A Flow Shop with Compatibility Constraints in a Steelmaking Plant*

Jürgen Dorn and Wolfgang Slany

Christian Doppler Laboratory for Expert Systems
Technical University of Vienna
E184/2, Paniglgasse 16, A-1040 Wien, Austria, Europe
Fax: +43-1-5055304, Email: {dorn|wsi}@vexpert.dbai.tuwien.ac.at

Abstract

We present a scheduling methodology for applications 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 single jobs and and the difficulty to schedule them is defined on the different constraints and is used to control the generation of schedules. A preliminary schedule is generated by considering the important jobs and those that are difficult to schedule first. Easy or not so important jobs are scheduled later. Finally, the achieved schedule is "repaired" until a schedule is found that achieves a given level of satisfaction. Since the goodness of solutions is rated by fuzzy sets, robust schedules achieve better evaluations than weak schedules. However, if no robust solution is found constraints will be relaxed. 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.

1 Introduction

Scheduling of factory processes is the allocation of required resources to jobs to be performed. 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*. 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

---

* *appeared in Intelligent Scheduling, eds Monte Zweben and Mark S. Fox, Morgan Kaufmann, pp. 629–654.*

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 consider both the explicit ordering constraints imposed by the plans and the implicit ordering constraints derived from the availabilities of the resources. The scheduler also must consider release dates, expected due dates, set-up times, and maintenance intervals.

Scheduling or *resource allocation* is a problem that is examined in the Operations Research (OR) literature since the early fifties (Conway *et al.* 1967). Methods and theories developed are based on strong mathematical theories. Although the theoretical work brought several improvements over the years, it was never very successful in realistic applications. There are mainly three problems with these formal-analytical models from OR:

- the algorithms are too complex for real world applications
- the models demand exact knowledge about durations and technical constraints
- the effort to formalize a new scheduling problem is considerable

Knowledge-based techniques promise a remedy, because they offer solutions for the cited problems. First, in knowledge-based systems it is easier to apply heuristics to reduce the inherent complexity. Second, knowledge-based systems offer the possibility of reasoning with incomplete, uncertain, vague, or probabilistic knowledge (Kerr & Walker 1989), (Berry 1992). Finally, in knowledge-based systems the existing knowledge is described explicitly and therefore it is easier to develop and maintain. The scheduling problem from the steelmaking industry described here, can be used as a prototypical case to illustrate the advantages of knowledge-based techniques.

In this application, approximately 45 jobs have to be sequenced for one production line. In a mathematical model without using heuristics the scheduler must check 45! possible sequences of jobs for constraint violations. It will be even more complex if conflicts between production lines are examined, or even more complex considering arbitrary idle times between operations. The described methodology manages this complexity by applying heuristics that the human experts used too.

An important problem in the domain is the uncertainty and vagueness. The durations for operations are only known approximately, and constraints are often specified vaguely. Most mathematical approaches from OR could not address this problem. We solve this problem by applying qualitative reasoning based on fuzzy logic.

OR 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 a knowledge-based model an additional rule or constraint is added very easily.

2 The Problem

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 for steel heats for one week (Dorn & Shams 1991). The system was implemented with PAMELA, a rule-based system developed by the *Alcatel Austria-ELIN Research Center* (Barachini & Theuretzbacher 1988). Although this system works well, we generalize the problem and set the applied method on a firmer ground by making it robust with respect to the influence of uncertainty and vagueness.

2.1 Description of the Environment

The Böhler company is divided into several plants. The steelmaking plant is the first in the production process. The produced steel is delivered to plants including the rolling mill or the forges. The steelmaking plant receives orders from these plants to produce slabs or ingots of a certain 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 the quality, some orders have due 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. For each production line of the steelmaking plant a list of jobs for one week is worked out manually. Usually the first shift in the steelmaking plant starts Sunday evening and the last shift ends Saturday. Sometimes fixed sequences for two or three jobs are given to facilitate the scheduling in the next plant. The task of the scheduler is to find a possible sequence for all jobs with 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 explain what combinations of orders may be produced in one week are given to the subsequent plants. Nevertheless, these constraining rules may be violated to produce important orders.

2.2 The Steelmaking Process

Pig iron produced in *blast furnaces* has usually more than 4 % carbon and is therefore brittle. To get a deformable product, *steel* is produced by reducing carbon in pig iron down to 2 %. For many high-tech products the quality of steel must be even higher. *High-grade steel* is crude steel refined with *alloying metals* like manganese, tungsten, chromium, or others. These alloying metals have desired effects 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 elements 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 quality. These operations and sequences must be only replanned 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 17 t to 55 t. The duration of the melting process depends on the ingredients but also on external causes. Since the furnaces consume much electric energy, they are sometimes switched off due to voltage peaks. Up to five hours can be required for the melting, but usually two to two and a half hour are enough. A fixed set-up and a maintenance interval of twenty minutes are 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 and whose duration is usually about the same as the melting. 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 the processing of the steel in a two-stranded *horizontal continuous caster* (HCC) to form *slabs* or the casting of it into *moulds* to form *ingots*. The teeming rate for the HCC is abouz 50 t/

h.. If the casting format must be altered, a set-up time must be considered, too. For casting ingots, space in one of the four *teeming bays* (TB), where ingots can solidify in the moulds, is required. Normally the solidification time for ingots in hours is half as much as the weight of the ingots in tons.

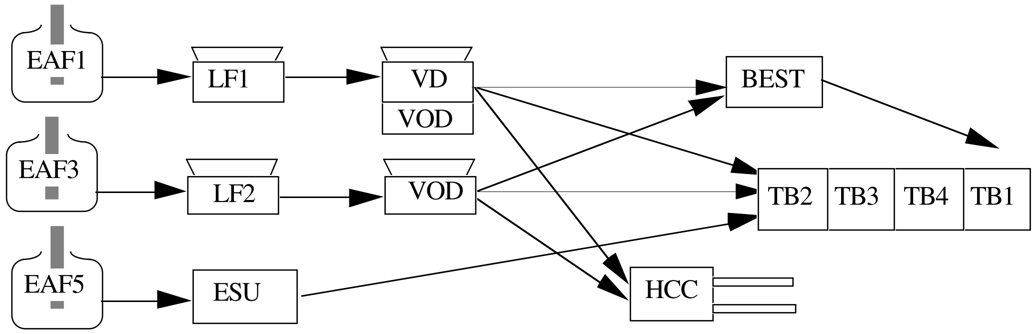

Figure 1: Aggregates in the Steelmaking Plant

For example, a big ingot of 52 t needs about one day. The storage places for big *forging grade ingots* (> 19 t) are limited. On the other hand, the effort for casting 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 and for them only one place in the teeming bays (TB 1) exists. All aggregates and the routings for heats are shown in figure 1.

2.3 Constraints in Scheduling

During the construction of a schedule several constraints have to be considered. These constraints are often vague, and they conflict with each other. The engineer has no pretension to generate an optimal schedule because he knows that the uncertainty in the execution of his 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:

2.3.1 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 in some cases the expert relaxes this factor.

Assume a heat h_1 contains 20 % nickel and a heat h_2 that is processed afterward in the furnace should have 4 % nickel. The second heat will take approximately

$$\frac{20\,\% \;-\; 4\,\%}{100\,\%} \cdot 3\ \% = .48\ \%\ \textit{nickel from the wall.}$$

When scrap iron is inserted in the furnace this amount is considered and only 3.52 % of nickel are taken. If the second heat may contain only up to .25 % of nickel, then this sequence of heats is not allowed.

The rule is effective for 42 chemical elements, but usually only 8 main elements are considered. Since not all elements react uniformly, exceptions exist that must be handled separately. 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 (inclusive ladles), usually only the electric arc furnace constraints are observed.

2.3.2 Temporal Constraints

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

For some jobs no appointments are made, although their subsequent treatment should be done immediately after casting. These jobs should not be scheduled at the end of the week because usually the subsequent plants are not working then. Some jobs with difficult treatments should be performed during day shifts, because an engineer should supervise these jobs. Usually the treatments in the aggregates behind the electric arc furnace are 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 have 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 function. The main objective is always to have as few destroyed heats as possible. Since this objective cannot be determined exactly, the secondary goal function is regarded seldom.

2.3.3 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 that are 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 (52 t) takes about one day. Since only one slot exists for such big ingots, only one per day can be produced. 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 be cast both 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 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.

2.4 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 and 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.

2.4.1 Alloying Cycles

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. The following figure illustrates the overlapping alloying cycles. The figure visualizes the amount of nickel and chrome for the jobs of schedule 2-1 that was generated from the orders of table 2.

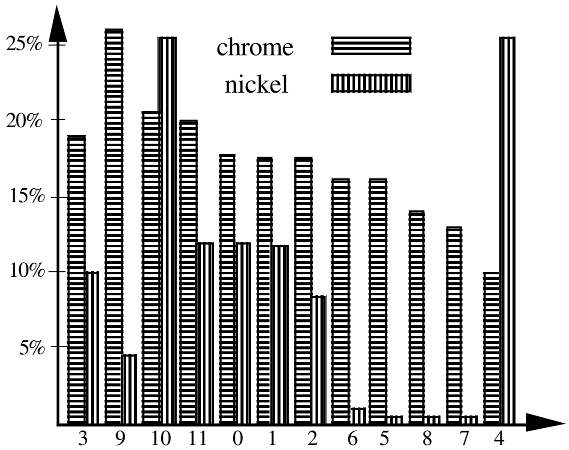

Figure 2: Overlapping Alloying Cycles

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.

2.4.2 Scheduling BEST-Ingots

BEST-ingots introduce some problems. Typically there are groups of such jobs with all of them having the same chemical quality requirements. They are usually forging grade ingots, should be delivered hot to the forge, and are low-alloyed, which means their amount of alloying metals is very low. From the compatibility view point 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 intervening 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.

2.4.3 Conflicts on the Continuous Caster

The HCC-unit is most economical if several heats with the same quality and format are cast continuously without breaks. In case that 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 set-up 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 very uncertain, the engineers prefer to charge the caster only with heats from one furnace. However, they sometimes change furnaces once per week. In this case, they decide when this change should take place before starting to schedule single jobs. Further, they accept only one format conversion. The expert schedules this conversion also a priori. Later, each job that requires the HCC-unit can be assigned to one part of the week.

3 The Methodology

Our approach to solve the problems mentioned in the introduction is as follows: for a given planning horizon a first 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, that means that the importance of one job may grow over time and is also dependent on the other jobs to be scheduled.

The 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 as defined in (Negoita 1985) and (Zadeh 1989) to model this evaluation. A fuzzy set is a set for which the boundary is non-sharp, with a gradual change from membership to non-membership. Fuzzy set theory is a more general form of normal set theory and is associated with fuzzy logic, a class of many-valued logic, in which statements may have fuzzy-truth-values rather than being limited to the values *true* or *false*. Several formalisms have been proposed for the propagation of vague knowledge.

Motivation for this choice was the ease of formulating knowledge that reflects the complex non-linear behavior expressed by the human experts. 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. In the third subsection, we explain the generation of a preliminary schedule. The system generates the schedules iteratively, important jobs are scheduled first, then gaps in the schedule are filled and finally other jobs are scheduled. The schedule generated in this phase may violate constraints and additionally some jobs may exist that were not scheduled due to conflicting constraints.

Therefore, the last phase is a repair phase with a search for a proper schedule. This approach is similar to those in (Minton *et al.* 1990) and (Zweben *et al.* 1990). They have shown that the repair-based methods work well on large, real-world scheduling problems and are useful for optimization problems and over-constrained problems. Such schedules can only be found if constraints are relaxed, because many constraints are antagonistic. This relaxation will be based on fuzzy sets.

3.1 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.

The name of each job identifies the quality of the steel. The column "time" gives the delivery date or the preferred time. The column "type" is used to give 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.

No	Name	Time	Type	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 1: List for Furnace EAF1

No	Name	Time	Type	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 2: List for Furnace EAF3

3.2 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.

3.2.1 Temporal Constraints

Temporal fuzzy values can be used to describe the duration of operations and that jobs are too early or too late. The fuzzy value describes a degree of uncertainty in both directions. The following values 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:

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

Since (Prade 79), and as recently as in (Bel *et al.* 1989), (Dubois 1989), (Dubois & Prade 1989), (Kerr & Walker 1989), fuzzy logic has been successfully employed to represent temporal constraints for knowledge-based scheduling. In those 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. Thus, 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 not using fuzzy evaluation. Our approach generalizes the mentioned ones to include, beside such temporal constraints, other kinds like chemical or organizational fuzzy constraints.

3.2.2 Constraints between Jobs

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 sets 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 following some 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. 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: Fuzzy compatibility rules for one element

The calculation of the nickel-compatibility is illustrated in figure 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 1, the heat h_3 must contain $h_3[Ni] = 1.2$ % of the chemical element nickel, while heat h_4 should contain only $h_4[Ni] = .1$ %. The relative percentage of $h_3[Ni]$ is therefore 1200 % of $h_4[Ni]$. Considering only nickel, the question is whether the sequence h_3 preceding h_4 is allowed or not, and if yes, how good is this sequence compared to other sequences. To decide this with the given fuzzy inference rules, somehow the vague linguistic variables and crisp but uncertain numeric values must be matched. This is done with fuzzy membership functions as defined in figure 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.* 1989) and (Kanemoto *et al.* 1990) where the computations are done in a comparable way.

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

$$\mathrm{comp}_{[Ni]}(h_3, h_4) = \mathrm{low}_{[Ni]}(h_3, h_4) \vee \mathrm{medium}_{[Ni]}(h_3, h_4)$$

is a new membership function defining the nickel-compatibility of h_3 preceding h_4, which is generally speaking more low than medium. The fuzzy-or operator represents the maximum of both arguments. We obtain as a result that the nickel-compatibility for h_3 preceding h_4 is more *low* than *medium*.

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 those 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 be at least present in this heat, the graduation is asymmetric, too, and only logarithmic on the right half. Beside 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 variables positively.

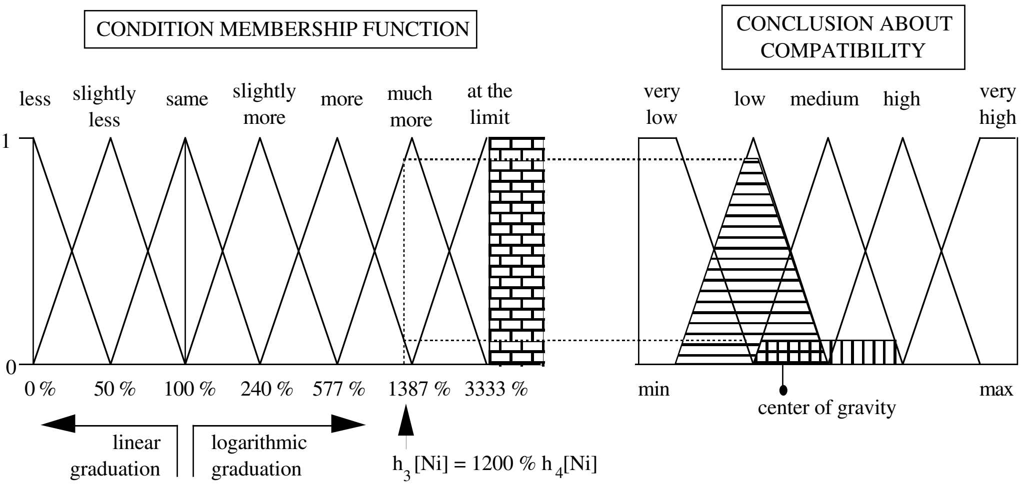

Figure 3: Computation of nickel-compatibility between heat h_3 and h_4 of table 1

The fuzzy inference rules like those in table 3 give several fuzzy judgments about the compatibility between heats. These judgments are in the form of membership functions and can be simplified to the linguistic variable 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:

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

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. 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:

$$\mathrm{defuzzy}(\mathrm{comp}(H_i, H_j)) = \frac{\int_{x=min}^{max} x\, \mathrm{comp}(H_i, H_j)(x)\, dx}{\int_{x=min}^{max} \mathrm{comp}(H_i, H_j)(x)\, dx}$$

This computation is done for every pair of jobs that may be scheduled. The result is a matrix of defuzzyfied 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 4 shows this matrix for our example. It will be used for the construction of the preliminary schedule and during the improvement process.

The values in the table are the previously used linguistic variables, since for the sake of understandability we have replaced the defuzzyfied values with the name of the fuzzy set to which the defuzzyfied value mainly belongs.

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. For a given schedule S with N jobs, the evaluation function is given by:

$$\mathrm{comp}(S) = \bigwedge_{i=1}^{N} \mathrm{comp}(H_i, H_{i+1})$$

3.2.3 Classification of Jobs

In a real-world application like scheduling a steelmaking plant, many optimization criteria compete with each other. (Chen *et al.* 1987) describe one approach to handle multi-objective scheduling using fuzzy sets. Similarly, the system presented in this paper 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*.

$H_0 \backslash H_1$	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	medium	medium	medium	high	high
h_1	medium	–	high	high	medium	medium	medium	medium	medium	medium	medium	high
h_2	low	medium	–	very high	low	low	low	high	medium	medium	medium	medium
h_3	low	medium	very high	–	low	low	low	high	medium	medium	medium	medium
h_4	high	medium	medium	medium	–	very high	medium	medium	high	high	medium	medium
h_5	high	medium	medium	medium	very high	–	medium	medium	high	high	medium	medium
h_6	medium	low	medium	medium	medium	medium	–	low	high	high	low	low
h_7	very low	very low	high	high	very low	very low	very low	–	low	low	low	low
h_8	low	low	very low	very low	low	low	medium	low	–	very high	medium	high
h_9	low	low	very low	very low	low	low	medium	low	very high	–	medium	high
h_{10}	medium	medium	medium	medium	low	low	medium	low	high	high	–	high
h_{11}	medium	low	low	low	very low	very low	low	low	very high	very high	very high	–

Table 4: Compatibility Matrix for Heats on Furnace EAF 1

The classification of jobs in the list is dependent 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 are to be scheduled that are cast into small ingots, 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 table 2, 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 1 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 1, we obtain the following classification of jobs:

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 5: Classification of Jobs

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:

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

3.3 Generating 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 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 with a duration 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 4 is used.

A special strategy is applied to prune the search space. It is comparable with preprocessing techniques in constraint satisfaction problems (CSP) as described in (Dechter & Meiri 1989). The objects in our CSP are the heats and 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 a job, it cannot be any more the successor of another 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 generation of a schedule from the orders given in table 1. We use the classification of jobs given in table 5. The whole schedule for both furnaces is shown in table 6. The system considers the jobs in the following sequence:

h_8: In the list of jobs given in table 1, 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.

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.

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 the second furnace, there is another job that will be produced on the caster. Since this job is cast with a different format, a maintenance interval between these jobs is necessary. 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.

h_7: The urgent and very important jobs are 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 possible. There are four potential successors with low compatibility. To generate fewer small intervals, job h_7 is scheduled before h_8.

h_9: Job h_9 is then scheduled optimally after h_8.

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.

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.

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

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

schedule 1-1: med, med, low, v. high, high, v. high, med, med, high, v. high, med

EAF 1	h_2	h_1	h_7	h_8	h_9	h_{11}	h_{10}	h_6	h_0	h_4	h_5	h_3

schedule 2-1: high, high, high, high, high, low, med, high, high, low, high

EAF 3	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 7pm 9pm 11pm 1am 3am 5am

Table 6: Intermediate schedule for example heats

The resulting schedule is illustrated in the first part of table 6. The compatibilities are shown in the line above the heat sequence. We assume that schedule 2-1 was constructed for the second furnace. The problems in the list for the second furnace are the molybdenum- and manganese-compatibilities. The algorithm to generate this schedule is simple and it is sketched in the following:

```
s := {};
list-of-jobs := classify(list-of-jobs);
matrix := classify(list-of-jobs);
repeat
  job := get-most-important(list-of-jobs);
  s := schedule(job);
  list-of-jobs := reclassify(list-of-jobs);
until very-important(job) ∉ list-of-job;
repeat
  while gaps-exist(s)
    s := fill-gap(s);
  repeat
    job := get-most-important(list-of-jobs);
    s := schedule(job);
    list-of-jobs := reclassify(list-of-jobs);
  until gap-exists(s) or empty(list-of-jobs)
       or full(s);
until empty(list-of-jobs) or full(s);
```

Algorithm 1: Generation of Schedule

3.4 Improving the Schedule by Relaxation of Constraints

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 one hand, hard constraints that may not be relaxed must be considered, 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 these antagonistic constraints, an evaluation function based on the fuzzy values seems to be more adequate, since the grade of the satisfaction of a constraint is evaluated too.

We have defined a repair strategy based on these fuzzy values. 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 probably h_7 causes the conflict because it is a very difficult job to schedule. A better place is sought such as 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 done easier. The result is shown in schedule 1-2 in table 7. 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, more improvements are possible.

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

Table 7: Intermediate schedule for example heats

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 1989), (Glover 1990) 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. However, to restrict the search, a tabu list describing which operations may not be performed anymore in the search process is used.

```
tabu-list := {};
s := current-best-schedule;
search-effort := 0;
repeat
  job1 := find-violation(s, tabu-list);
  op := choose-best-repair-operator(job1, s);
  case op:
    1: s := shift(job1, s);
    2: find(job2, list-of-jobs);
       s := exchange(s, job1, job2);
    3: find(job2, s);
       s := exchange(s, job1, job2);
  end case;
  if better(s, current-best-schedule)
    then current-best-schedule := s
    else tabu-list :=
                  add(tabu-list, {job1, op});
  search-effort := search-effort + 1;
until distance(s) / search-effort < limit;
```

Algorithm 2: Repair of Schedule

If no further constraint violation can be detected or no further improvement is achievable, the search for the best schedule ends. Judging whether a further improvement can 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 summation 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 as algorithm 2.

4 Comparison to Related Systems

In other projects it was shown that steel production is a worthy domain for the application of knowledge-based scheduling systems (Numao & Morishita 1989), (Shah *et al.* 1990), and (Stohl *et al.* 1991). In contrast to our application, these systems are used in plants for mass steel production where steel qualities do not vary as much as in our application. The main problem and bottleneck resource in these applications are the continuous casters. 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 contrast, we apply 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.

In (Shah *et al.* 1990) a minimization of waiting time is 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 (Kerr & Walker 1989), (Smith *et al.* 1986). 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 (Berry 1992), 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 (Kerr & Walker 1989). 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. In fact, it was even designed to this very end, i.e. to allow an easy formulation of subjective fuzzy rules. 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 (Kruse *et al.* 1991). Nevertheless, we must conceed that some membership function tuning is required to really get an application right. Especially, the overlapping of membership functions has been shown by (Bovery *et al.* 1991) to be a major influence factor, whereas their number or exact shape seem to be of minor importance.

(Woodyatt *et al.* 1992) 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.

5 Conclusion

Due to the highly unreliable data, the vague knowledge formulation, and the conflicting objectives in scheduling applications, mathematical-analytical methods as used in OR 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 relaxation. 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 considered. However, due to this ad hoc generation of schedules, some jobs usually remain unscheduled. We have proposed a "repair-based" control strategy that deals with several types of constraints (temporal, spatial, and chemical), and supports the dynamic relaxation of conflicting constraints. Although we have not compared this strategy with other approaches we believe that this is the most simple and efficient way to regard antagonistic constraints that have no clear hierarchical order.

Additionally, the generation of robust schedules is stimulated by using fuzzy sets. The heats that are cast on the continuos casters were scheduled on different ends of the schedule because this improves the robustness of the schedule. We describe the duration between both heats with a temporal fuzzy set that evaluates the schedule better in which the interval between both heats is longer.

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

Additionally, improper conditions for consecutive jobs require immediate and dynamic adaptation. We have supported this adaptation by providing the human expert assistance 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 figure 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 figure 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 variable to the next. Second, the compatibility rule can be changed for one element from 3 % to 2.5 %, which would mean that this latter element does not remain in the furnace as a residual to the same extent as the others. 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 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.

The idea behind our methodology is to allow easier modeling of the activities of human scheduling experts. Our prototype 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. Whether the solutions of our system or those of the previously developed expert system are better can not decided definitively. However, we assume that with the given techniques, better schedules can be generated, since the human expert can easily tune the problem solving process.

The described methodology was implemented in a prototype for the described application. We deploy this methodology at the moment for another steelmaking plant with different production characteristics and constraints. Later on, we intend to develop a reusable tool for different applications ranging over the whole steelmaking process.

Acknowledgments

We would like to acknowledge the work in knowledge acquisition done by Dorothea Czedik and Alexandra Eder, the helpfulness of the experts Ing. Pauker and Ing. Gromann, the frankness to new techniques of Dr. Meyer, the head of the steelmaking plant, Reza Shams for his work in the implementation of the expert system that has influenced our work heavily, as well as Rainhard Steindl for his implementation and critique of our fuzzy constraint based approach. Thanks also to Monte Zeben for careful review of this paper.

References

Barachini, F.; Theuretzbacher, N. 1988. The Challenge of Real-time Process Control for Production Systems. Proceedings of the 7th National Conference on Artificial Intelligence, St. Paul, 705–709.

Bel, G.; Bensana, E.; Dubois, D.; Erschler, J.; Esquirol, P. 1989. A Knowledge-Based Approach to Industrial Job-Shop Scheduling. Knowledge-Based Systems in Manufacturing, Andrew Kusiak (ed.), Taylor & Francis.

Berry, P. M. 1992. Scheduling: A Problem of Decision-Making Under Uncertainty. Proceedings of the European Conference on Artificial Intelligence, Vienna, 638–642.

Boverie, S.; Demaya, B.; Titli, A. 1991. Fuzzy Logic Control compared with other Automatic Control Approaches. Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, U.K.

Chen, D.; Pan, Y.; Xue, J. 1987. A Fuzzy Production System with Backtracking Control Strategy for Multiobjective Scheduling to a One-Machine-N-Parts Problem. Modern Production Management Systems, Andrew Kusiak (ed.), Elsevier.

Conway, R. W.; Maxwell, W. L.; Miller, L. W. 1967. Theory of Scheduling. Addison Wesley.

Dechter, R.; Meiri, I. 1989. Experimental Evaluation of Preprocessing Techniques in Constraint Satisfaction Problems. Proceedings of the 11th International Joint Conference on Artificial Intelligence, 271–277.

Dorn, J.; Shams, R. 1991. An Expert System for Scheduling in a Steelmaking Plant. Proceedings of the World Congress on Expert Systems, Orlando Fla., 395–404.

Dubois, D.; Prade, H. 1989. Processing Fuzzy Temporal Knowledge. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 4, 729–744.

Dubois, D. 1989. Fuzzy Knowledge in an Artificial Intelligence System for Job-Shop Scheduling. Applications of Fuzzy Set Methodologies in Industrial Engineering, Gerald W. Evans et al. (eds.), Elsevier, 73–89.

Glover, F. 1989. Tabu Search–Part I. ORSA Journal on Computing, Vol. 1, No. 3, 190–206.

Glover, F. 1990. Tabu Search–Part II. ORSA Journal on Computing, Vol. 2, No. 1, 4–32.

Kanemoto, M.; Yamane, H.; Yoshida T.; Tottori, H. 1990. An Application of Expert System to LD Converter Processes. Journal of the Iron and Steel Institute of Japan, ISIJ International, Vol. 30, No. 2,128–135.

Kerr, R.M.; Walker, R. N. 1989. A Job Shop Scheduling System Based on Fuzzy Arithmetic. Proceedings of the 2nd International Conference on Expert Systems and Leading Edge in Production and Operations Management, Hilton Head Island, S.C., 433–450.

Kruse, R.; Schwecke, E.; Heinsohn, J. 1991. Uncertainty and Vagueness in Knowledge Based Systems - Numerical Methods. Springer.

Maki, Y.; Masuda, Y.; Sawada, T.; Matsumoto, T.; Obata, H.; Takashima, N. 1989. Application of Fuzzy Theory for Automatic Control of Hot Stove Combustion Gas Flow. Proceedings of the 6th IFAC Symposium on Automation in Mining, Mineral and Metal Processing, Buenos Aires, 278–284.

Minton, S.; Johnston, M.; Philips, A.; Laird, P. 1989. Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method. Proceedings of the 8th National Conference on Artificial Intelligence, Menlo Parc, 17–24.

Negoita, C.V. 1985. Expert Systems and Fuzzy Systems. Benjamin/Cummings.

Numao, M.; Morishita, S. 1989. A Scheduling Environment for Steel-Making Processes. Proceedings of the 5th International Conference on Artificial Intelligence for Industrial Applications, Miami, 279–286.

Prade, H. 1979. Using Fuzzy Set Theory in a Scheduling Problem: A Case Study. Fuzzy Sets and Systems, Vol. 2, No. 2, 153–165.

Shah, M.J.; Damian, R.; Silverman, J. 1990. Knowledge Based Dynamic Scheduling in a Steel Plant. Proceedings of the 6th International Conference on Artificial Intelligence for Industrial Applications, St. Barbara, 108–113.

Smith, S.F.; Fox, M.S.; Ow, P.S. 1986. Constructing and Maintaining Detailed Construction Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems. AI Magazine 7(4),45–61.

Stohl, K.; Snopek, W.; Weigert, T.; Moritz, T. 1991. Development of a Scheduling Expert System for a Steelplant. Proceedings of the IFIP Conference on Experts Systems in Mineral and Metallurgy, Finnland.

Woodyatt, L.R.; Stott, K.L.; Wolf, F.E.; Vasko, F.J. 1992. Using Fuzzy Sets to Assign Metallurgical Grades to Steel. Journal of Metallurgy, February.

Zadeh, L.A. 1989. Knowledge Representation in Fuzzy Logic. IEEE Transactions on Knowledge and Data Engineering, Vol. 1, No. 1, 89–100.

Zweben, M.; Deale, M.; Gargan, R. 1990. Anytime Rescheduling. Proceedings of the DARPA Workshop on Innovative Approaches to Planning and Scheduling.