Interactive Optimization with Long-Term Preferences Inference on a Shift Scheduling Problem

David Meignan*   Sigrid Knust†

Abstract
In this article we describe a method for enriching interactively an optimization model with long-term preferences and illustrate it by a use case. Interactive enrichment is the process of adjusting an optimization model using the feedback of a user on intermediate results. The integration of user's preferences allows capturing aspects of the optimization problem that have not been modelled initially. In existing interactive optimization approaches, preferences are typically short-term, i.e. specific to the problem instance. The proposed approach investigates long-term preferences that can be reused for solving multiple problem instances. The interactive optimization method has been developed for solving a shift scheduling problem. An Iterated Local Search (ILS) metaheuristic provides solutions that have to be evaluated by the user. Based on this feedback of the user, long-term preferences are inferred as conjunctive rule sets. The user can then add some of the proposed conjunctive rule sets to enrich sustainably and transparently the optimization model with new soft constraints. For testing the proposed interactive enrichment method a decision support system prototype has been implemented. Preliminary results allowed identifying interesting perspectives and applications for the approach.

Keywords: Interactive optimization, shift scheduling, decision support, rule inference.

1 Introduction

In an interactive optimization system, the user or decision maker interacts during the solving process and by his feedback on intermediate results he can significantly modify final solutions or performance of the optimization process. At present, different interactive optimization methods have been proposed to address several issues related to the integration of optimization methods in decision support tools.

First, the user may guide the search process in order to obtain better performance. For instance, the user may know some characteristics of the problem instance that are not fully exploited by the optimization algorithm. In addition, when the optimization method is unable to produce an optimal solution in a reasonable amount of time, the user may also contribute to determine a compromise between computational time and quality of the solution. This interactive guiding process has been investigated in Human-Guided Search (HuGS) approaches [10, 13] and is also related to interactive parameter optimization approaches [9].

Second, interaction may be used to enrich or adjust the optimization model. In many situations where an optimization method is integrated into a decision support tool, the solutions of the optimization method have to be adjusted in order to capture additional aspects of the real problem. In an interactive context, the problem-domain expertise of the user may be exploited by the optimization method for generating more realistic or relevant solutions. Interactive enrichment or adjustment of the optimization model is considered in interactive approaches for multiobjective optimization [14] and, to some extent, in interactive evolutionary algorithms [16].

Finally, interaction between the user and the optimization method may facilitate the acceptance of the system by the user as well as favour the understanding of the optimization problem. This may results in a more efficient use of the decision support tool. An overview of individual learning in the context of interactive multiobjective optimization is given in [1], and we refer the reader to [15, 4] for the main concepts about user acceptance and user trust in the context of decision aids.

We identify two limitations of existing interactive optimization approaches for enriching or adjusting an optimization model. First, preferences are typically short-term, i.e. the adjustments made on the optimization model are specific to a problem instance. Second, the possibilities to *enrich* an optimization model with new features that are not initially modelled are limited. For example, in interactive approaches for multiobjective optimization problems the feedback of the user allows determining preferences related to the objectives considered. These preferences are either trade-off values between objectives (trade-off based methods), or expected objective values (reference point approaches), or priority ranks of objectives (classification-based methods) [14]. The user evalu-

*Dr. David Meignan is a researcher at the Institute of Computer Science, University of Osnabrück, Osnabrück, Germany (email: dmeignan@uos.de)

†Prof. Dr. Sigrid Knust, Institute of Computer Science, University of Osnabrück, Osnabrück, Germany (email: sigrid.knust@uni-osnabrueck.de)

ates several intermediate solutions for progressively adjusting the preferences related to the objectives. This interactive process that has been proven successful for solving multiobjective optimization problems is, however, limited to the adjustment of preferences related to the objectives. The type of preference is specified in advance. The interactive enrichment with new constraints or objectives is not considered. In addition, the preferences learned for solving a given problem instance may not be valid or reusable for another problem instance. It is, therefore, necessary to repeat the interactive adjustment process for solving additional optimization problem instances, and the accumulated amount of information about preferences is lost.

The method proposed in this study addresses these two limitations by introducing an interactive enrichment procedure with long-term preferences. These long-term preferences are new soft constraints inferred from user feedback that can be added to the optimization model and reused for solving multiple problem instances. In addition, only few assumptions are made on the preferences, which allow enriching the model with a wide variety of possible new constraints.

The proposed interactive enrichment method has been developed for a shift scheduling problem [5]. This problem basically corresponds to the optimization of staff schedule according to a demand while satisfying contract requirements and staff preferences.

For solving a problem instance, the interactive process alternates between an optimization step for computing a solution, and an evaluation phase where the user gives a feedback on the solution for guiding the next optimization step. Candidate solutions are generated by an Iterated Local Search (ILS) metaheuristic. Each solution is a complete schedule of the employees. On proposed schedules, the user can identify the assignments of employees to shifts that are satisfactory and the ones that are inadequate. These preferences on assignments are used in the next optimization steps to guide the search toward a more satisfactory solution that integrates practical aspects of the problem that are not initially modelled. These assignment evaluations are short-term preferences considering that they cannot be re-used as it is for other schedule periods. With this short-term preference mechanism the user has to adjust the proposed schedules for each new problem instance, even if the preference can be generalized. The goal of the proposed long-term preference enrichment method is to address this limitation by extracting preferences that can be re-used. In addition to assignment evaluations that guide the search, the accumulated feedback of the user is dynamically analysed to identify potential long-term preferences. These preferences are inferred as conjunctive rule sets. The more accurate rule sets are presented to the user. He can then add some of the rule sets as new soft constraints.

In the next sections we present this approach through a use case. The studied optimization problem is introduced in Section 2.1. Then, the ILS and the short-term preference mechanism are described in Sections 2.2 and 2.3 respectively. Finally, the inference and integration of long-term preferences are presented in Sections 2.4 and 2.5.

A decision support prototype has been implemented and tested on real datasets [11]. Preliminary results and perspectives are discussed at the end of the article.

2 Use case

The remainder of the paper will focus on the following use case. A manager uses a planning system that includes the proposed interactive enrichment method for determining the week schedule of his employees. After having specified the input data such as the availability of employees and the required number of employees per shift, the system proposes an initial schedule. However, this solution is not entirely satisfactory. There are some exceptions, specific to the planning period, that must be manually defined (e.g. an employee that should be scheduled at the same time as another employee). In addition, as a more general rule the manager tries to not schedule employees with part-time contract on Saturdays. The manager expresses these preferences by identifying assignments that are inadequate. Additionally he can defined which assignments are preferred instead. The optimization procedure is run once again to try to satisfy the user preferences and schedule requirements. New assignment preferences can be iteratively defined by this process.

After having accumulated enough feedback from the user the system is able to extract some patterns to identify inadequate assignments. In the considered use-case, the system identifies that an employee with part-time contract assigned on Saturday is not satisfactory. This rule is proposed to the user which has the possibility to add it as a new soft constraint in the optimization model. If the rule is validated by the manager, the subsequent solutions provided by the system will automatically integrate this new constraint. Contrary to assignment preferences, the added constraint can be used for solving multiple problem instances.

2.1 Shift scheduling problem

The optimization problem considered in this study and depicted in the use-case is a shift scheduling problem adapted from [11]. The optimization model in [11] deals with shift scheduling of tank trucks. It has been reformulated into a more general shift scheduling problem. The objective is to optimize the assignments of employees to shifts for a given planning period. A shift is a time interval in a day (e.g. early 7:00-15:00, late 15:00-22:00) over which employees has to be scheduled [2]. A number of employees required by day for each shift is defined. A solution to the problem is a set of daily assignments of employees to shifts such as the required number of employees by shift and by day is satisfied. In addition to this constraint related to the staff demand, a solution must satisfy a set of hard constraints and optimize some soft constraints.

Hard constraints considered in the optimization model are the following:

Coverage: Each employee is assigned to at most one shift per day. The number of employees assigned to a shift on a day must match the demand. An employee can be assigned to a shift only if he is available.

Skill: An employee is assigned to a shift slot only if he has the required skills.

Maximum week working time: The working time per week of an employee must not exceed the maximum week working time defined by the contract.

Consecutive shift type exclusion: Some shift types cannot be combined on two consecutive days (e.g. assigning an early shift the day after a night shift is not permitted).

Initial soft constraints are the following:

Isolated day: Isolated working days and days off should be avoided.

Desired working time: The total working time should match as much as possible the desired working time.

Multiple shift types: The number of different shift types assigned to an employee in a week should be minimized.

Preferred shift type: Preferred and undesired shift types for specific assignments should be maximized and minimized respectively.

2.2 Iterated local search

As described in the use-case, the optimization procedure provides the initial solution and also re-optimizes the current solution when requested by the user. The solutions are computed by an ILS metaheuristic [12]. ILS iteratively performs two steps until a stopping criterion is met. After having generated and improved the initial solution, the first step perturbs the current solution. The second step improves the perturbed solution by local-search. Finally, a criterion determines if the resulting solution is accepted for the next iteration.

In our implementation of ILS, the initial solution is generated by a greedy algorithm. The local search procedure is a local descent using block-swap neighbourhood [3]. Perturbation is produced by random block rotations.

A parallel implementation of the ILS allows obtaining satisfactory solutions in a few seconds for the considered datasets. The stopping criterion corresponds to a maximum number of iterations and a time limit. However, the user can dynamically visualize the current best found solution with the detail of the cost. He can stop the optimization process at any time.

2.3 Feedback and short-term preferences

When a solution computed by the ILS procedure is proposed to the user, he can give a feedback on it. The feedback exploited by the proposed interactive approach is an evaluation of employees' assignments. The user can identify the assignments that are satisfactory and the ones that are inadequate.

Figure 1 represents a hypothetical schedule presented to the manager in the use-case. The shift assignments have been optimized according to the initial data and constraints. Some of the assignments have been marked by the manager as inadequate and others as satisfactory. These evaluations are respectively represented by crosses and ticks.

Employees	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
Employee 1 (Part-time)	Early	Early	Early			Early	
Employee 2 (Part-time)		Late	Late	Late	Late	Late	
Employee 3 (Part-time)	Night	Night	Night			Early	
Employee 4 (Part-time)			Early	Early	Early		
Employee 5 (Full-time)		Early	Early	Early	Early	Early	
Employee 6 (Full-time)		Late	Late	Late	Late		

Figure 1: A hypothetical schedule evaluated by a user. Crosses represent assignments considered as inadequate by the user, and ticks are satisfactory assignments. Emply slots in white are non-working days.

This feedback is used in two different ways by the interactive optimization method. The first mechanism directly uses these assignment preferences in the optimization procedure. Inadequate assignments generate a penalty if they appear in the solution and assignments marked as satisfactory generate a penalty if they are not present in the solution. The penalties correspond to two additional constraints introduced in the optimization model with a lexicographic priority ordering [6]. Hard constraints are considered first, then the inadequate assignments constraint, followed by the satisfactory assignments constraint, and finally soft constraints.

Considering the schedule and assignment preferences in Figure 1, the user can re-optimize the solution with the ILS procedure. For this new run, the optimization procedure will search for a solution that avoids inadequate assignments, preserves satisfactory assignments while satisfying hard constraints and minimizing soft constraints.

This first mechanism, allows integrating aspects of the problem that are not initially modelled. However, assignment evaluations are short-term preferences since they cannot be re-used for optimizing another schedule period. The second mechanism tries to generalize the feedback in order to enrich the optimization model with long-term preferences.

2.4 Long-term preference inference

In the proposed approach, long-term preferences are a generalization of the user's feedback. Note that the feedback can be accumulated on different planning periods. In Figure 1 a potential generalization of the feedback is that employees with part-time contract should not work on Saturday. The objective of the long-term preference inference procedure is to extract such a pattern in order to enrich the optimization model with new soft constraints. Due to the possible impact of adding a soft constraint, the long-term preference inferred must be understood by the user and validated before being added to the model.

In order to identify such a pattern in the feedback, the assignment preferences are characterized by a set of attribute values. Eighteen attributes have been used to characterize the assignment preferences. Figure 2 reports the values of some attributes for the preferences given in Figure 1. Each assignment preference corresponds to a new set of attribute values. For instance, the inadequate assignment of `Employee 1` to the `Early` shift on `Saturday` is characterized by the first line of attribute values in Figure 2. The value of the last attribute named `satisfactory` indicates whether the assignment has been evaluated as inadequate or satisfactory by the user.

Attributes

Feedback	employee	employee -contract	day	shift -type	work	...	satisfactory
	Employee 1	Part-time	Sat.	Early	Yes		No
	Employee 2	Part-time	Sat.	Late	Yes		No
	Employee 3	Part-time	Tue.	Night	Yes		Yes
	Employee 3	Part-time	Wed.	Night	Yes		Yes
	Employee 3	Part-time	Sat.	Early	Yes		No
	Employee 4	Part-time	Fri.	Early	Yes		Yes
	Employee 4	Part-time	Sat.	No	No		Yes
	Employee 5	Full-time	Fri.	Early	Yes		Yes
	Employee 5	Full-time	Sat.	Early	Yes		Yes

Figure 2: Characterization of the user's feedback of Figure 1. Each assignment preference corresponds to a set of attribute values.

Based on this characterization of the feedback, long-term preferences are extracted as conjunctive rule sets that classify the assignment preferences according to the `satisfactory` attribute value. Rule sets consist of rules that are "OR-ed" together. Premises of individual rules are tests on attributes that are "AND-ed" together. Only classifiers for inadequate assignments are considered. The preference "employees with part-time contract should not work on Saturday" can be expressed by the conjunctive rule: `IF employee-contract = "Part-time" AND day = "Saturday" AND work = "Yes" THEN satisfactory = "No"`. The objective of the inference mechanism is to find such a rule set in the user's feedback.

Conjunctive rule sets have been adopted to represent long-term preference, first, because it is an efficient way to represent a lot of constraints for shift scheduling problems. In addition, conjunctive rule set is one of the most understandable classifier. Finally, various machine learning algorithms can infer conjunctive rule sets from the proposed characterization of the feedback [17].

Conjunctive rule sets are extracted by a separate-and-conquer algorithm without global optimization [8]. The method is quite similar to PART [7]. Each rule in a set is built from a depth-first search path in a decision tree as illustrated in Figure 3. Tests in the decision tree are selected using the information gain heuristic [17, chap. 4 § 3]. When a rule is built from a decision tree, covered instances are removed, and additional rules are created for the remaining instances. A minimum coverage criterion stops the process. Several rule sets are generated using different first tests.

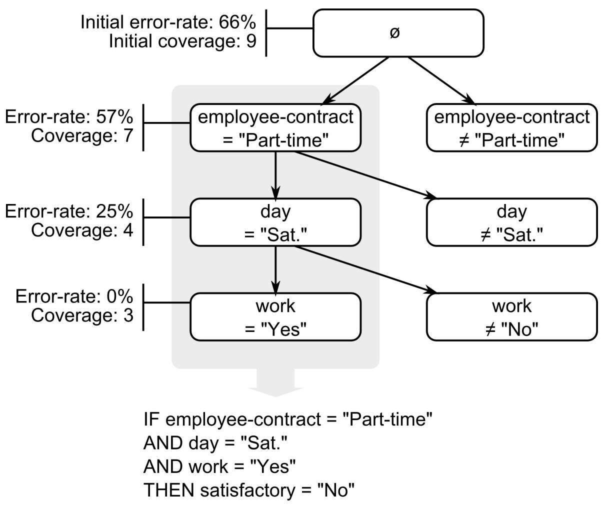

Figure 3: Decision-tree exploration for the feedback of Figure 2. The path to the leaf corresponds to the conjunctive rule `IF employee-contract = "Part-time" AND day = "Saturday" AND work = "Yes" THEN satisfactory = "No"`.

Conjunctive rule sets with the lowest error rates and shortest lengths are presented to the user. They correspond to potential long-term preferences. Computational time to infer these rule sets does not exceed few seconds for hundreds of assignment evaluations (i.e. set of attribute values depicted in Figure 2). Thus, potential long-term preferences can be updated dynamically when the user gives a feedback on a solution.

2.5 Long-term preference integration

When a long-term preference is presented to the user, he can add it to the optimization model as a new soft constraint. In the use-case, the inferred preference corresponds to the conjunctive rule set: `IF employee-contract = "Part-time" AND day = "Saturday" AND work = "Yes" THEN satisfactory = "No"`. If the user validates this conjunctive rule set, a new constraint is added to the optimization model. Then, for the next optimization steps the ILS procedure will try to satisfy this constraint in addition to initial hard and soft constraints.

Evaluation of a conjunctive rule set for a solution is based on the classification of assignments. The rule set is evaluated on each assignment of the solution. If the assignment is classified as inadequate (i.e. `satisfactory = "No"`), the constraint is considered as violated for the assignment.

Conjunctive rule set constraints can be added dynamically to the optimization model. Integration of these long-term preferences does not require significant changes for the ILS procedure. However, the local-search procedure has been optimized with *delta-evaluation*[1] of solutions for the exploration of the neighbourhood. During the local search, the costs of solutions in a neighbourhood is computed from the costs of the solution at the origin and the difference of costs resulting from the moves. The difference of costs between two neighbouring solutions may be computed faster than a full evaluation of a solution. For conjunctive rule set constraints, the evaluation of the number of assignments violated by a constraint is optimized by this method. Note that this optimization depends on the attributes in the rule set.

3 Conclusion

A decision support system prototype has been implemented to test the proposed method and conduct preliminary experiments. The datasets presented in [11] have been adapted for this purpose. The goal of preliminary experiments was twofold. First, computational performances of the ILS procedure were evaluated. In addition, a second part of the experiments aimed at confirming that long-term preferences can be extracted from a user feedback with the proposed method.

The 15 problem instances adapted from [11] have been solved using IBM® ILOG CPlex 12, with a time limit of 1 hour per instance. The best found solutions (optimal solution for 9 of 15 instances) have been compared to the results of the ILS procedure with a time limit of 60 seconds on a standard PC (Intel® Core™ i5 2.5GHz, 2GB memory). The ILS procedure is able to solve all hard constraints. The average gap between ILS and best found solutions is 9.2%. This gap represents an average of 1.1% of additional assignments for which soft constraints are not satisfied. This first result suggests that the implemented ILS procedure could be adequate for an interactive optimization approach.

Inference of long-term preferences has been tested in a second set of experiments. These experiments are based on a simulated feedback. An automatic procedure generates the feedback of some given preferences by evaluating the assignments of solutions. Long-term preferences are then extracted and compared to the expressed preferences. Preliminary results indicate that the inference method can efficiently generalize the feedback. Conjunctive rule sets that correspond to the exhibited preferences are obtained within few evaluations of intermediate solutions.

Future work on this interactive approach for the shift scheduling problem will focus on three aspects. First, in its current form the proposed method is not able to handle the “noise” in the feedback. If random inadequate assignments are introduced in the feedback (for simulating exceptional preferences that should not be generalized), the resulting conjunctive rule sets will contain additional rules or tests for covering these exceptions. Different improvements of the inference method have already been identified to deal with this issue. Second, we plan to conduct an extensive experiment with real users. The objective is to evaluate the potential impact of the interactive approach on the decision-making process, and determine if the system is efficient for real and long-term usages. Finally, we intend to generalize this interactive optimization approach to other optimization problems and also investigate additional presentation and interaction modes in the context of interactive optimization.

[1]sometimes referred to as *incremental evaluation*

Acknowledgments

The authors would like to thank the contribution of Google for supporting the project under the *Google Focused Grant Program on Mathematical Optimization and Combinatorial Optimization* 2012.

References

[1] V. Belton, J. Branke, P. Eskelinen, S. Greco, J. Molina, F. Ruiz, and R. Slowinski. *Multiobjective Optimization*, chapter Interactive multiobjective optimization from a learning perspective, pages 405–433. Springer, 2008.

[2] E. K. Burke, P. D. Causmaecker, G. V. Berghe, and H. V. Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7:441–499, 2004.

[3] E. K. Burke, T. Curtois, R. Qu, and G. V. Berghe. A time predefined variable depth search for nurse rostering. *INFORMS Journal on Computing*, Published online, June 6, 2012.

[4] F. Davis, R. Bagozzi, and P. Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, 35(8):982–1003, 1989.

[5] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004.

[6] P. C. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20(11):1442–1471, 1974.

[7] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 144–151, 1998.

[8] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13:3–54, 1999.

[9] F. Hutter, T. Bartz-Beielstein, H. Hoos, K. Leyton-Brown, and K. Murphy. Sequential model-based parameter optimisation: an experimental investigation of automated and interactive approaches. In T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, editors, *Empirical Methods for the Analysis of Optimization Algorithms*, chapter 15, pages 361–411. Springer, 2010.

[10] G. W. Klau, N. Lesh, J. Marks, and M. Mitzenmacher. Human-guided search. *Journal of Heuristics*, 16(3):289–310, 2010.

[11] S. Knust and E. Schumacher. Shift scheduling for tank trucks. *Omega*, 39:513–521, 2011.

[12] H. R. Lourenço, O. C. Martin, and T. Stützle. *Handbook of Metaheuristics*, chapter Iterated Local Search, pages 321–353. Kluwer Academic, 2003.

[13] D. Meignan, J.-M. Frayret, and G. Pesant. An interactive heuristic approach for the p-forest problem. In *2011 IEEE International Conference on Systems Man and Cybernetics (SMC)*, pages 1009–1013, 2011.

[14] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. *Multiobjective Optimization*, chapter Introduction to Multiobjective Optimization: Interactive Approaches, pages 27–58. Springer, 2008.

[15] B. M. Muir. Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies*, 27(5-6):527–539, 1987.

[16] H. Takagi. Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

[17] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining, Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, third edition, 2011.