Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp 357-363

On Estimation Algorithms versus Approximation Algorithms

Uriel Feige[1*]
Weizmann Institute
Rehovot, Israel
uriel.feige@weizmann.ac.il

ABSTRACT.
In a combinatorial optimization problem, when given an input instance, one seeks a feasible solution that optimizes the value of the objective function. Many combinatorial optimization problems are NP-hard. A way of coping with NP-hardness is by considering approximation algorithms. These algorithms run in polynomial time, and their performance is measured by their approximation ratio: the worst case ratio between the value of the solution produced and the value of the (unknown) optimal solution.
In some cases the design of approximation algorithms includes a nonconstructive component. As a result, the algorithms become estimation algorithms rather than approximation algorithms: they allow one to estimate the value of the optimal solution, without actually producing a solution whose value is close to optimal.
We shall present a few such examples, and discuss some open questions.

1 Introduction

In a combinatorial optimization problem, when given an input instance, one seeks a feasible solution that maximizes (or minimizes) the value of the objective function. For example, in the Travelling Salesperson (TSP) problem, given an input graph with edge lengths, one is to find a tour (Hamiltonian cycle) of minimum length. Combinatorial optimization problems are very common in practice, and are also of great theoretical interest. Many combinatorial optimization problems are NP-hard (informally meaning that we know of no polynomial time algorithm that solves every instance optimally). A way of coping with NP-hardness is by considering *approximation algorithms*. These algorithms run in polynomial time (or sometimes, random polynomial time), but are not guaranteed to produce optimal solutions. Their performance is measured by their *approximation ratio*. For a maximization problem, an approximation algorithm is said to have approximation ratio $0 \leq \rho \leq 1$ if on every instance, the value of the solution output by the algorithm is at least ρ times the value of the optimal solution. (For minimization problems, $\rho \geq 1$, and the value of the solution output by the algorithm is at most ρ times the optimal.) It is often the case that the approximation ratio of an algorithm is not a fixed constant that holds for all input sizes n, but rather it deteriorates as the input size grows. In this case, rather than just saying that the approximation ratio is 0 (for maximization problems) or unbounded (for minimization problems), we measure the rate at which the the approximation ratio deteriorates (as a function of n). For example, the greedy algorithm for set cover has approximation ratio $\ln n$. The approximation ratio

[*] Supported in part by The Israel Science Foundation (grant No. 873/08)

of an optimization problem is the best approximation ratio achieved by any approximation algorithm for the problem. For more details see for example [24, 26, 6, 37].

We say that a combinatorial optimization problem has a *threshold* at ρ if there is a polynomial time (randomized) algorithm for it with approximation ratio ρ, and it is NP-hard to approximate it within a ratio better than ρ. (Here we ignore low order terms in the approximation ratio.) Problems that have approximation ratios arbitrarily close to 1 (a so called Polynomial Time Approximation Scheme, PTAS) have a threshold at 1. Perhaps surprisingly, many other problems (such as k-center, set cover, max coverage, max 3SAT) also have approximation thresholds, though the locations of the thresholds may differ among problems.

Needless to say, for many problems (such as metric TSP, max SAT, min bisection and dense k-subgraph) we do not know if they have a threshold or not. Problems with no known threshold are the ones relevant to the discussion that follows.

At this point it will be convenient to distinguish between notions that we shall call here *approximation* algorithms and *estimation* algorithms. For the approximation problem, one is required to find a feasible solution whose value is close to that of the value of the optimal solution. For estimation algorithms, one is required to estimate the value of the optimal solution, without necessarily outputting a solution that meets this estimate. This is potentially an easier task. It turns out that hardness of approximation results are essentially always also hardness of estimation results, within the same ratio. That is, our techniques for establishing hardness of approximation do not distinguish between approximation and estimation. On the algorithmic side, most positive results apply equally well to estimation and approximation. However, there are some exceptions where at the moment the known estimation ratios are better than the known approximation ratios.

2 Some research directions

The distinction between estimation algorithms and approximation algorithms offers interesting research directions.

Prove new estimation ratios. For some problems there are large gaps between the known approximation ratios and the known hardness of approximation results. For such problems, try to establish estimation ratios that are better than the known approximation ratios.

Close the gaps between estimation and approximation ratios. For some problems there are large gaps between the known approximation ratios and the known estimation ratios. For such problems, try to improve the approximation ratio (hopefully, replacing the nonconstructive arguments that lead to the estimation ratios by constructive arguments that lead to the same approximation ratio).

Relating between open questions. Introduce complexity classes that capture current gaps between estimation and approximation (similar in spirit to the work of [32]). That is, we would like to be able to show that if this gap is closed for one problem, this automatically implies that the gap will be closed for other problems.

Relating to external open questions. At the moment we do not have convincing evidence that there should be a gap between approximation ratios and estimation ratios. For many optimization problems these ratios provably match (when there is a known approximation threshold, such as for max-3SAT or min set cover), in others they currently match (such as for min vertex cover or sparsest cut), and in the remaining cases the theory of NP-completeness does not appear to apply, because it deals with decision problems rather than search problems. Try to establish connections between previously defined concepts (such as PPAD-completeness) and gaps between approximation and estimation. (To appreciate the subtleties involved consider the following example. Finding a locally maximal cut is PLS-complete, but the known approximation ratios for max-cut [25] are better than those that local search gives. Hence PLS-completeness by itself is not an obstacle to bridging the gap between estimation and approximation.)

Development of techniques. There are some proof techniques that originally were nonconstructive, and algorithmic versions of them (or of special cases) were discovered only later. See for example [7] for the local lemma and [4] for the regularity lemma. Design algorithmic versions of nonconstructive arguments, regardless of any immediate applicability to combinatorial optimization.

Random instances. Nonconstructive arguments often show that random instances (such as random 3CNF formulas) are likely to either have or not have solutions (depending on the density of the underlying instance). Find algorithmic versions of these results. These type of questions have indirect connections to approximation algorithms, and may well require similar sets of techniques (see [15] for example).

3 Examples

Below we list some examples of current gaps between approximation ratios and estimation ratios (or conjectured estimation ratios).

Max-min allocation.

In max-min allocation problems, there is a threshold t, a set of m items, a set of n players, and nonnegative valuations v_{ij} that for every player i and item j specify the value of item j to player i. The goal is to allocate items to the players in a way that every player gets total value (sum of his values for the items allocated to him) at least t. This problem is NP-hard. A linear program relaxation of this problem provides an upper bound on the maximum possible value of t. It is known that the gap between this upper bound and true optimum may be $\Omega(\sqrt{n})$. However, in an interesting special case, the *restricted assignment* version, there is a nonconstructive proof (in fact, two different nonconstructive proofs by now, [18] using the local lemma, [5] using local search) that the gap is at most constant. Hence the value of the linear program provides a constant factor estimation for the restricted assignment version of the max-min allocation problem. No constant factor approximation ratio is known for this problem.

Metric TSP.

The Held-Karp conjecture states that the value of a certain linear program provides a $4/3$ estimation for metric TSP in undirected graphs. If true, this conjecture provides a $4/3$ estimation ratio for metric TSP, which is better than the known approximation ratio of $3/2$.

For undirected graphs it is known that the integrality gap of the LP is no better than $4/3$ and no worse than $3/2$. For directed graphs, the integrality gap is known to be no better than 2, and no sublogarithmic approximation ratios are known.

Edge colorings in multigraphs.

There is a famous theorem by Vizing that states (and gives an algorithm) that in every simple graph there is a legal edge coloring with one more color than the maximum degree in the graph. This gives an approximation of the edge chromatic number within additive 1. It was conjectured (e.g., by Seymour) that a similar result can be extended to multigraphs, using a linear programming relaxation. If true this would provide an estimation algorithm for the edge chromatic number within an additive error of 1. There are nonconstructive proofs (using the local lemma) that give a $1 + \epsilon$ multiplicative estimation when the edge chromatic number of multigraphs is sufficiently large [29].

Discrepency.

Many discrepancy problems can be viewed as coloring problems on hypergraphs. The goal is to color the vertices such that every hyperedge remains nearly balanced (has roughly the same number of vertices of each color). Techniques used in the proofs that low discrepancy colorings exist are sometimes constructive (such as the Beck-Fiala theorem that iteratively uses basic feasible solutions of linear programs), and sometimes nonconstructive (such as the first use of the Lovasz local lemma, or Spencer's proof that "six standard deviations suffice" that uses the *pigeon hole principle* in a nonconstructive way). The reader is referred to [9, 31] were references to these and other results can be found. In general, it is often the case that statements involving discrepancy involve nonconstructive proofs (see also [2, 16]).

It would be desirable to replace some of the nonconstructive proofs in discrepancy theory by algorithmic proofs (as was done by Beck in the context of the local lemma). Perhaps more ambitiously, improve some of the known discrepancy bounds. (For example, it is conjectured that the Beck-Fiala theorem can be improved when the degrees are large.)

Graph bandwidth.

A linear arrangement of a graph is a numbering of its n vertices from 1 to n. The bandwidth of the linear arrangement is the maximum difference between numberings of endpoints of an edge. The *bandwidth* of a graph is the bandwidth of its minimum bandwidth linear arrangement. The *local density* of a graph is a natural lower bound on the bandwidth. It is known that the gap between bandwidth and local density can be $\Omega(\log n)$, and there is an algorithm that finds a linear arrangement of bandwidth $O(\log^{3.5} n)$ times the local density [14]. It is reasonable to conjecture that the maximum ratio between bandwidth and local density is $O(\log n)$. If true, then local density provides an $O(\log n)$ estimation ratio for the bandwidth. The best approximation ratio known for the bandwidth is currently $O(\log^3 n)$ [13].

Random 3CNF.

Work on refuting dense random 3CNF formulas offers a lot of interplay between existential and algorithmic arguments. For example, it is shown in [20] that formulas of density above $n^{0.4}$ are likely to have polynomial size witnesses for nonsatisfiability. There is no known efficient algorithm for finding these witnesses. Or another example, the notion of *even covers*, originally studied in coding theory, is used in [20, 17] as part of refutation algorithms and witnesses. Further progress is hampered because we are missing an existential result – we do not know how to prove that small even covers must exist at densities below

$\sqrt{n}$, and because we are missing an algorithmic result – we do not know how to find small even covers when they do exist.

4 Conclusions

The list of references is not based on a careful study of all related references. Hence it may miss some important references, and include some papers whose relevance to this manuscript is questionable. A short overview of the topics addressed by some of the references is provided.

A well known nonconstructive proof technique is the Lovasz local lemma (see for example [3]). It had been used in the design of estimation algorithms [30, 22, 19, 18]. In some cases, algorithmic versions of the local lemma are known [7, 12].

The use of linear programming relaxations is common in approximation algorithms. Sometimes general principles (such as the existence of basic feasible solutions) can be used in order to show show the existence of high quality integer solutions (as in [8]). In some cases the underlying linear programs are of exponential size (as in [2, 16]). These lead naturally to estimation algorithms rather than approximation algorithms. Sometimes, the result inferred from the exponential LP may be obtained by a more direct efficient algorithm (see [23] for one such example), leading to approximation algorithms.

In the context of random instances of CNF formulas there are many nonconstructive arguments that lack a constructive counterpart. See examples of work in this area in [1, 11, 15, 17, 20, 21].

Local search is a common algorithmic tool that does not always lead to polynomial time algorithms [27, 28, 33, 35]. When used for optimization problems, it might result in estimation algorithms rather than approximation algorithms [5].

There are certain complexity classes that attempt to capture nonconstructive principles. See [32, 10] for example.

In the context of counting problems [36] there are many randomized approximation algorithms (such as [34]). In our terminology, we would view them as estimation algorithms rather than approximation algorithms, since they are only required to output an estimation for the number of solutions, rather than to list the solutions (which in typical situations would require exponential output size).

In conclusion, the distinction between approximation and estimation algorithms has been an explicit or implicit part of research for many years. The purpose of this manuscript is to bring this distinction and the research opportunity that it offers to the awareness of more researchers.

References

[1] Dimitris Achlioptas, Yuval Peres. The threshold for random k-SAT is $2k(\ln 2 - O(k))$. *STOC 2003: 223–231.*

[2] Gagan Aggarwal, Amos Fiat, Andrew V. Goldberg, Jason D. Hartline, Nicole Immorlica, Madhu Sudan. Derandomization of auctions. *STOC 2005: 619–625.*

[3] N. Alon, J. Spencer. *The probabilistic Method.* Wiley Interscience.

[4] Noga Alon, Richard A. Duke, Hanno Lefmann, Vojtech Rodl, Raphael Yuster. The Algorithmic Aspects of the Regularity Lemma. *J. Algorithms 16(1): 80–109 (1994).*

[5] A. Asadpour, U. Feige, A. Saberi. Santa Claus Meets Hypergraph Matchings. *Proceedings of APPROX-RANDOM 2008: 10–20.*

[6] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. *Complexity and Approximation*. Springer Verlag, 1999.

[7] J. Beck. An algorithmic approach to the Lovasz Local Lemma. *Random Structures and Algorithms, 2 (1991), pp, 343–365.*

[8] J. Beck and T. Fiala. "Integer-making" theorems. *Discrete Applied Mathematics, 3:1–8, 1981.*

[9] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, 2000

[10] Xi Chen, Xiaotie Deng. Settling the Complexity of Two-Player Nash Equilibrium. *FOCS 2006: 261–272.*

[11] Amin Coja-Oghlan, Andreas Goerdt, Andre Lanka: Strong Refutation Heuristics for Random k-SAT. *APPROX-RANDOM 2004: 310-321.*

[12] Artur Czumaj, Christian Scheideler. A new algorithm approach to the general Lovasz local lemma with applications to scheduling and satisfiability problems (extended abstract). *STOC 2000: 38–47.*

[13] John Dunagan, Santosh Vempala. On Euclidean Embeddings and Bandwidth Minimization. *RANDOM-APPROX 2001: 229-240.*

[14] Uriel Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences*, 60(3), 510–539, 2000.

[15] Uriel Feige. Relations between average case complexity and approximation complexity. *STOC 2002: 534-543.*

[16] Uriel Feige. You can leave your hat on (if you guess its color). *Technical report MCS04-03 of the Weizmann Institute, 2004.*

[17] Uriel Feige. Refuting smoothed 3CNF formulas. *Proc. of 48th FOCS*, 2007, 407–417.

[18] Uriel Feige. On Allocations that Maximize Fairness. *SODA 2008, 287–293.*

[19] Uriel Feige, Magnus M. Halldorsson, Guy Kortsarz, Aravind Srinivasan. Approximating the Domatic Number. *SIAM J. Comput. 32(1): 172-195 (2002).*

[20] Uriel Feige, Jeong Han Kim, Eran Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. *FOCS 2006: 497–508.*

[21] Uriel Feige, Eran Ofek. Easily Refutable Subformulas of Large Random 3CNF Formulas. *ICALP 2004: 519–530.*

[22] Uriel Feige, Christian Scheideler. Improved Bounds for Acyclic Job Shop Scheduling. *Combinatorica 22(3): 361–399 (2002).*

[23] Uriel Feige, Jan Vondrak. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. *FOCS 2006: 667–676.*

[24] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[25] Michel X. Goemans, David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM 42(6): 1115–1145 (1995).*

[26] Dorit Hochbaum (Ed.). *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.

[27] David S. Johnson, Christos H. Papadimitriou, Mihalis Yannakakis. How Easy is Local Search? *J. Comput. Syst. Sci. 37(1): 79–100 (1988).*

[28] Gil Kalai. Upper Bounds for the Diameter and Height of Graphs of Convex Polyhedra. *Discrete and Computational Geometry 8: 363–372 (1992).*

[29] Jeff Kahn. Asymptotics of the Chromatic Index for Multigraphs. *J. Comb. Theory, Ser. B 68(2): 233–254 (1996).*

[30] Frank Thomson Leighton, Bruce M. Maggs, Satish Rao. Packet Routing and Job-Shop Scheduling in O(Congestion + Dilation) Steps. *Combinatorica 14(2): 167-186 (1994).*

[31] J. Matousek. *Geometric Discrepancy*. Springer 1999.

[32] Christos H. Papadimitriou. On Graph-Theoretic Lemmata and Complexity Classes. *FOCS 1990: 794–801.*

[33] Christos H. Papadimitriou, Alejandro A. Schaeffer, Mihalis Yannakakis. On the Complexity of Local Search. *STOC 1990: 438–445.*

[34] Alistair Sinclair, Mark Jerrum. Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains. *Inf. Comput. 82(1): 93-133 (1989).*

[35] Mihalis Yannakakis. The Analysis of Local Search Problems and Their Heuristics. *STACS 1990: 298–311.*

[36] Leslie G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM J. Comput. 8(3): 410–421 (1979).*

[37] Vijay Vazirani. *Approximation Algorithms.* Springer 2001.