Global optimization in LocalSolver 9.5

Nikolas Stott
LocalSolver, 36 avenue Hoche, Paris, France
e-mail : nstott@localsolver.com

1 Contexte

LocalSolver is a mathematical optimization solver and modeling tool with a "model & run" approach. Problems can be modeled using the classical numerical variable types from integer linear programming (booleans, integers and floats) as well as collection-valued variables (lists and sets). The latter are useful for modeling routing, packing and scheduling problems.

The computation of solutions relies on its historical core of heuristic methods, using local search techniques, and on a global optimization module to handle MINLPs.

LocalSolver natively accepts problems that have several objectives, that are treated in lexicographical order. In order to show the quality of the solutions that have been produced, LocalSolver computes lower bounds (resp. upper bounds) for each objective that has been minimized (resp. maximized). The strategy of LocalSolver is to produce solutions of good quality very quickly on the whole spectrum of problems that can be modeled. These solutions are then iteratively improved. The same philosophy is adopted for the computation of lower bounds : get good bounds quickly and improve them afterwards.

This talk focuses on the set of techniques that are used within LocalSolver to produce bounds and shows their application on a set of academic benchmarks and industrial cases.

2 Lower bound computation techniques

2.1 Propagation and inference during preprocessing

An optimization problem is represented in LocalSolver as a directed acyclic graph in which the roots are the objectives and constraints and the leaves are the decision variables. The values that any node of the graph can take are stored as an interval. The bounds on each node can either be deduced from their children's bounds or infered from their parents'. This interval arithmetic allows LocalSolver to get lower bounds on each objective as soon as the model has been preprocessed.

2.2 Linear and nonlinear mixt integer problems

In the case where the problem being modeled has a single objective, the global optimization module uses linear or convex relaxations and a *branch-and-*

reduce mechanism to compute lower bounds. If linear relaxations are used, the subproblem is solved by a state-of-the-art linear programming solver, otherwise a dedicated nonlinear solver is used.

Thus, LocalSolver provides optimal solutions to 40% of the problems in the MINLPLib and feasible solutions for 70% of them within 60s.

When the model has several objectives, bounds are computed on each objective separately. Each objective must be solved to global optimality by constraining the previous ones to be equal to their optimal value before starting to solve the next one. We show the benefits of this approach on several industrial cases, on which the quick resolution to optimality on the first objectives improves significantly the solution of the last objectives.

2.3 Collection-based problems

The MINLP subsolver is also used to compute solutions and bounds when the decision variables are sets, by exact linearization of the set variables and constraints.

On *bin-packing* problems, we use the extended formulation by Kantorovich, whose LP relaxation already give a great bound, often differing only by 1 from the optimal solution.

Routing problems are modeled with lists in LocalSolver. Using only propagation and inference based arguments results in weak bounds, so these problems are preprocessed with a dedicated technique : the Held-Karp algorithm for symetric and asymetric TSPs. Coupled with the heuristic methods in LocalSolver, we quickly get solutions with a proven gap of less than 3% on instances of the TSPLib that have less than 1000 cities.