Applying Machine Learning for Solver Selection in Scheduling: A Case Study

Nysret Musliu

Vienna University of Technology
musliu@dbai.tuwien.ac.at

Abstract

We investigate the automated algorithm selection for a workforce scheduling problem that is solved by two different approaches. The solver based on constraint programming techniques has several advantages and it has been used successfully in the industry. However, this algorithm can not solve very large instances in a reasonable amount of time. The metaheuristic solver overcomes this limitation and is able to find solutions even for huge real world instances. We apply machine learning algorithms to select the best suited solver for a particular instance based on problem features. The preliminary experimental results on application of different learning techniques are presented.

1 Introduction

Automated algorithm selection is currently an intensive research topic in different problem domains. The aim is to select the most appropriate technique to solve a particular problem instance based on its features. Usually various supervised machine learning techniques are used to predict the best suited algorithm. An important issue in algorithm selection is to find appropriate features that characterize well the problem. Furthermore, the availability of sufficient large set of instances is crucial to train machine learning algorithms. Automated algorithm selection has been used successfully for several problems. Such problems include SAT [9], Nurse Rostering [4], Graph Coloring [7] etc. The reader is refereed to [8] for a survey on automated algorithm selection.

In this paper we investigate the application of machine learning techniques for solver selection in a situation when the decision maker should decide between two existing solvers. The particular case study we consider is the rotating workforce scheduling problem.

For the rotating workforce scheduling problem we have given ([6]):

- Number of employees: n.
- Set A of m shifts (activities) : $a_1, a_2, \ldots, a_m$, where a_m represents the special day-off "shift".
- w: length of schedule. The total length of a planning period is $n \times w$ because of the cyclic characteristics of the schedules.
- A cyclic schedule is represented by an $n \times w$ matrix $S \in A^{nw}$. Each element $s_{i,j}$ of matrix S corresponds to one shift. Element $s_{i,j}$ shows on which shift employee i works during day j or whether the employee has time off. In a cyclic schedule, the schedule for one employee consists of a sequence of all rows of the matrix S. The last element of a row is adjacent to the first element of the next row, and the last element of the matrix is adjacent to its first element.
- Temporal requirements: The requirement matrix R $((m-1) \times w)$ (we use here $m-1$, because shift a_m represents the day-off), where each element $r_{i,j}$ of the requirement matrix R shows the required number of employees for shift i during day j.

The aim is to find a cyclic schedule (assignment of shifts to employees) that satisfies the requirement matrix, and the constraints about sequences of shifts not permitted to be assigned to employees, maximum and minimum length of periods of consecutive shifts, and maximum and minimum length of blocks of workdays.

To solve this problem several exact and heuristic methods have been proposed in the literature [1], [3], [6], [5] etc.

2 Applying classification techniques for solver selection

In this section we shortly describe the applied solvers and give the information for the instances and the features that are used to characterize these problems. Furthermore, the preliminary experiments on application of machine learning algorithms for solver selection are presented.

2.1 Solvers

The solver called First Class Scheduler (FCS) proposed in [6] is based on constraint programming techniques and it involves the user in different phases for selection of partial solutions. The software has been used in industry since 1999 as part of a shift scheduling package called Shift-Plan-Assistant (SPA) of XIMES[1] Corp. FCS is a complete solver and it is very appropriate to be used in the consulting process, where different solutions should be provided and discussed. However, FCS does not show very good results on very large instances and sometimes one should wait too long for a solution (or the solver can not find a solution within hours). As an alternative for such problems a metaheuristic solver was developed in [5]. This solver is a combination of min-conflicts heuristic and tabu search and it gives a solution for all available benchmark instances in a short amount of time. The prediction of performance of both solvers would help the consultant to decide which solver to use in situations when she quickly needs a solution that should be discussed with other relevant decision makers.

In this paper we consider the following scenario regarding automated solver selection. For a new instance that has to be solved the features of problem are extracted and a machine learning technique is used to predict if that instance can be solved by FCS in a reasonable amount of time. If the prediction is positive the FCS solver is used, otherwise the heuristic solver will be applied.

2.2 Problems and their features

In the previous works 20 real life problems were used for evaluation of different techniques [5]. This set is too small to be used as a training set for machine learning algorithms. Therefore, we generated random problems based on the characteristics of real world examples. Our set of examples consists of 1191 problems. Both solvers were executed in all problems and the running time of solvers was recorded (for the metaheuristic solver the average running time in 10 runs is recorded).[2]

In the current experiments for each problem the following features are extracted: number of employees, average minimum length of shifts, average maximum length of shifts, smallest maximum length of shifts, largest maximum length of shifts, smallest minimum length of shifts, largest minimum length of shifts, minimum length of days-off blocks, maximum length of days-off blocks, minimum length of blocks of workdays, maximum length of blocks of workdays, and number of forbidden shift sequences.

2.3 Preliminary results

In our first experiment we applied several machine learning techniques to predict if the constraint based solver FCS can solve the problem in a reasonable amount of time. The answer to this question is very important in the consulting process, because FCS is an interactive system that consists of four phases and waiting too long for solutions is not possible in a discussion session. Our training data set contains 272 examples that could not be solved by FCS in a reasonable amount of time.

We experimented with these machine learning algorithms: k-nearest neighbor, naive bayes, decision tree and random forest. The implementations available in WEKA software package [2] were used and we experimented with different parameters for these algorithms. To compare the algorithms we used the 10-fold cross-validation. Currently, the best results are obtained with the random forest technique. By this algorithm we could achieve 81.2% classification accuracy. Out of 223 misclassified examples, 52 examples were classified as false negative and 171 as false positive.

[1] http://www.ximes.com/

[2] Instances and the data set can be downloaded here: www.dbai.tuwien.ac.at/staff/musliu/benchmarks/rotatingschedules/

In the scenario when the algorithm predicts that FCS will not find a solution, an interesting question is how long it will take for the metahuristic solver to find the first solution. In the current version we experimented only with classification algorithms and the running time of heuristic algorithm was divided into three classes: problems solved within 10 seconds, problems solved within 100 seconds, and problems solved within 260 seconds. Currently, the best results are also obtained with the random forest algorithm. By this technique we could achieve 88.9% accuracy. However, the algorithm does not show very good classification results in the third class that contains 23 problems for which the solver needed more than 100 seconds to generate a solution. To improve the results for classes which contain much less instances we also experimented with cost-sensitive learning. Our initial experiments showed that the results for classes with few instances could be improved, while the overall classification accuracy was slightly worse.

3 Conclusions

We applied the machine learning techniques for solver selection in a real life scheduling domain. A large set of problems was created to train the learning algorithms and several relevant features were identified. The current results regarding the classification accuracy are promising. However, the prediction of classes with small number of examples should be still improved.

For the future work we will consider more learning algorithms and plan to investigate more deeply the application of cost-sensitive learning to improve results for imbalanced data. One another important issue is the consideration of more features for this problem. This includes features similar to those used for nurse scheduling problem [4] and other new features specific to this problem. Furthermore, it would be interesting to consider the application of regression techniques to predict the runtime of our solvers in the new instances.

Acknowledgments

The work was supported by the Austrian Science Fund (FWF): P24814-N23.

References

[1] N. Balakrishnan and R. T. Wong. A network model for the rotating workforce scheduling problem. *Networks*, 20:25–42, 1990.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[3] G. Laporte and G. Pesant. A general multi-shift scheduling system. *Journal of the Operational Research Society*, 55/11:1208–1217, 2004.

[4] T. Messelis and P. De Causmaecker. An algorithm selection approach for nurse rostering. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence,*, pages 160–166. Nevelland, Nov. 2011.

[5] N. Musliu. Heuristic methods for automatic rotating workforce scheduling. *International Journal of Computational Intelligence Research*, 2(4):309–326, 2006.

[6] N. Musliu, J. Gärtner, and W. Slany. Efficient generation of rotating workforce schedules. *Discrete Applied Mathematics*, 118(1-2):85–98, 2002.

[7] N. Musliu and M. Schwengerer. Algorithm selection for the graph coloring problem. In *Learning and Intelligent OptimizatioN Conference LION 7, Catania, Italy*, 2013.

[8] K. A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):6:1–6:25, jan 2009.

[9] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Satzilla: Portfolio-based algorithm selection for sat. *J. Artif. Intell. Res. (JAIR)*, 32:565–606, 2008.