Finding (Good) Feasible Solutions for Large Resource-Allocation Problems

Cynthia Phillips*, Robert Carr, Jean-Paul Watson
Sandia National Laboratories

Summary

We give the first algorithm guaranteed to find a feasible solution to a large class of resource-allocation problem represented as a set of linear and integrality constraints. We improved the quality of the solutions returned by of one of the best general heuristics by 30%.

Scheduling a telescope, placing computing jobs on the processors of a parallel computer, locating guards or cameras in a facility, managing data collection in a wireless sensor network: all are examples of problems where one wishes to optimally allocate limited resources to accomplish a task. We have developed new algorithms for finding good approximate solutions for a broad class of problems that include all of the above as well as drug design, protein comparison, logistics, and environmental clean-up and monitoring, to name a few.

We can formulate all of these example problems as *mixed-integer programs (MIPs)*: maximizing or minimizing a linear objective function subject to linear and integrality constraints. Binary (0/1) variables represent yes/no decisions. It is theoretically intractable to even find a feasible solution, one that satisfies all the constraints. But in practice MIP solvers like our PICO code or other commercial or free codes, can find provably (near) optimal solutions for many problems using intelligent search.

In the MIP search process we repeatedly partition the space of possible solutions into subproblems. For each subproblem we compute a bound, a value guaranteed to be better than any feasible solution for the subproblem. If this bound is worse than the best solution found so far, we can ignore (prune) the subproblem. We compute a bound by relaxing the integrality constraints, which leaves an efficiently solvable *linear program* (LP). But there is no general way to find a good feasible MIP solution. Finding a good feasible solution early in the search enables pruning, so it is critical for good overall performance. Furthermore, a good solver should return some feasible solution if the user stops the search early.

The solution to the LP relaxation, x^*, provides global structural information. There are theoretically good approximation algorithms for specific problems based upon transforming x^* into an integer-feasible solution. We focused on LP-based techniques for finding feasible solutions for *general* MIPs. We improved *feasibility pump*, one of the best known LP-based heuristics for finding a feasible solution to a general MIP. We also designed the *fractional decomposition tree (FDT)*

---

* (505) 845-7296, caphill@sandia.gov. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

algorithm. This algorithm provably finds a feasible solution for a large class of MIPs. Fischetti et al. introduced the *feasibility pump (FP)* in 2005. FP rounds the LP solution to create an (infeasible) integer solution x. It then finds an LP-feasible solution x' that minimizes the distance to x and iterates (with x' as the starting point) until it finds an integral x' or gives up.

We modified the basic FP algorithm to achieve approximately a 30% improvement in heuristic solution quality on standard benchmark problems. The new version, now available in PICO, runs in approximately the same time as the original on a moderately parallel machine. The modifications were 1) randomize the rounding for fractional binary variables near 0.5, 2) round multiple times, 3) apply FP to subproblems; Fischetti et. al. only applied it to the full problem, and 4) apply FP to perturbations of good feasible solutions to find other nearby solutions.

Many resource allocation problems (e.g. scheduling without deadlines) always have a feasible solution. However, given only a matrix representation, it is difficult to infer this underlying combinatorial structure. Even the best commercial solvers can fail to find a feasible solution for large instances.

The FDT algorithm is guaranteed to efficiently find a feasible solution for a broad class of MIPs. Roughly, FDT works for non-negative objective functions if for all constraint bounds there is always an integer-feasible solution whenever there is an LP-feasible solution.

FDT computes a *convex decomposition*. In Figure 1 the black box represents feasible variable assignments for the LP relaxation. Any integer point inside it is feasible for the MIP. The blue object is the integer polytope, the smallest convex region that contains all integer solutions. The red box is the LP region scaled by a factor ρ. This scaling brings the LP optimal point inside the integer polytope. It is now an "average" or convex combination of feasible integer points (the 3 blue points in Figure 1). One of these points is guaranteed to have a value at most ρ times optimal. The FDT algorithm expresses the LP relaxation solution x* as the average of two partial solutions, each closer to integral than x*. It repeats this on these two partial solutions and so on. When the number of partial solutions exceeds a threshold, FDT selects a subset of them that can still be resolved to a full decomposition and is small enough to guarantee a theoretically good running time. In practice we expect FDT will be slower than feasibility pump, but it should succeed on a much larger class of problems.

We designed a parallel heuristic management system to coordinate application of these methods with the heuristics already in PICO.

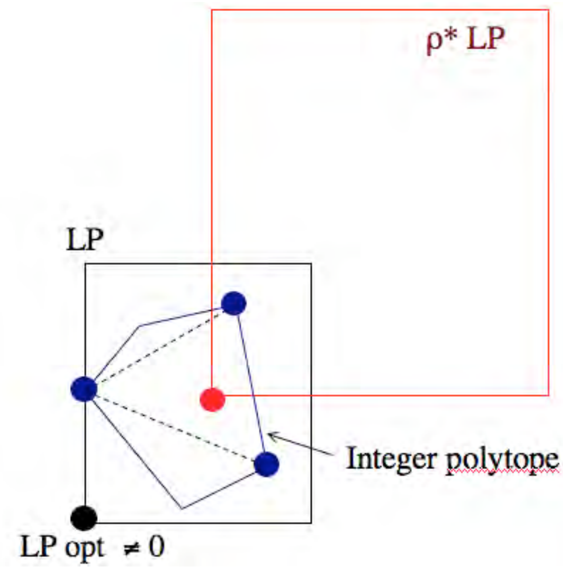

Figure 1: Representing a scaled LP optimal point as an "average" of integer-feasible points.

For further information on this subject contact:
Dr. Anil Deane
Applied Mathematics Research Program
Office of Advanced Scientific Computing Research
Phone: 301-903-1465
deane@mics.doe.gov