Formal Languages for Integer Programming Modeling of Shift Scheduling Problems

Marie-Claude Côté[1], Bernard Gendron[2], Claude-Guy Quimper[3], and Louis-Martin Rousseau[1]

[1] École Polytechnique de Montréal], PO Box 6079, succ. Centre-Ville, Montréal, H3C 3A7, Canada
marie-claude.cote@polymtl.ca, louis-martin.rousseau@omegaoptimisation.com
[2] Université de Montréal, PO Box 6128, succ. Centre-Ville, Montréal, H3C 3J7, Canada,
bernard@crt.umontreal.ca
[3] Omega Optimisation, 4200 St-laurent #407,Montréal, H2W 2R2, Canada,
cgquimper@optime.net

Given a sequence of n decision variables X_i, each with a finite domain D_i, $i = 1, \dots, n$, a constraint on such a sequence is a set of n-tuples $L \subseteq D_1 \times \dots \times D_n$ called a *language*. The constraint over the sequence is satisfied when the tuple $\langle X_1, \dots, X_n \rangle \in L$ belongs to the language. Such constraints arise in many optimization and satisfaction problems. In this paper, we focus on shift scheduling problems, where a sequence of activities (work activities, break, lunch, rest) must be assigned to a set of employees. In these problems, the difficulty lies in building shifts that comply with work regulations such legal placement of breaks and lunches, and transitions between activities.

In this paper, we study how to model constraints on sequences of decision variables using a Mixed Integer Programming (MIP) framework. Our approach is inspired by global constraints in Constraint Programming (CP) that use formal languages. First, we suggest using automata to represent constraints on sequences of decision variables, as the CP `regular` [1] constraint does. From the automaton, we automatically generate a network flow model that can be included into any MIP model involving constraints on sequences of decision variables. Second, we propose a way to use context-free grammars instead of automata to describe the constraints on sequences of decision variables. To apply this to MIP, we use an and/or graph structure associated to the CP `grammar` constraint [2, 4, 3] and derive the associated linear constraints.

These approaches allow MIP to benefit from CP expressiveness in modeling, by automatically generating MIP models from intuitive modeling tools, such as automata and context-Free grammars. Furthermore, our experimental results on a shift scheduling model show that they can not only facilitate the modeling, but also give models easier to solve by MIP solvers compared to classical MIP formulations.

References

1. G. Pesant. A regular language membership constraint for finite sequences of variables. In *Proceedings of the Tenth International Conference on Principles and Practice of Constraint Programming (CP 2004)*, pages 482–495, 2004.
2. C.-G. Quimper and T. Walsh. Global grammar constraints. In *Proceedings of the Twelfth International Conference on Principles and Practice of Constraint Programming (CP 2006*, pages 751–755, 2006.
3. C.-G. Quimper and T. Walsh. Decomposing global grammar constraints. In *Proceedings of the Thirteenth International Conference on Principles and Practice of Constraint Programming (CP 2007)*, pages 590–604, 2007.
4. M. Sellmann. The theory of grammar constraints. In *Proceedings of the Twelfth International Conference on Principles and Practice of Constraint Programming (CP 2006*, pages 530–544, 2007.