

The Primal-Dual Schema for Approximation Algorithms: Where Does It Stand, and Where Can It Go?

Vijay Vazirani

Georgia Institute of Technology (USA)

December 11, 2000

Summary by Claire Kenyon

Introduction

NP-hard problems cannot be solved exactly *and* efficiently at the same time. Can they be approximated in polynomial time? When doing so, we want a guarantee: for every instance, the solution must be within some factor of the optimal solution. Such questions are discussed systematically in Vijay Vazirani's book [6] on which the present lecture is based.

Linear programming duality theory provides many efficient algorithms with a good *approximation* factor. Designing *exact* algorithms is a main topic of the paper by Grötschel, Lovász, and Schrijver in 1981; see [3]. As we shall see, the primal-dual scheme provides the broad outline of an algorithm; working out the details for each individual problem then often provides a specific approximate solution with good complexity characteristics.

1. The Vertex Cover Problem

Given a graph, a subset of its vertices is a vertex cover if and only if every edge has at least one vertex in the subset. Each vertex has a cost—the cover having cost equal to the sum of the costs of its vertices—and we wish to obtain the cover of minimum cost. This problem is NP-hard (as proved by Karp in 1971, see [5]). We need to compare the cost of an approximate solution constructed by an algorithm to the cost of the optimal solution (OPT), but we do not know the cost of OPT; so we need a good lower bound on the cost of OPT. This is a key first step in the design of approximation algorithms.

1.1. Linear programming approximation. To the end of obtaining bounds on OPT for vertex cover, we start with an integer programming formulation of it. There is one variable x_v for each vertex v , and it is equal to 0 or 1; there is one constraint for each edge $\{u, v\}$, i.e., $x_u + x_v \geq 1$, which expresses that the sum of its two endpoint variables is at least 1; it is then required to minimize a linear combination of vertex variables times vertex costs, i.e. $\sum_v \text{cost}(v) \times x_v$.

We then do a relaxation of the problem by allowing the variables to be real numbers between 0 and 1 (instead of being integers). Each feasible solution provides a *fractional* vertex cover whose cost is necessarily a lower bound to OPT. We know since the works of Khachian and Karmarkar around 1980 that linear programming is polynomial-time solvable, both theoretically and effectively. The best fractional solution is thus polynomial-time computable, which gives us our lower bound. The relaxation algorithm is then as follows:

Linear Programming Algorithm. First find the optimal fractional solution, then put in the cover all vertices v such that $x_v \geq 1/2$. It is easy to see that this is a vertex cover, and

the cost is at most 2 times the lower bound, hence at most 2 times OPT. This algorithm has the defect of requiring to solve a linear program, a polynomial-time but expensive step.

1.2. A combinatorial algorithm. The principle of a combinatorial algorithm that has an approximation factor of 2 is as follows. Initially the cover C is empty. While C is not a vertex cover, pick an uncovered edge $\{u, v\}$, look at the smaller of the two current costs of u and v , subtract this smaller current cost from the costs of u and of v , put the corresponding vertex in C , and charge its cost to the edge. What we charge to the edges turns out (by induction) to be a lower bound on OPT. The cost of the cover is obviously at most twice the amount charged to the edge. Hence this technique gives rise to a combinatorial algorithm with an approximation factor of 2; the outcome is in fact a very fast linear-time algorithm.

This alternative algorithm is actually related to the LP-based algorithm seen previously. There is currently no approximation algorithm known which beats this factor of 2.

2. LP Relaxation and Dual LP

An original linear programming (LP) problem (the “primal”) always admits a “dual” formulation.

Primal linear program (LP). Determine $\min \sum_v \text{cost}(v) \times x_v$ subject to $\forall e \ x_u + x_v \geq 1$ and $\forall v \ x_v \geq 0$.

One can prove an upper bound on the OPT solution to the primal LP by exhibiting a particular solution (x_v) which satisfies all the constraints. One can prove a lower bound by exhibiting a particular linear combination of the constraints which equals the objective function. This corresponds to a dual LP solution.

Dual linear program. Determine $\max \sum_e y_e$ subject to $\forall v \ \sum_{e|v \in e} y_e \leq \text{cost}(v)$ and $\forall e \ y_e \geq 0$.

Equality of the optimal solutions of the primal and dual programs constitutes the *strong duality theorem*. The idea of a primal-dual algorithm is precisely to use a feasible solution of the dual LP as a *lower* bound on OPT. (Note that duality exchanges ‘min’ and ‘max’.)

How to design the primal-dual algorithm? We need the complementary slackness theorem, which says that if x is a feasible solution to the primal LP and y a feasible solution to the dual LP, then both are optimal if and only if for every v either $x_v = 0$ or $\sum_{e|v \in e} y_e = \text{cost}(v)$, and for every edge e either $y_e = 0$ or $x_u + x_v = 1$. Thus if (x, y) are not both optimal, we can find a slack and decrease the corresponding x_v or increase the corresponding y_e . To design an *approximation* algorithm, we change the equality relative to $\text{cost}(v)$ into an inequality.

Primal-dual algorithm for vertex cover. Initially x and y are set to 0. Let C be the set of “tight” vertices. While C is not a cover, do: pick an uncovered edge e , pick y_e and raise it until one of its two endpoints is tight. Iteratively improve the primal and dual solutions until a primal feasible solution is obtained; compare the primal and dual solutions to establish the approximation guarantee.

The set cover problem can be solved in the same fashion. In this problem, one has a set U of elements and a collection of subsets U_I , each with a positive cost, and one wishes to construct a minimum collection of subsets whose union is U . (Exercise: Let the frequency of element e be the number of subsets containing e , and let f be the maximal frequency of an element. Design a primal-dual approximation algorithm with an approximation factor of f .) By design, the best approximation factor we can get by these methods is the integrality gap, i.e., the ratio between the OPT solution to the integer linear program and the OPT solution to the relaxed linear program.

History. This paradigm started in 1955 (Kuhn) in the context of weighted bipartite matching. The primal-dual terminology is due to Dantzig, Ford, and Fulkerson in 1956. It was used to design *exact* algorithms for many polynomial-time algorithms much before linear programming was recognized to be polynomial-time solvable. Examples of this technique include matching, network flow, shortest paths, minimum spanning trees, branchings, and so on.

These exact primal-dual algorithms all use the fact that the polyhedron defined by the LP has integral vertices, and so the LP has integral optimal solutions. It is the relaxation of the complementary slackness solutions that essentially leads to approximation algorithms.

In 1981 Bar-Yehuda and Even [2] gave an approximation algorithm with a factor of 2 for vertex cover. In retrospect, their work can be reframed in the setting of primal-dual algorithms so that it can be regarded as the first primal-dual approximation algorithm.

3. Other Problems

Many other problems can be solved approximately using the primal-dual approach. We give a short list below and refer to the book [6] for details.

Steiner tree problem. Given a graph and a set of red vertices in the graph, find a tree which connects all the red vertices (possibly using the other graph vertices in the tree) and has minimal total cost. Gauß also had a version on the plane (given a set of vertices in the plane, connect them into a tree, possibly branching out at other points in the plane).

Steiner network problem. Design a network with a prescribed number of edge-disjoint paths between pairs of vertices. There are numerous applications of this problem in networks.

Steiner forest problem. The connectivity requirement is 0 or 1 between pairs of vertices. In 1991 factor-of-2 algorithms were designed by Agrawal, Klein, and Ravi [1] on the one hand, Goemans, Williamson on the other hand. These authors use the idea of *simultaneously* raising the violated minimal constraints. In 1992 Williamson, Goemans, Vazirani, and Mihail [7] found a $2k$ approximation algorithm for the extended Steiner network problem when the maximum connectivity requirement is k ; their algorithm has been implemented at Bellcore.

Facility location problem. What is given is a set of locations for installing proxy servers and a set of clients; the goal is to minimize the sum of server installation cost plus the sum of client's connection costs. For this problem, in the late 1990s, several primal-dual approximation algorithms using LP rounding were designed; they are nice but not so practical. Recently Jain and Vazirani [4] got an approximation algorithm with a factor of 3 based on a practical combinatorial solution, which stems from the primal-dual scheme.

The k -median problem. This problem is like the facility location problem, except that facilities are free, one is constrained to open at most k facilities; what is required is to minimize the connection cost. This has applications to data mining *inter alia*. In 1998 there was an $O(1)$ -approximation primal-dual algorithm based on LP-rounding, but that again had the disadvantage of requiring to solve a linear program. In 1999 Jain and Vazirani designed a combinatorial algorithm that is more complicated and relies on randomized rounding. This last algorithm can then be derandomized using the method of conditional expectations.

The techniques discussed in this talk are very robust in the sense that once you solve one problem, you can get solutions to many closely related problems as well.

4. Open Problems

Our approximation algorithms always deal with dual variables in a greedy fashion, whereas exact primal-dual algorithms are much more sophisticated: there is a long way to go to bring the two approaches closer!

Some of the main open problems are: get a factor better than 2 for vertex cover, and better than $3/2$ for the traveling salesman path; get a factor of 2 for the Steiner network; design a bidirected cut relaxation for Steiner trees.

Bibliography

- [1] Agrawal (Ajit), Klein (Philip), and Ravi (R.). – When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, vol. 24, n° 3, 1995, pp. 440–456.
- [2] Bar-Yehuda (R.) and Even (S.). – A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, vol. 2, n° 2, 1981, pp. 198–203.
- [3] Grötschel (M.), Lovász (L.), and Schrijver (A.). – The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, vol. 1, n° 2, 1981, pp. 169–197.
- [4] Jain (Kamal) and Vazirani (Vijay V.). – Approximation algorithms for metric facility location and k -median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, vol. 48, n° 2, 2001, pp. 274–296.
- [5] Karp (Richard M.). – Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pp. 85–103. – Plenum, New York, 1972.
- [6] Vazirani (Vijay V.). – *Approximation algorithms*. – Springer-Verlag, Berlin, 2001, xx+378p.
- [7] Williamson (David P.), Goemans (Michel X.), Mihail (Milena), and Vazirani (Vijay V.). – A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, vol. 15, n° 3, 1995, pp. 435–454.