolfgang Slany, editors. Scheduling von Produktionsprozessen - Von linearen Integermodellen zu symbolischen AI-Modellen. CD-Studie 93/9, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, July 1993. In German.

[XXXV] Jürgen Dorn, Wolfgang Slany, Wolfgang Snopek, Christian Stary, Wolfgang Steindl, and Klaus Stohl. Aufgabenananlyse der Dispatcher im Stahlwerk LD3. CD-Studie 93/11, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993. In German.

[XXXVI] Jürgen Dorn, Wolfgang Slany, Christian Stary, Wolfgang Steindl, Wolfgang Snopek, and Klaus Stohl. Aufgabenbasierte Spezifikation des Bildschirmarbeitsplatzes für Dispatcher im Stahlwerk LD3. CD-Studie 93/12, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1993. In German.

[XXXVII] Marcus Herzog, Riccardo Peratello, Christian Kühn, and Wolfgang Slany. Exploring architectural design cases. In Richard Furuta et al., editor, *Workshop Notes of the CIKM '93 Workshop on Intelligent Hypertext, held in conjunction with the ACM Conference on Information and Knowledge Management*, Washington, D.C., November 1993. University of Maryland Baltimore County.

[XXXVIII] Erich Peter Klement and Wolfgang Slany, editors. *Fuzzy Logic in Artificial Intelligence. Proceedings of the 8th Austrian Artificial Intelligence Conference, FLAI'93, Linz, Austria, June 1993*, volume 695 of *Lecture Notes in Artificial Intelligence*. Springer Verlag Berlin Heidelberg, 1993. A conference report by Thalhammer [399] is available on the net: URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/flai.txt.

[XXXIX] Wolfgang Slany. Fuzzy constraint relaxation techniques for knowledge-based scheduling. In Hans-Jürgen Zimmermann, editor, *EUFIT'93, First European Congress on Fuzzy and Intelligent Technologies*, pages 1124–1127, Aachen, Germany, September 1993. Augustinus Buchhandlung. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/eufit93.ps.Z.

[XL] Wolfgang Slany. Fuzzy constraint relaxation techniques for knowledge-based scheduling. In Roger Kerr, editor, *Workshop Notes of the Workshop on Fuzzy Scheduling Systems*, Linz, Austria, June 1993. University of Linz, Department of Mathematics.

[XLI] Wolfgang Slany. Fuzzy expert system for maintenance interval prediction. In Franz Moser, Hans Schnitzer, and Hansjörg Bart, editors, *European Symposium on Computer Aided Process Engineering-3. ESCAPE-3, 25th European Symposium of the Working Party on Computer Aided Process Engineering, 494th Event of the European Federation of Chemical Engineering (EFChE)*, Supplement to Computers & Chemical Engineering, pages S155–S159, Graz, Austria, July 1993. Pergamon Press Ltd.

[XLII] Wolfgang Slany. Fuzzy expert system to predict maintenance intervals in a continuous caster. In Duk-Hyon Baik, editor, *Preprints of the international conference CPC-93, Computerized Production Control in Steel Plants*, pages 291–296, Seoul, Korea, November 1993. The Korean Institute of Metals and Materials, The Institute of Materials, UK. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/fespmicc.ps.Z.

[XLIII] Wolfgang Slany and Christian Stary. The art of the belly. CD-Technical Report 93/49, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, March 1993.

[XLIV] Wolfgang Slany and Christian Stary. The art of the belly. In Gavriel Salvendy and Michael J. Smith, editors, *Proceedings of the HCI International '93 Conference*, volume 2, pages 705–710. Elsevier Science Publishers B.V., August 1993.

[XLV] Jürgen Dorn, Mario Girsch, Günther Skele, and Wolfgang Slany. Comparison of iterative improvement techniques for schedule optimization. CD-Technical Report 94/61, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994.

[XLVI] Jürgen Dorn and Wolfgang Slany. A flow shop with compatibility constraints in a steelmaking plant. In Mark Fox and Monte Zweben, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.

[XLVII] Roger M. Kerr and Wolfgang Slany. Research issues and challenges in fuzzy scheduling. CD-Technical Report 94/68, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/riacifs.ps.Z.

[XLVIII] Erich P. Klement and Wolfgang Slany. Fuzzy logic in artificial intelligence. CD-Technical Report 94/67, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994.

[XLIX] Wolfgang Slany. Fuzzy scheduling. CD-Technical Report 94/66, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/cd-tr9466.ps.Z.

[L] Wolfgang Slany. Scheduling as a fuzzy multiple criteria optimization problem. CD-Technical Report 94/62, Christian Doppler Laboratory for Expert Systems, Technical University of Vienna, 1994. URL: ftp://mira.dbai.tuwien.ac.at/pub/slany/cd-tr9462.ps.Z.

[LI] Wolfgang Slany. Scheduling with fuzzy decision making methods. In Mario Fedrizzi, Erich P. Klement, Aldo Ventre, and Alessandro Zorat, editors, *Proceedings of CIFT'94: Current Issues in Fuzzy Technologies: Decision Models and Systems*, Trento, Italy, June 1994.

[LII] Eva Valsky, Marcus Herzog, Riccardo Peratello, and Wolfgang Slany. The Department Information System of the Information Systems Department at the Technical University of Vienna. In *Proceedings of the Eurographics Symposium and Workshop on Multimedia: Multimedia/Hypermedia in Open Distributed Environments*, June 1994.

The preceding list contains only the author's own publications from 1986 as part of his Curriculum Vitae. The Bibliography of this thesis can be found starting on page 130.