

Using ECLiPSe to solve large-scale piecewise linear scheduling problems

ECLiPSe provides us with an excellent tool to create hybrid algorithms to solve large-scale combinatorial optimization problems (**LSCO**s). A particularly hard LSCO class, resource constrained scheduling with a non-convex piecewise linear objective function, has been tackled in IC-Parc, resulting in several extremely powerful algorithm implementations.

Resource constrained scheduling problems with piecewise linear objective function

In the literature, combinatorial optimization problems that have the scale and complexity of real-world requirements are called large-scale combinatorial optimization problems (**LSCO**s). They are interesting because of their commercial importance (for example production and transportation scheduling, risk management and communication network design), and for their NP-hardness. General polynomial-time algorithms may never be found to solve them. Nevertheless, most LSCO's have characteristics that can be exploited to improve algorithm efficiency and thus reduce computational effort.

In IC-Parc, LSCO's are approached from a constraint programming (CP) angle. Especially, the **ECLiPSe** CP platform strongly supports such algorithm specialization with various useful facilities such as attributed variables, tentative values and waking priorities. The source of an LSCO's complexity is often its symbolic and disjunctive constraints, and CP provides a paradigm that supports the development of simple and powerful techniques to satisfy them.

A typical LSCO includes several sub-problems with different characteristics. The nature of real world requirements means that LSCO's are unique because of the relative sizes of their sub-problems, and the differences among the constraints that relate them. Taking these two features and the large-scale of the input data, it is unlikely that a single algorithm can deal with them perfectly without extension or modification. Thus, there is an emerging interest in the CP research community to focus on the **hybridization** of several specialized algorithms when solving LSCO's.

One such real-world based LSCO is the kernel resource feasibility problem (**KRFP**), which generalizes most scheduling benchmarks, including job shop scheduling, ship loading, bridge building, and resource constrained project scheduling. In the KRFP, we have several types of resources and a given quantity of each. We have also a fixed number of non-preemptive activities. Each activity requires a quantity of a specific resource type during its execution. An arbitrary set of metric temporal constraints relate the start and end times of activities. The aim is to schedule activities so that all the temporal constraints are satisfied, and the demand on any resource does not exceed its resource quantity at any time on the scheduling horizon.

While the KRFP itself represents only a constraint satisfaction problem, an objective function is often needed. Many real world optimization criteria such as revenue and cost optimization in transportation can be approximated by a piecewise linear function. In the problem of our particular interest (motivated by a large-scale industrial scheduling problem), a non-convex piecewise linear function maps the start time (or end) of each of the activities of the KRFP to an activity cost. The objective function is then a minimization of the sum of the values of these functions. The kernel resource feasibility problem with a piecewise linear objective function is abbreviated by **PL-KRFP**.

Hybridization form: probe backtracking

Like many LSCO's, the PL-KRFP can be easily divided into different sub-problems, and thus, it provides us with an interesting test-bed for **hybrid algorithms** that make use of good characteristics of different algorithm classes. A particularly suitable algorithm hybridization form for resource constrained scheduling is **probe backtracking** presented by El Sakkout and Wallace [1].

In probe backtracking, a constraint satisfaction and optimization problem is solved by a high-level backtracking algorithm that hybridizes another algorithm. All the problem constraints are classified into two sets: "easy" and "hard". The "easy"

constraints are those constraints that are easily solved by the hybridized algorithm, known as the **prober**. At any search node, the backtrack search applies the prober to a sub-problem consisting of the "easy" constraints and an objective function. The prober will generate a complete assignment that satisfies all the "easy" constraints, and is good (and, for some probers, **super-optimal**) with respect to the objective function. If the assignment also happens to satisfy all the "hard" constraints, a solution has been generated, and the search may terminate, or continue with a tighter cost bound if better solutions are sought as part of the branch and bound process. If the assignment has been found to violate one or more "hard" constraints, then a new "easy" constraint is identified and posted to the prober. This "easy" constraint is chosen to force the prober to generate a different assignment that reduces the violation of one of the violated "hard" constraints. The posting of this constraint moves the backtrack search to a new node, and the process repeats. On backtracking, the negation of this "easy" constraint is posted (the negation must also be an "easy" constraint).

In the case of PL-KRFP, a natural approach is to keep **temporal constraints** within the "easy" sub-problem addressed by the prober, while relaxing the **resource constraints** and handling them as "hard" constraints at the backtrack search level. If the solution to the sub-problem does not satisfy the relaxed resource constraints, additional temporal precedence constraints are added to the sub-problem in order to relieve violations.

The KRFP experiments presented in [1] show that even with a simple convex piecewise linear objective function, probe backtrack search with an LP prober clearly outperforms pure CP and pure MIP techniques.

Probe backtracking with linear and mixed-integer programming

For **non-convex** PL-KRFP, Ajili and El Sakkout [2] introduced two mixed-integer programming (**MIP**) and one linear programming (**LP**) based probe backtracking algorithms. All of them take an advantage of the fact that the temporal constraints of KRFP correspond to the **simple temporal problem** that can be solved efficiently by LP, guaranteeing the discreteness of temporal variables at the same time.

In the two MIP approached the temporal sub-problems with non-convex piecewise linear objective functions are solved by a MIP prober with a sophisticated use of the SOS2 constraint. In the LP approach, which appears to be more efficient than the first two, the prober applies an LP solver to a linear relaxation of the piecewise linear objective function. Possible discrepancies between the linearly relaxed objective and the allowed values of the true piecewise linear function are resolved by a small search procedure which branches over the temporal variables' domains, until this discrepancy has been eliminated.

An important advantage of these probers is that the returned assignments not only satisfy all the "easy" temporal constraints but also are super-optimal in terms of the piecewise linear objective function. This leads to algorithm completeness in a general sense, i.e. these MIP and LP probe backtracking algorithms maintain two levels of completeness for constraint satisfaction and optimization problems, namely

- sat-completeness**, where the algorithm returns a solution, or proves no solution exists, in a finite number of steps;
- and
- opt-completeness**, where the algorithm returns an **optimal** solution, or proves no solution exists, in a finite number of steps.

Probe backtracking with local search

However, one limitation of the approaches presented in [2] is their scalability, since they may produce relatively large backtrack search trees. This may easily lead to a smaller proportion of solved problems, if the global timeout limit is tight. One way to tackle this is to **sacrifice opt-completeness** by replacing the LP or MIP prober with a local search algorithm (**LS**).

A local search based probe backtracking technique, **local probing**, is presented in [3]. Here, an LS algorithm is run at each node of the probe backtracking tree in order to find, to the temporal sub-problem, an assignment that is as good as possible but not super-optimal w.r.t. the piecewise linear objective function. Furthermore, the neighbourhood operator used in LS always finds an assignment that satisfies all the temporal constraints if there exists such an assignment in the search space of the sub-problem. This is necessary in order to maintain sat-completeness of local probing.

An advantage of using LS prober instead of LP/MIP prober is that the objective function is not restricted to be linear or piecewise linear. This allows us, for example, to use an extended objective function including a component for penalizing possible resource constraint violations. Some experimental comparisons of local probing, LP probing and “pure” local search are presented in [3], indicating that although LP probing generally achieves solutions of better optimization quality, the local search characteristics of local probing provides us with significantly more instances with a solution found. Naturally, within these tests, “pure” local search suffered from its inability to detect infeasibility, unlike the two sat-complete hybrid approaches.

*

In conclusion, we have shown with large-scale non-convex piecewise linear scheduling instances, that different probe backtracking hybrids can marry the strengths of constraint satisfaction techniques, including good satisfaction characteristics and proofs of problem infeasibility, with the superior optimization characteristics of both linear/mixed-integer programming and local search. More references to hybrid algorithm research in IC-Parc can be found in the ECLiPSe home page (see <http://www.icparc.ic.ac.uk/eclipse/reports>).

[1] H. El Sakkout and M. Wallace. Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints*, 5(4):359-388, 2000.

[2] F. Ajili and H. El Sakkout. A Probe-based Algorithm for Piecewise Linear Optimization in Scheduling. *Annals of Operations Research*, to appear.

[3] O. Kamarainen and H. El Sakkout. Local Probing Applied to Scheduling. In *Proc. of CP'02*, pages 155-171, Springer 2002.