Reformulation techniques for a class of permutation problems

Toni Mancini

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
tmancini@dis.uniroma1.it

The long-term goal of our research is to provide syntactic criteria for the automatic reformulation of combinatorial and constraint problems specifications, in order to improve solver efficiency. Our current investigations focus on the selection of constraints that can be safely "delayed" and solved afterwards, yet guaranteeing that every solution of the reformulated spec can be reconduced, via the application of the delayed constraints, to a solution of the original one. A consequence of this approach is that the set of solutions is enlarged, thus hopefully obtaining a speed-up in the solving process.

In this paper we focus on reformulation techniques for the subclass of *permutation problems* characterized by constraints that bind an element of the permutation either to the next or the previous one. A paradigmatic example is the *Hamiltonian circuit* (HC) of a graph problem, which aims to find a permutation of the graph nodes (i.e., a bijective mapping from the n nodes to the integer range $[1..n]$) s.t. every node is linked to its successor, and the last to the first one. Another example is the *permutation flow-shop scheduling* problem. In this context, we identify the Injective constraint (Inj) as *safe-delay.* By ignoring it and slightly changing the other constraints, we obtain possible reformulations of the original problems. As an example, for HC, removing (Inj) results in clusters of nodes that can be visited in an arbitrary way (thus, identifying only a partial order among them). To guarantee that every solution of the reformulated problem can be reconduced to a solution of the original one, the adjacency constraint has to be extended to nodes with the same order number, thus forcing clusters to be cliques. In this way, we allow abstraction from the level of nodes to the level of clusters, by first identifying cliques, and then solving the HC problem on the abstract graph. Further elaborations of the above idea are possible, leading to two additional reformulations of the original problem which result to abstract graphs with less edges. As a consequence, they seem more efficiently evaluable wrt the original spec, especially for negative instances, for particular classes of solvers (e.g., SAT based ones). As for the planned work, we aim to extend the experimentation through the use of state-of-the-art solvers for CP, e.g., OPL.

From a methodological point of view, it is important to emphasize that our goal is not to provide a new and more efficient algorithm for solving a particular problem, but to show that consistent speed-ups can be achieved by a mere and syntactically-based reformulation of a pure declarative spec, which can be in principle performed automatically by the system in a preprocessing stage.

Acknowledgments. The author is grateful to his advisor, Prof. Marco Cadoli, for guiding, supporting, and encouraging his research activity.