Domination Analysis of Combinatorial Optimization Problems

Gregory Gutin* Alek Vainshtein † Anders Yeo‡

Abstract

We use the notion of domination ratio introduced by Glover and Punnen in 1997 to present a new classification of combinatorial optimization (CO) problems: $\mathcal{DOM}$-easy and $\mathcal{DOM}$-hard problems. It follows from results proved already in the 1970's that `min TSP` (both symmetric and asymmetric versions) is $\mathcal{DOM}$-easy. We prove that several CO problems are $\mathcal{DOM}$-easy including `weighted max` k-`SAT` and `max cut`. We show that some other problems, such as `max clique` and `min vertex cover`, are $\mathcal{DOM}$-hard unless P=NP.

Keywords: Combinatorial Optimization; Domination analysis; Approximation Algorithms

1 Introduction

In this paper, we use the notion of domination ratio introduced by Glover and Punnen [6] to present a new classification of combinatorial optimization (CO) problems: $\mathcal{DOM}$-easy and $\mathcal{DOM}$-hard problems. It follows from the main results in [17, 18] that `min TSP` (both asymmetric and symmetric versions) is $\mathcal{DOM}$-easy. We prove that several CO problems are $\mathcal{DOM}$-easy including `weighted max` k-`SAT` and `max cut`. We show that some other problems, such as `max clique` and `min vertex cover`, are $\mathcal{DOM}$-hard unless P=NP.

This classification does not have certain drawbacks inherent in some well-known classifications of CO problems based on the best possible value of performance ratio of their approximation algorithms. For example, APX is the class of CO problems that admit polynomial time approximation algorithms with a constant performance ratio [2]. It is well known that while `max TSP` belongs to APX, `min TSP` does not. This is at odds with the simple fact that a 'good' approximation algorithm for `max TSP` can be easily transformed into an algorithm for `min TSP`. Thus, it seems that both `max` and `min TSP` should be in the same class of CO problems. The above asymmetry was viewed as a drawback of performance ratio based classifications already in the 1970's, see, e.g., [5, 12, 19].

Another example is `max independence set` and `min vertex cover`. It is well known and easy to prove that, in a graph G, every independent set complements a vertex cover and vise versa. Nevertheless, `min vertex cover` is considered to be 'easy' (and is in APX), while `max independence set` is viewed as a 'very hard' CO problem (not in APX). In our classification both `min TSP` and `max TSP` are easy, $\mathcal{DOM}$-easy, and both `max independence set` and `min vertex cover` are hard, $\mathcal{DOM}$-hard, unless P=NP.

*Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK, gutin@cs.rhul.ac.uk

†Department of Mathematics and Department of Computer Science, University of Haifa, Haifa, Israel, alek@cs.haifa.ac.il

‡Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK, anders@cs.rhul.ac.uk

Zemel [19] was the first to characterize measures of quality of approximate solutions (of binary integer programming problems) that satisfy a few basic and natural properties: the measure becomes smaller for better solutions, it equals 0 for optimal solutions and it is the same for corresponding solutions of equivalent instances. While the performance ratio and even the relative error (see [2]) do not satisfy the last property, the parameter $1-r$, where r is the domination ratio, does satisfy all the properties. Thus, our new classification does not have the drawbacks of some well-known classifications of CO problems based on performance ratio. (We do not dispute the fact that the performance ratio by itself is a very useful parameter for many CO heuristics, but no single parameter can cover a complex issue of quality of heuristics.) For results on another quality measure satisfying Zemel's properties, see [11] and references therein.

Notice that previous papers on domination analysis (see, e.g., [3, 4, 6, 7, 8, 9, 13, 15, 16, 17]) have dealt with evaluation and classification of algorithms. In the present paper we, for the first time, concentrate on domination properties of CO problems themselves.

2 Terminology and Notation

Let $\mathcal{P}$ be a CO problem and $\mathcal{A}$ be an algorithm for finding an approximate solution of $\mathcal{P}$. For an instance $\mathcal{I}$ of $\mathcal{P}$, the *domination number* $\mathrm{domn}(\mathcal{I},\mathcal{A})$ of $\mathcal{A}$ *on* $\mathcal{I}$ is the number of feasible solutions of $\mathcal{I}$ that are not better than the solution x found by $\mathcal{A}$ (including x itself). (In this paper we consider only CO problems that have the property that every instance has only a finite number of feasible solutions.) For example, consider an instance $\mathcal{T}$ of the Symmetric TSP (STSP) on 5 vertices. Suppose that the weights of tours in $\mathcal{T}$ are 4,5,5,6,7,9,9,11,11,12,14,14 (every instance of STSP on 5 vertices has 12 tours) and suppose that the greedy algorithm computes the tour T of weight 7. Then $\mathrm{domn}(\mathcal{T},\mathrm{greedy})=8$. In general, if $\mathrm{domn}(\mathcal{I},\mathcal{A})=\mathrm{sol}(\mathcal{I})$, where $\mathrm{sol}(\mathcal{I})$ is the number of feasible solution of $\mathcal{I}$, then $\mathcal{A}$ computes an optimal solution for $\mathcal{I}$.

The *domination number* $\mathrm{domn}(\mathcal{P},n,\mathcal{A})$ of an algorithm $\mathcal{A}$ for a CO problem $\mathcal{P}$ is the minimum $\mathrm{domn}(\mathcal{I},\mathcal{A})$ over all instances $\mathcal{I}$ of $\mathcal{P}$ of size n. It was proved in [8] (see also [9]) that $\mathrm{domn}(\mathrm{STSP},n,\mathrm{greedy})=1$. This means that for every $n\geq 2$ there is an instance of STSP for which the greedy algorithm finds the unique worst possible tour. Not every heuristic for STSP is that bad: already in 1973 Rublineckii [17] (see also [9]) proved that vertex insertion algorithms for STSP are of domination number at least $(n-2)!/2$. Thus, certain STSP (and Asymmetric TSP) heuristics always produce tours that are at least as good as $\Omega((n-2)!)$ other tours, see e.g. [7, 9, 15].

When the number of feasible solutions depends not only on the size of the instance of the CO problem at hand (for example, the number of independent sets of vertices in a graph G on n vertices depends on the structure of G), the domination ratio of an algorithm $\mathcal{A}$ is of interest: the *domination ratio* of $\mathcal{A}$ for a CO problem $\mathcal{P}$, $\mathrm{domr}(\mathcal{P},n,\mathcal{A})$, is the minimum of $\mathrm{domn}(\mathcal{I},\mathcal{A})/\mathrm{sol}(\mathcal{I})$, where as above $\mathrm{sol}(\mathcal{I})$ is the number of feasible solutions of $\mathcal{I}$, taken over all instances $\mathcal{I}$ of size n. Clearly, exact algorithms are of domination ratio 1.

An algorithm $\mathcal{A}$ for a CO problem $\mathcal{P}$ is $\mathcal{DOM}$-*good* if $\mathcal{A}$ is of polynomial time complexity and there exists a polynomial p in size of $\mathcal{P}$ such that the domination ratio of $\mathcal{A}$ is at least $1/p(n)$ for any size n of $\mathcal{P}$. A CO problem $\mathcal{P}$ is $\mathcal{DOM}$-*easy* if it admits a $\mathcal{DOM}$-good algorithm and $\mathcal{P}$ is $\mathcal{DOM}$-*hard* if there is no $\mathcal{DOM}$-good algorithm for $\mathcal{P}$. The above mentioned vertex insertion algorithms for STSP are of domination ratio $\Omega(1/(n-1))$ and thus STSP is $\mathcal{DOM}$-easy.

In this paper, we prove that several CO problems are $\mathcal{DOM}$-easy. Interestingly, `max SAT` is among them despite the fact that some well-known algorithms for `max SAT` are of very small domination ratio [4]. We also show that several other CO problems, such as `max clique` and `min vertex cover`, are $\mathcal{DOM}$-hard unless P=NP.

3 $\mathcal{DOM}$-hard Problems

Consider `max clique`, the problem to find the cardinality of a maximum clique in a graph. Håstad [10] proved that, provided P≠NP, `max clique` is not approximable within $n^{1/2-\epsilon}$ for any $\epsilon > 0$, where n is the number of vertices in a graph. We will use this remarkably strong result to prove the following:

Theorem 3.1 `max clique` *is $\mathcal{DOM}$-hard unless P=NP.*

Proof: Let G be a graph with n vertices, and let q be the number of vertices in a maximum clique Q of G. Let $\mathcal{A}$ be a polynomial time algorithm and let $\mathcal{A}$ find a clique M with m vertices in G.

Since the clique Q 'dominates' all 2^q of its subcliques and the clique M 'dominates' at most $\binom{n}{m}2^m$ cliques in G, the domination ratio r of $\mathcal{A}$ is at most $\binom{n}{m}2^m/2^q$. By the above non-approximability result of [10], we may assume that $mn^{0.4} \leq q$. Thus,

$$r \leq \frac{\binom{n}{m}2^m}{2^q} \leq \frac{(en/m)^m 2^m}{2^q} \leq \frac{(n/m)^m(2e)^m}{2^{mn^{0.4}}} = 2^s,$$

where $s = m(\log n - \log m + 1 + \log e - n^{0.4})$. Clearly, 2^s is smaller than $1/p(n)$ for any polynomial $p(n)$ when n is sufficiently large. $\square$

Obvious graph duality properties immediately imply the following:

Corollary 3.2 `max independence set` *and* `min vertex cover` *are $\mathcal{DOM}$-hard unless P=NP.*

Theorem 3.1 holds for some cases of the following much more general problem: `max induced subgraph with property` Π (see Problem GT25 in the compendium of [2]). The property Π must be hereditary, i.e., every induced subgraph of a graph with property Π has property Π, and non-trivial, i.e., it is satisfied for infinitely many graphs and false for infinitely many graphs. Lund and Yannakakis [14] proved that `max induced subgraph with property` Π is not approximable within n^ϵ for some $\epsilon > 0$ unless P=NP, if Π is hereditary, non-trivial and is false for some clique or independent set (e.g., planar, bipartite, triangle-free). This non-approximability result can be used as in the proof of Theorem 3.1.

4 $\mathcal{DOM}$-easy Problems

Recall that `min partition` is the following problem: given n numbers $V = \{a_1, a_2, \ldots, a_n\}$, find a bipartition of the set $\{1, 2, \ldots, n\}$ into sets X and Y such that $f(X,Y) = |\sum_{i\in X} a_i - \sum_{i\in Y} a_i|$ is minimum. For simplicity we assume, for `min partition` (and `max cut` considered below), that a bipartition (X,Y) is an ordered pair and $(\emptyset, V)$ and $(V, \emptyset)$ are feasible solutions. Thus, `min partition` and `max cut` have 2^n feasible solutions each.

Consider the following greedy-type algorithm $\mathcal{G}$ for `min partition`. $\mathcal{G}$ sorts the numbers such that $a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(n)}$, initiates $X = \{\pi(1)\},\ Y = \{\pi(2)\}$, and, for each $j \geq 3$, puts $\pi(j)$ into X if $\sum_{i\in X} a_i \leq \sum_{i\in Y} a_i$, and into Y, otherwise. It is easy to see that any solution (X,Y) produced by $\mathcal{G}$ satisfies $f(X,Y) \leq a_{\pi(1)}$.

Consider any solution (X',Y') of `min partition` for the input $\{a_1,a_2,\dots,a_n\} - \{a_{\pi(1)}\}$. If we add $a_{\pi(1)}$ to Y' if $\sum_{i\in X'} a_i \leq \sum_{i\in Y'} a_i$ and to X', otherwise, then we obtain a solution (X'',Y'') for the original problem with $f(X'',Y'') \geq f(X,Y)$. Thus, the domination number of $\mathcal{G}$ is at least 2^{n-1} and its domination ratio is at least 0.5. We have proved the following:

Proposition 4.1 `min partition` *is $\mathcal{DOM}$-easy.*

Recall that `max cut` is the following problem: given a weighted complete graph $G=(V,E,w)$ (weights w on the edges), find a bipartition (X,Y) of V such that the sum of weights of the edges with one end vertex in X and the other in Y, called the *weight of the cut* (X,Y), is maximum.

Theorem 4.2 `max cut` *is $\mathcal{DOM}$-easy.*

Proof: Let $G=(V,E,w)$ be a complete graph with $n=|V|$ vertices and let W be the sum of the weights of the edges in G. Clearly, the average weight of a cut of G is $\overline{W}=W/2$.

Consider the following well-known approximation algorithm $\mathcal{C}$ that always produces a cut of weight at least $\overline{W}$. The algorithm $\mathcal{C}$ considers the vertices of G in any fixed order $v_1,v_2,\dots,v_n$, initiates $X=\{v_1\},\ Y=\{v_2\}$, and appends v_i, $i\geq 3$, to X or Y depending on whether the sum of the weights of edges between v and Y or between v and X is larger. We will prove that $\mathcal{C}$ is $\mathcal{DOM}$-good. To show this, it suffices to prove that the number of cuts in G of weight at most $\overline{W}$ (we call them *bad cuts*) is at least a polynomial part of all cuts.

We call a cut (X,Y) of G a *k-cut* if $|X|=k$. We evaluate the fraction of bad cuts among k-cuts when $k\leq n/2-2\sqrt{n}$.

For a fixed edge uv of G among $\binom{n}{k}$ k-cuts there are $2\binom{n-2}{k-1}$ k-cuts that contain uv. Thus, the average weight of a k-cut is $\overline{W}_k = 2\binom{n-2}{k-1}W/\binom{n}{k}$. Let b_k be the number of bad k-cuts. Then, $(\binom{n}{k}-b_k)\overline{W}/\binom{n}{k} \leq \overline{W}_k$. Hence,

$$b_k \geq \binom{n}{k} - 4\binom{n-2}{k-1} \geq \binom{n}{k}\left(1-\frac{4k(n-k)}{n(n-1)}\right).$$

It is easy to verify that $1-4k(n-k)/(n(n-1)) > 1/n$ for all $k\leq n/2-2\sqrt{n}$. Hence, G has more than $\frac{1}{n}\sum_{k\leq n/2-2\sqrt{n}}\binom{n}{k}$ bad cuts. By the famous DeMoivre-Laplace theorem of probability theory, it follows that the last sum is at least $c2^n$ for some positive constant c. Thus, G has more than $c2^n/n$ bad cuts. $\square$

Consider `weighted max` k-`SAT`: Given a set $U=\{x_1,\dots,x_n\}$ of variables, and a collection $\{C_1,\dots,C_m\}$ of disjunctive clauses each with some positive weight w_i, find a truth assignment for U for which the sum of the weights of satisfied clauses is maximum. We assume that the *constant* $k\geq 2$. For simplicity, in the sequel `true` (`false`) will be replaced by the binaries 1 (0).

Berend and Skiena [4] analysed two algorithms for `(unweighted) max` k-`SAT`. Both turned out to be of domination number at most $n+1$ (despite the fact that one of the algorithms is a local search heuristic). This might indicate that `weighted max` k-`SAT` is $\mathcal{DOM}$-hard. Below we prove that `weighted max` k-`SAT` is, in fact, $\mathcal{DOM}$-easy.

Assign every variable its value 0 or 1 independently with probability 1/2. Let p_i be the probability that C_i is satisfied. Clearly, if C_i contains a variable and its negation, then $p_i=1$, otherwise $p_i = 1-2^{-k_i}$, where k_i is the number of distinct literals in C_i. Thus, the expectation of the total weight of clauses satisfied by a random assignment is $E = \sum_{i=1}^m w_i p_i$.

By the construction described in Section 15.2 of [1], there exists a binary matrix $A = (a_{ij})$ with n columns and $r = O(n^{\lfloor k/2 \rfloor})$ rows such that every $r \times k'$ submatrix of A contains an equal number of all binary k'-vectors (i.e., vectors with k' coordinates) for each $k' \leq k$. This matrix can be constructed in polynomial time. Consider the truth assignment $\beta_j : x_1 = a_{j1}, \dots, x_n = a_{jn}$. Let T_j be the total weight of clauses satisfied by β_j. Consider a polynomial algorithm $\mathcal{S}$ that computes $T_1, \dots, T_r$ and outputs $\beta^*(A)$ for which the total weight of satisfied clauses is $T^*(A) = \max_{j=1}^r T_j$. We have $\sum_{j=1}^r T_j = \sum_{i=1}^m r w_i p_i = rE$. Thus, $E \leq T^*(A)$. Besides, let $T_*(A) = \min_{j=1}^r T_j$, and let $\beta_*(A)$ be the corresponding assignment. Clearly, $T_*(A) \leq E$.

Consider all subsets of columns of A. For every such subset Q, replace all zeros (ones) by ones (zeros) in the columns of A. This results in a binary matrix A_Q with the same property as A. Thus, for the worst assignment $\beta_*(A_Q)$ the total weight of satisfied clauses is at most E. This way, we can find 2^n worst assignments (with repetitions) for which the total weight of satisfied clauses is at most E. Moreover, every worst assignment can be picked up at most $r = O(n^{\lfloor k/2 \rfloor})$ times as it may appear only in at most r matrices A_Q obtained from A by the operation described above. Thus, for at least $\Omega(2^n/n^{\lfloor k/2 \rfloor})$ truth assignments the total weight of satisfied clauses is at most that for $\beta^*(A)$. Hence, the domination number of $\mathcal{S}$ is at least $\Omega(2^n/n^{\lfloor k/2 \rfloor})$.

We have proved the following:

Theorem 4.3 `weighted max` k-`SAT` *is* $\mathcal{DOM}$*-easy.*

Consider `unweighted max SAT` (`max` k-`SAT` when k is not fixed). This is `weighted max SAT` in which the weight of each clause equals 1. We have

Theorem 4.4 `unweighted max SAT` *is* $\mathcal{DOM}$*-easy.*

Proof: The simplest randomized algorithm for `unweighted max SAT` consists of assigning every variable `true` (`false`) independently with probability 0.5. This algorithm is derandomized (by the method of conditional probabilities) in Section 5.4 of [2], where it is shown that the resulting polynomial deterministic algorithm $\mathcal{D}$ always satisfies at least $E = \sum_{i=1}^m p_i$ clauses (we use the notation in the proof of the previous theorem).

We claim that $\mathcal{D}$ is of domination ratio at least $1/(m+1)$, where, as above, m is the number of clauses. Clearly, $\mathcal{D}$ finds a truth assignment, which satisfies at least $A = \lceil E \rceil$ clauses. Observe that $A \leq m$. Assume that $\mathcal{D}$ is of domination ratio less than $1/(m+1)$. Then $(1 - 1/(m+1))(A+1) < A$, which implies $A > m$, a contradiction. □

5 Further Research

In this paper, we determined that some CO problems are $\mathcal{DOM}$-easy, others are $\mathcal{DOM}$-hard provided P≠NP. For many CO problems, it seems to be a quite non-trivial task to find out which of the two classes they belong to. One such example, is the quadratic assignment problem, a generalization of `min TSP`, which seems to be a good candidate into the class of $\mathcal{DOM}$-easy problems according to some results in [7]. Nevertheless, we do not know whether this or some other problems including `weighted max SAT` are $\mathcal{DOM}$-easy or not.

An interesting problem to investigate is the capacitated vehicle routing problem, a well-known generalization of `min TSP`. Another extension of `min TSP` is the Generalized TSP (`GTSP`): given a weighted complete k-partite digraph D with m vertices in each partite set, find a lightest cycle in D that contains exactly one vertex from each partite set. It is proved in [3] that there is a polynomial time algorithm for `GTSP` with domination ratio at least $1/((k-1)m^2)$. Thus, `GTSP` is $\mathcal{DOM}$-easy.

We have seen that `min partition` admits a polynomial time algorithm with domination ratio bounded from below by a constant. It would be interesting to determine what other $\mathcal{DOM}$-easy problems have this property.

Acknowledgement Part of the work was carried out while the first author was visiting University of Haifa, Israel in September 2002. He thanks the HIACS Research Center and the Caesarea Edmond Benjamin de Rothschild Foundation Institute for Interdisciplinary Applications of Computer Science (Israel) for financial support. He is also grateful to Noga Alon and Michael Krivelevich for useful discussions of the results of this paper and very helpful suggestions.

References

[1] N. Alon and J.H. Spencer, *The Probabilistic Method*, 2nd edition, Wiley, New York, 2000.

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

[3] D. Ben-Arieh, G. Gutin, M. Penn, A. Yeo and A. Zverovitch, Transformations of Generalized ATSP into ATSP. To appear in *Oper. Res. Letters*.

[4] D. Berend and S.S. Skiena, Combinatorial dominance guarantees for heuristic algorithms. Manuscript, 2002.

[5] G. Cornuejols, M.L. Fisher and G.L. Nemhauser, Location of bank accounts to optimize float; an analytic study of exact and approximate algorithms. *Management Sci.* **23** (1977) 789–810.

[6] F. Glover and A. Punnen, The traveling salesman problem: New solvable cases and linkages with the development of approximation algorithms, *J. Oper. Res. Soc.* **48** (1997) 502–510.

[7] G. Gutin and A. Yeo, Polynomial approximation algorithms for the TSP and the QAP with a factorial domination number. *Discrete Appl. Math.* **119** (2002) 107–116.

[8] G. Gutin, A. Yeo and A. Zverovitch, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Appl. Math.* **117** (2002) 81–86.

[9] G. Gutin, A. Yeo and A. Zverovitch, Exponential Neighborhoods and Domination Analysis for the TSP. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A.P. Punnen, eds.), Kluwer, Dordrecht, 2002.

[10] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, *Acta Mathematica* **182** (1999) 105–142.

[11] R. Hassin and S. Khuller, z-Approximations. *J. Algorithms* **41** (2001) 429–442.

[12] H. Kise, T. Ibaraki and H. Mine, Performance analysis of six approximation algorithms for the one-machine maximum lateness scheduling problem with ready times. *J. Oper. Res. Soc. Japan* **22** (1979) 205–223.

[13] A.E. Koller and S.D. Noble, Domination analysis of greedy heuristics for the frequency assignment problem. Submitted.

[14] C. Lund and M. Yannakakis, The approximation of maximum subgraph problems. *Proc. 20th Int. Colloq. on Automata, Languages and Programming*, Lect. Notes Comput. Sci. 700 (1993) Springer, Berlin, 40–51.

[15] A.P. Punnen, F. Margot and S.N. Kabadi, TSP heuristics: domination analysis and complexity. To appear in *Algorithmica*.

[16] A.P. Punnen and S. Kabadi, Domination analysis of some heuristics for the asymmetric traveling salesman problem, *Discrete Appl. Math.* **119** (2002) 117–128.

[17] V.I. Rublineckii, Estimates of the accuracy of procedures in the traveling selesman problem. *Numerical Mathematics and Computer Technology*, no. 4 (1973) 18–23. (in Russian)

[18] V.I. Sarvanov, On the minimization of a linear from on a set of all n-elements cycles. Vestsi Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk no. 4 (1976) 17–21 (in Russian).

[19] E. Zemel, Measuring the quality of approximate solutions to zero-one programming problems. *Math. Oper. Res.* **6** (1981) 319–332.