A new polynomial algorithm for nested resource allocation, speed optimization, and other related problems.

Thibaut VIDAL
LIDS, Massachusetts Institute of Technology
vidalt@mit.edu

Patrick JAILLET
Department of Electrical Engineering and Computer Science, LIDS & ORC,
Massachusetts Institute of Technology
jaillet@mit.edu

Nelson MACULAN
COPPE, Systems Engineering and Computer Science, Federal University of Rio de Janeiro
maculan@cos.ufrj.br

We propose an exact polynomial algorithm for a nested resource allocation problem with convex costs and constraints on partial sums of resource consumptions, in the presence of either continuous or integer variables. No assumption of strict convexity or differentiability is needed. This resource allocation problem, albeit extremely simple to formulate, appears prominently in a variety of applications related to production and resource planning, lot sizing, speed optimization in vehicle routing, and telecommunications. The fastest current method [D.S. Hochbaum. Lower and upper bounds for the allocation problem and other nonlinear optimization problems. *Mathematics of Operations Research*, 19(2):390-409, 1994] is based on a greedy algorithm and scaling concepts.

In the proposed new hierarchical decomposition method, solutions from sub-problems are used to convert constraints on sums of resources into bounds for separate variables at higher levels. The resulting time complexity for the integer problem is $O(n \log m \log (B/n))$, and the complexity of obtaining an ε-approximate solution for the continuous case is $O(n \log m \log (B/\varepsilon))$, n being the number of variables, m the number of nested constraints, ε a desired precision, and B the total resource. This matches the best-known complexity of Hochbaum (1994) when m=n, and improves it when $m = o(n)$.

Extensive experimental analyses are conducted with four recent algorithms on various continuous problems issued from theory and practice, demonstrating the high performance of

the proposed approach. All problems with up to one million variables are solved in less than one minute on a modern computer, and small-size problems of less than 100 variables are solved in a few milliseconds. This method can also significantly contribute to solve a variety of combinatorial optimization problems involving speed optimization sub-problems.