z-Approximations

Refael Hassin* Samir Khuller†

Abstract

Approximation algorithms for NP-hard optimization problems have been widely studied for over three decades. Most of these measure the quality of the solution produced by taking the ratio of the cost of the solution produced by the algorithm to the cost of an optimal solution. In certain cases, this ratio may not be very meaningful – for example, if the ratio of the worst solution to the best solution is at most some constant α, then an approximation algorithm with factor α may in fact yield the worst solution! To overcome this hurdle (among others), several authors have independently suggested the use of a different measure which we call z-approximation. An algorithm is an α z-approximation if it runs in polynomial time, and produces a solution whose distance from the optimal one is at most α times the distance between the optimal solution and the worst possible solution. The results known so far about z-approximations are either of the inapproximability type or rather straightforward observations. We design polynomial time algorithms for several fundamental discrete optimization problems, in particular we obtain a z-approximation factor of $\frac{1}{2}$ for the DIRECTED TRAVELING SALESMAN PROBLEM (TSP) (with no triangle inequality assumption). For the UNDIRECTED TSP this improves to $\frac{1}{3}$. We also show that if there is a polynomial time algorithm that for any fixed $\epsilon > 0$ yields an ϵ z-approximation then $P = NP$. We also present z-approximations for several other problems such as MAX CUT, STACKER CRANE, MAXIMUM ACYCLIC SUBGRAPH, MINIMUM DISJOINT CYCLE COVER.

1 Introduction

One way of coping with computational intractability of NP-hard optimization problems is to settle for polynomial time algorithms that produce sub-optimal solutions. This raises the question of selecting an appropriate measure for assessing the quality of a suboptimal solution. One popular measure, also called the "relative error", is to bound the maximum of $\frac{c(S)-c(S_{best})}{c(S_{best})}$ if we are solving a minimization problem. $c(S)$ is the cost of the solution S produced by the algorithm, and $c(S_{best})$ is the cost of an optimal solution. This is the same as measuring the approximation factor of an algorithm, which is $\frac{c(S)}{c(S_{best})}$ for a minimization problem.

Significant progress has been made in the field of approximation algorithms during the last decade (see [28] for a survey of results). Constant factor approximations have been obtained

*Department of Statistics and Operations Research, School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: hassin@post.tau.ac.il

†Department of Computer Science and UMIACS, University of Maryland, College Park, MD 20742, USA. Research supported by NSF Award CCR-9820965 and NSF CAREER Award CCR 9501355. Email: samir@cs.umd.edu

for many NP-hard problems, or proofs that no such algorithms exist, assuming $P \neq NP$, have been developed. However, in certain cases, an approximation algorithm may actually give the worst possible solution for that instance, and still be considered a "good" algorithm. For example, in a well-known (see [12]) factor 2 approximation algorithm for unweighted vertex cover, we find a maximal matching and take all the vertices that are matched in the matching. If the matching found is perfect, then we select the entire vertex set as the vertex cover, and simply argue that the optimal cover includes at least half the vertice. But this is the absolutely worst solution we could possibly produce! The above phenomenon is just one drawback of the commonly used "ratio criterion" which measures the quality of an approximate solution relative to the optimal one by their ratio. Early deviations from this measure can be found in the papers by Corneujols, Fisher and Nemhauser [13] and Kise, Ibaraki and Mine [29].

Several drawbacks of the "relative error" measure were raised in the seminal work of Zemel [38]. For example, if we change the problem instance so that we add some fixed constant to all solutions for an instance, the error ratio may not be preserved. [1] The second drawback is illustrated through the example of MINIMUM VERTEX COVER and MAXIMUM INDEPENDENT SET, which are very similar since the complement of an independent set is a vertex cover in the graph. However, the approximabilities of these two problems are vastly different. VERTEX COVER can be approximated within a factor of 2 [27, 7, 8] (see [28] for additional references), whereas MAXIMUM INDEPENDENT SET cannot be approximated within a factor of $\Omega(n^{1-\epsilon})$ for some fixed $\epsilon > 0$ assuming $NP \neq ZPP$ [21].

Ways of overcoming such drawbacks were suggested by Zemel [38], where an axiomatic approach is used to construct different measures. The measures suggested by Zemel are invariant under the following operations:

- adding a constant to the weight of every feasible solution (equivalently, adding a constant to the objective function);
- formulating the problem as a maximization or a minimization problem (with the objective function multiplied by -1);
- replacing the problem by its 'complement' which is defined in terms of the complement binary variables $\{y_i\}$ where $y_i = 1 - x_i$ instead of in terms of the variables $\{x_i\}$.

In the present paper we adopt one of Zemel's measures which we call *z-approximation*. In this approach our goal is to find a solution S with the property that

$$\frac{c(S) - c(S_{best})}{c(S_{worst}) - c(S_{best})} \leq \alpha$$

where $0 < \alpha < 1$ and S_{best} and S_{worst} are the optimal and worst cost solutions for the problem instance. For a minimization problem, the measure translates into

$$c(S) \leq (1-\alpha)c(S_{\min}) + \alpha c(S_{\max}),$$

[1] In the TSP, we can simply add a large constant to the weight of each edge, so that the edges now satisfy the triangle inequality, and use Christofides' heuristic, but this does not provide any relative error guarantee.

where $0 < \alpha < 1$ and $S_{\min}$ and $S_{\max}$ are the minimum and maximum cost solutions for the problem instance. For a maximization problem, it becomes

$$c(S) \geq (1-\alpha)c(S_{\max}) + \alpha c(S_{\min}).$$

An α z-approximation is *bounded* if $\alpha < 1$, for any constant α. The closer α is to 0, the better the algorithm. If $\alpha = 1$ then we might get a solution that is as bad as the worst solution.

Several other authors have independently recognized the advantages of the z-measure: Ausiello, D'Atri, and Protasi [4] denote it as 'the proximity degree' of a solution and attribute its formulation to [1] where a constant factor z-approximation is given for MAX CUT (see also Aiello, Burattini, Furnari, and Ventriglia [2]). Ausiello, Marchetti-Spaccamela, and Protasi [5] show that MAX-SUBSET SUM obtains a fully polynomial z-approximation scheme while MAX-CLIQUE and MAX CUT do not.

The recent work by Demange, Grisoni and Paschos [15, 14] develops an approach similar to the prior work by Zemel [38] and also give algorithms under the new measure for a set of problems. In particular, [15] contains a $1 - \frac{1}{\Delta}$ z-approximation for UNWEIGHTED SET COVER with maximum set size Δ, and proves the existence of an asymptotic 0 z-approximation for instances of VERTEX COLORING with $\Delta = o(|V|)$ where Δ is the maximum vertex degree in the graph. (The bounds in [15] are actually $1 - \alpha$, where α is the z-approximation factor.)

Finally, Nemirovsky and Yudin [33], Vavasis [37] and Bellare and Rogaway [6] use this measure in the context of non-linear programming.

We note that the meaning of a 'worst' solution ($S_{\max}$ in a minimization problem or $S_{\min}$ in a maximization problem) needs to be defined together with the problem. For example, in VERTEX COLORING one possibility is to give each vertex a distinct color, and thus $c(S_{\max}) = |V|$. Another possibility is to allow only *minimal colorings*, that is solutions such that no two color classes can be merged without creating a conflict (an edge whose two ends are colored the same way). In this case, $c(S_{max})$ is the maximum number of colors in a minimal coloring, which may be considerably less than $|V|$ and the problem turns out to be harder to z-approximate.

In this paper we investigate a few central combinatorial optimization problems and analyze their z-approximability. It is of interest to compare these results to those obtained for the same problems with respect to the ratio measure and to note that some problems are harder to z-approximate while others are easier than the respective ratio approximation. For example, VERTEX COLORING is known to be hard to approximate under the standard measure but it turns out that bounded z-approximations can be easily obtained (see Section 4). For MAXIMUM INDEPENDENT SET the two measures are identical and therefore the known hardness results for this problem (see [21]) are also valid for z-approximation. Since the z-measure is invariant with respect to defining the problem in terms of the complement binary variables, it follows that vertex cover is also hard to z-approximate with $\alpha < 1$. However, this problem has very simple 2-approximations. The Traveling Salesman Problem (TSP) which asks for a minimum cost (or length) Hamiltonian cycle (a *tour*), is an interesting case since its minimization version is hard to approximate while for the maximization version there are bounded approximations [30, 26] (this is true for both the directed and undirected version). On the other hand, the z-measure is invariant with respect to whether we maximize or minimize and we show that even for the directed case there is a bounded z-approximation (see Section 2). This implies one for the simpler version, the undirected case (the factor is a little better). This also makes no assumption about edge weights such as non-negativity and triangle inequality.

2 The traveling salesman problem

In this section we describe a $\frac{1}{2}$ z-approximation algorithm for the directed TSP. We assume that we are given a directed graph $G = (V, A)$ with weights $c(i, j)$ $(i, j) \in A$. Note that *we do not assume non-negativity of the weights* and (in particular) the results apply to both the minimization and maximization versions of the problem.

We observe that well-known algorithms which guarantee bounded ratio approximations for minimum (metric) TSP on undirected graphs may still output worst solutions on such instances. Note that the triangle inequality is irrelevant for z-approximations since the latter is invariant under the addition of a constant to all of the edge weights. For example, consider a complete graph with unit length edges for all but a single edge, say (a, b), whose length is two. Christofides' algorithm [11], whose approximation factor for the metric TSP is $\frac{3}{2}$, may compute as a minimum spanning tree a Hamiltonian path between vertices a and b. The matching step will add to this path the edge (a, b) to form a Hamiltonian cycle of total length $n + 1$, which is the worst (maximum) possible length.

Remark: Another measure suggested by Zemel is the relative location of a solution in a sorted list of feasible solutions to the problem. Several papers computed solutions for the TSP that will be not worse than many other solutions. A recent paper by Gutin and Yeo [20] shows that a tour whose weight is at most that of an *average* tour (such a tour can be computed in polynomial time) dominates at least a fraction of about $1/n^2$ of the solutions to a TSP with n vertices. We note however, that this solution may still be close to the worst one in an instance where many solutions of the same (worst) value exist.

A *cycle cover* is a set of vertex-disjoint cycles which span the vertex set. If the graph is directed then this cycles should be directed. A minimum cost cycle cover can be computed in polynomial time in the directed case, by reducing the problem to the assignment problem. (In the undirected case a cycle cover is also called a *binary 2-matching*. This can also be obtained in polynomial time [32, 23].)

Algorithm:

Compute a minimum cost directed cycle cover DCC. If it contains a single cycle return this cycle (it is an optimal tour). Otherwise, arbitrarily fix a cyclic order of the cycles in DCC. Denote the cycles in the order fixed above as $C_1, ..., C_l$, $l \geq 2$.

Randomly choose a vertex $v_1^i \in C_i$ for $i = 1, ..., l$. Index the vertices of C_i starting from v_1^i in the direction dictated by the cycle's orientation, as $v_1^i, v_2^i, ..., v_{k_i}^i$, where $k_i = |C_i|$. Let $P_{p,q}^i$ denote the sequence of vertices between $p, q \in C_i$ starting from p in the direction dictated by the orientation of C_i.

Return the tour S_{good} which visits the vertices in order of $P_{2,1}^1, ..., P_{2,1}^l$.

Theorem 2.1 *Let $c(S_{good})$ denote the expected cost of the solution S_{good}, then*

$$c(S_{good}) \leq c(S_{min}) + \frac{1}{2}(c(S_{max}) - c(S_{min})).$$

Proof: Consider a tour $S_{bad} = (P^i_{3,1})_{i=1,\ldots,l}(v^i_2)_{i=1,\ldots,l}$. If $k_i = 2$ then $v^i_3 \equiv v^i_1$ and $P^i_{3,1} =$ contains just this vertex. Denote by $c(S_{bad})$ the expected length of S_{bad}. S_{good} deleted an arc from each cycle and added an arc from each cycle to its successor. S_{bad} deleted two arcs in each cycle and added two arcs from each cycle to its successor. Since each arc in a cycle has the same probability of being deleted, and similarly each arc from a given cycle to its successor has the same probability of being used, it follows that

$$c(S_{max}) - c(S_{good}) \geq c(S_{bad}) - c(S_{good}) = c(S_{good}) - c(DCC) \geq c(S_{good}) - c(S_{min})$$

and $c(S_{good}) \leq \frac{1}{2}(c(S_{max}) + c(S_{min}))$. ∎

The algorithm can be derandomized to obtain a deterministic $\frac{1}{2}$ z-approximation. Applying the method of conditional expectations, we first choose the arc to be deleted from C_1 to satisfy

$$\min_{(j,k)\in C_1}\{\sum_{p\in C_l} c(p,k) + \sum_{q\in C_2} c(j,q) - c(j,k)\}.$$

Given the arcs deleted from $C_1, ..., C_{p-1}$, and in particular that an arc that leaves a vertex $i \in C_{p-1}$ was deleted, we choose an arc to be deleted from C_p $p = 2, ..., l-1$ to satisfy

$$\min_{(j,k)\in C_p}\{c(i,k) + \sum_{q\in C_{p+1}} c(j,q) - c(j,k)\}.$$

Finally, given that arcs leaving $i \in C_{l-1}$ and entering $j \in C_1$ where chosen for deletion, we delete an arc from C_l to satisfy

$$\min_{(s,t)\in C_l}\{c(i,s) + c(t,j) - c(s,t)\}.$$

The following result is obtained in a similar way and its proof is omitted. The main difference is that each cycle in a cycle cover in an undirected graph has size at least three, whereas in the directed case, each cycle in the cycle cover has size at least two. In the undirected case, we can perform an extra exchange operation to swap another edge from each cycle, thus creating a bad solution from the cycle cover by using two sets of swaps. We thus create a bad solution by going over the set of cycles three times, rather than two times as was done in the directed case.

Theorem 2.2 *There is $\frac{1}{3}$ z-approximation for the undirected TSP.*

One natural question that arises is "Are there ϵ z-approximations that run in polynomial time for any fixed constant ϵ for TSP?" We give a negative answer to this question by relating it to the MAX SNP hardness of a special case of metric TSP.

Theorem 2.3 *If there is a polynomial time algorithm that for any fixed $\epsilon > 0$ yields an ϵ z-approximation for TSP then $P = NP$.*

Proof: If there is a polynomial time algorithm that for each fixed $\epsilon > 0$ finds an ϵ z-approximation, then it finds a solution S with cost at most $c(S) \leq c(T_{\min}) + \epsilon(c(T_{\max}) - c(T_{\min}))$. Since the TSP with distances between vertices being either 1 or 2 (called 1/2 TSP) is MAX SNP-hard [34], we know that if we get a solution that is an ϵ z-approximation, then $P = NP$. For 1/2 TSP instances, this implies that $c(S) \leq c(T_{\min}) + \epsilon(2|V| - |V|) \leq c(T_{\min}) + \epsilon c(T_{\min}) \leq (1+\epsilon)c(T_{\min})$. This implies that there is a PTAS for for 1/2 TSP, which in turn implies that $P = NP$ [34]. ■

2.1 The Stacker crane problem

In the STACKER CRANE problem, an undirected graph is given with a set of *special directed arcs*. The goal is to compute a minimum length walk that traverses all of the special arcs in the desired orientation. For the metric case, Frederickson, Hecht and Kim [17] developed a 1.8-approximation algorithm. However, if the length of a solution is defined as the total length of the connecting edges in it (excluding the constant term consisting of the length of the special arcs) then the approximation factor is 3 (Algorithm LONGARCS [17]). Thus, a better approximation factor was made possible by implicitly adding a constant to the objective function.

We note that the z-measure is invariant under the addition of a constant to the objective function.

We view the STACKER CRANE problem as follows. Given a weighted graph with an oriented perfect matching $M = \{(s_1, t_1), ..., (s_{\frac{n}{2}}, t_{\frac{n}{2}})\}$ compute a minimum cost directed Hamiltonian cycle that uses edges of the matching M in the specified orientation. We note that for this version the best and worst solution values are well defined. Moreover, it turns out that under this definition the problem is in fact the directed TSP. To see this note that a vertex v_i in a directed TSP can be considered a special arc (s_i, t_i) with the arcs (v_i, v_j) represented by (t_i, s_j) for every $i \neq j$. Similarly, a special arc (s_i, t_i) in a Stacker Crane instance can be considered a vertex in an equivalent directed TSP instance where (t_i, s_j) is represented by a directed arc (v_i, v_j). It follows that our $\frac{1}{2}$ z-approximation for the directed TSP also applies to STACKER CRANE.

We note the transformation holds also for the general case, that is when the special directed arcs are not necessarily vertex disjoint, but the definition of the worst solution may be less natural since we must allow the solution to repeat edges.

3 Max cut and Min cluster

In MAX CUT a graph with non-negative edge weights is given. The goal is to partition the vertices into two non-empty sets so that the total weight of edges with end vertices in different sets is maximized. Similarly, the complementary MIN CLUSTER asks to minimize the total weight of edges whose two ends are in the same part.

There are many $\frac{1}{2}$-approximation algorithms for MAX CUT (see [18] for improvements).

These algorithms guarantee a stronger property, namely, the weight of the solution S_{apx} which they produce is guaranteed to satisfy $c(S_{apx}) \geq \frac{1}{2}c(G)$, where $c(G)$ is the total weight of the edges of G.

Denote by S_{max} a MAX CUT and similarly by S_{min} a minimum cut. We now show that when the latter is used as a 'worst' solution value the above property guarantees asymptotically a $\frac{1}{2}$ z-approximation.

Theorem 3.1 *S_{apx} is an α z-approximation, for $\alpha = \frac{1}{2}(\frac{n}{n-2})$.*

Proof: We need to argue that

$$c(S_{apx}) \geq \alpha c(S_{\min}) + (1-\alpha)c(S_{\max}).$$

Consider a graph with n vertices. The minimum cut value is bounded by the average weight of a set of edges (a star) incident with a vertex. Thus, $c(S_{min}) \leq 2\frac{c(G)}{n}$. We also have $c(S_{apx}) \geq \frac{1}{2}c(G)$ and $c(S_{max}) \leq c(G)$. (Note that for the latter inequality we use the assumption that the edge weights are non-negative.) This follows from the above inequalities for $c(S_{\max})$ and $c(S_{\min})$. ∎

4 Vertex cover, max clique and cover by independent sets

There is a simple greedy algorithm that gives an approximation factor of 2 [7, 27] for VERTEX COVER and this is the best asymptotic factor that is known. (See [8] for an improvement to $2 - \frac{\log\log|V|}{2\log|V|}$.) As an optimization problem, VERTEX COVER is equivalent to MAXIMUM INDEPENDENT SET since the complement of a vertex cover is an independent set. However, for MAXIMUM INDEPENDENT SET (or the equivalent MAXIMUM CLIQUE which is defined on the complement graph) the best known approximation factor is $O(|V|/(\log|V|)^2)$ [10]. On the other hand, letting the worst independent set solution consist of the empty set, with zero weight, the z-measure is exactly the ratio measure. Due to Hastad's hardness result [21] it follows that the problem is also hard to z-approximate. Since z-approximation is invariant under the complement operation it follows that also VERTEX COVER is hard to z-approximate.

We note that in contrast to the above observation, constructing bounded z-approximations for CLIQUE COVER (or VERTEX COLORING) is quite easy. Very simple algorithms with factors of $\frac{1}{2}$ and $\frac{1}{3}$ are described in [24], improvements are given in [22] (see also [36]), and finally a factor of $\frac{59}{72}$ in [16]. These algorithms approximate the COLOR SAVING PROBLEM which amounts to maximizing the saving in colors over the trivial solution which assigns a different color to each vertex. Since the minimum numbers of colors is some positive number, any factor obtained for COLOR SAVING induces at least the same bound for the z-approximation of the problem.

We now generalize the above results. An *independence system* is a pair $(V, \mathcal{F})$ where V is a ground set and $\mathcal{F}$ is a collection of subsets of V which are said to be *independent* satisfying the condition that if $F \in \mathcal{F}$ and $F' \subset F$ then $F' \in \mathcal{F}$. Trivially, any singleton of V is independent. Given an independence system $(V, \mathcal{F})$ MINIMUM COVERING BY INDEPENDENT SETS requires to compute a minimum number of independent sets whose union is V. An interesting special case of this problem is when V is the vertex set of a graph and a set of vertices is independent if there is no edge connecting any two vertices in the set. In this case the problem is exactly VERTEX COLORING discussed above. Another interesting case is one where each element of V is associated with a weight and a set is independent if the total weight of its members is at most 1. In this case the problem becomes the BIN PACKING PROBLEM.

Since the singletons are independent, $|V|$ sets are always sufficient to cover V. Thus, the z-factor turns to be the *complementary performance ratio* $\frac{|V|-c(S_{min})}{|V|-c(S)}$ [22]. The method used by [24, 22, 16] is to use a maximal set of independent sets of size at least 4 and then use smaller sets to cover the remaining elements. Using better approximations for the problem of covering a set by subsets of size at most 3, resulted in better z-approximation for COLOR SAVINGS. By following exactly the same procedure, the same z-factors are obtained for the general MINIMUM COVERING BY INDEPENDENT SETS.

5 Minimum disjoint cycle cover

Given an undirected graph $G = (V, E)$, a MINIMUM DISJOINT CYCLE COVER is to compute a set of vertex disjoint cycles of minimum cardinality, that span V. In this problem, a single vertex is considered a cycle and therefore a feasible solution always exists. There is no k-approximation for the problem, for any constant k, as proved by Sahni and Gonzalez [35].

We compute a $\frac{1}{3}$ z-approximation for the problem with the definition of the 'worst' solution as a collection of $|V|$ single vertices, as follows:

Construct a complete graph H with vertex set V and weights

$$c(i,j) = \begin{cases} 1 & i = j \\ 0 & (i,j) \in E \;\; i \neq j \\ |V|+1 & (i,j) \notin E \;\; i \neq j \end{cases} .$$

Compute a minimum cost cycle cover in H and denote the number of cycles by apx. Let apx_1 and apx_2 denote the number of cycles in this solution with a single vertex and more than one vertex, respectively.

Consider an optimal solution. Denote the number of cycles in this solution by opt and similarly break it into opt_1 and opt_2. Then:

$$apx_2 \leq \frac{|V| - apx_1}{3}.$$

Note that the cost of a cycle cover in H amounts to the number of loops it contains, so that the minimum cost of a cycle cover is exactly apx_1, and in particular

$$apx_1 \leq opt_1.$$

Therefore,

$$apx \;\; = \;\; apx_1 + apx_2$$

$$
\begin{aligned}
&\leq apx_1 + \frac{n - apx_1}{3} \\
&= \frac{n}{3} + \frac{2}{3} apx_1 \\
&\leq \frac{n}{3} + \frac{2}{3} opt_1 \\
&\leq \frac{n}{3} + \frac{2}{3} c(S_{min}) \\
&= \frac{c(S_{max})}{3} + \frac{2}{3} c(S_{min}) \\
&= c(S_{min}) + \frac{1}{3}(c(S_{max}) - c(S_{min})).
\end{aligned}
$$

We conclude that a minimum cost cycle cover in H is a $\frac{1}{3}$ z-approximation for the problem.

6 Minimum dominating set

A dominating set of a graph $G = (V, E)$ is a subset $U \subseteq V$ such that every vertex in $V \setminus U$ has at least one neighbor in U. The MINIMUM DOMINATING SET problem is to compute a dominating set of minimum cardinality. This is a special SET COVER instance and it can be approximated within $O(\log |V|)$. No constant performance guarantee is possible unless $P = NP$. Let δ denote the minimum vertex degree in G. Without loss of generality we can assume that $\delta > 0$ (incorporating isolated vertices is like adding a constant to the objective function). We also assume that the graph is connected (otherwise, we can run the algorithm on each connected component of the graph). It is straightforward to construct a dominating set of size at most $|V|/2$. For example, consider a bipartition induced by any spanning tree. and choose the smaller part. Thus, if we define the 'worst' solution as $|V|$ this gives immediately a $\frac{1}{2}$ z-approximation for the problem.

We note that stronger bounds are possible when δ is bounded from below. For example, Theorem 2.2 [3] states that G has a dominating set of size at most $|V|(1 + \ln(\delta + 1))/(\delta + 1)$. This bound can be achieved by a simple randomized algorithm, or deterministically by a greedy algorithm which at each step picks a vertex that covers the maximum number of uncovered vertices. This result immediately gives improved z-approximations for the problem, depending on δ.

7 Maximum acyclic subgraph and minimum feedback arc set

In MAXIMUM ACYCLIC SUBGRAPH an arc-weighted directed graph $G = (V, A)$ is given and we want to compute a subset $A' \subseteq A$ such that $G' = (V, A')$ is acyclic and $c(A')$ is maximal. The complementary problem in which $c(A \setminus A')$ is to be minimized is called MINIMUM FEEDBACK ARC SET. There is a simple $\frac{1}{2}$-approximation for MAXIMUM ACYCLIC SUBGRAPH: Compare the set of arcs $(i, j) \in A$ such that $i < j$ with its complement and choose the one with higher weight.

For the unit weight version of the problem there are better bounds for graphs with bounded degree. The algorithms of Berger and Shor [9] and Hassin and Rubinstein [25] are $\frac{1}{2}(1 + \Omega(1/\sqrt{\Delta}))$-approximations, where Δ is the maximum degree.

Clearly, if there are negative arc weights then an optimal solution will never choose these arcs. Let $c'(i, j) = \max\{0, c(i, j)\}$. Clearly $c(S_{min}) \leq 0$ since the empty set is a feasible solution, and therefore the approximate solution with value at least $c'(A)$ is also a $\frac{1}{2}$ z-approximation.

Let us consider a variation of the problem in which a solution corresponds to a permutation of the indices of the vertices, and its cost is the total cost of the arcs directed from a vertex to one with a higher index.

In other words, we now consider *maximal* sets of arcs that induce acyclic subgraphs, that is, subsets of A which induce acyclic subgraphs and are not properly contained in any other such set. We then define $c(S_{min})$ as the minimum weight of a maximal solution. As observed in [25] the maximal solutions correspond to permutations $\pi = (\pi_1, ..., \pi_{|V|})$ of the indices by taking the arcs (i, j) with $\pi_i < \pi_j$. Thus each maximal solution is associated with another one obtained by the reverse permutation and in particular $c(S_{min}) + c(S_{max}) = c(A)$.

Under non-negative costs this problem is equivalent (with respect to optimization) to MAXIMUM ACYCLIC SUBGRAPH. The main difference is however that now we may have to use negatively weighted arcs and thus the optimal solution cost is not necessarily bounded by the total cost of the graph.

We observe that the approximation for the weighted case by the above simple algorithm is guaranteed to be at least $\frac{1}{2}c(A)$. It follows that such a solution is also a $\frac{1}{2}$ z-approximation for the problem.

For the unit weights case, again the approximations are guaranteed to be at least $\frac{1}{2}(1 + \Omega(1/\sqrt{\Delta}))$ times $c(A)$ so that they give the same z-bound.

8 Concluding remarks

The entire theory of approximation algorithms that tries to characterize problems by levels of difficulty, in terms of approximability, can be redone in this framework. We have examples of problems that are hard to z-approximate (like VERTEX COVER and INDEPENDENT SET), that are easy to z-approximate (asymmetric TSP, DISJOINT CYCLE COVER, MAX CUT) and problems with ϵ z-approximations, for any fixed $\epsilon > 0$, like KNAPSACK (this is a maximization problem, and we could take the worst solution that corresponds to an empty knapsack with zero profit).

For example, we prove the hardness of undirected TSP by showing that there is no polynomial algorithm that can produce an ϵ z-approximation for any fixed $\epsilon > 0$, unless $P = NP$. Such results can be shown for problems for which we can "restrict" the cost of possible solutions. (For example, in the proof of Theorem 2.3 the cost of the tour is between n and $2n$.) For example, for the Vertex Cover problem, Nemhauser and Trotter [31] showed that if we take an optimal fractional solution for the relaxation of the Integer Program (IP) for Vertex Cover, one can prove some nice properties about the solution. Without loss of generality, the (optimal) fractional solution is restricted to the values $\{0, \frac{1}{2}, 1\}$. Moreover, Nemhauser and Trotter showed that there exists an optimal cover that includes all the vertices with a value of 1, and a subset of the vertices with value $\frac{1}{2}$. We can focus our attention on the subgraph induced by the vertices with value $\frac{1}{2}$. Thus, for the LP solutions for Vertex Cover [31] we can restrict the optimal vertex cover cost between $w(V)$ and $\frac{1}{2}w(V)$ and obtain similar hardness results for Vertex Cover.

References

[1] A. Aiello, E. Burattini, M. Massarotti, and F. Ventriglia, "A new evaluation function for approximation algorithm", *Sem. IRIA* (1977).

[2] A. Aiello, E. Burattini, M. Furnari, and F. Ventriglia, "Computational complexity: the problem of approximation". In: *Algebra, Combinatorics, and Logic in Computer Science* **1**:51–62, Colloquia Mathematica Societatis János Bolyai 42, North Holland, New York (1986).

[3] N. Alon and J. H. Spencer, *The Probabilistic Method*, Wiley & Sons, 1992.

[4] G. Ausiello, A. D'Atri, and M. Protasi, "Structure preserving reductions among convex optimization problems", *Journal of Computer and System Sciences* **21**:136–153 (1980).

[5] G. Ausiello, A. Marchetti-Spaccamela, and M. Protasi, "Toward a unified approach for the classification of NP-complete optimization problems", *Theoretical Computer Science* **12**:83–96 (1980).

[6] M. Bellare and P. Rogaway, "The complexity of approximating a nonlinear program", *Mathematical Programming* **69**:429–441 (1995).

[7] R. Bar-Yehuda and S. Even, "A linear time approximation algorithm for the weighted vertex cover problem", *Journal of Algorithms*, **2**:198–203 (1981).

[8] R. Bar-Yehuda and S. Even, "A local ratio theorem for approximating the weighted vertex cover problem", *Annals of Discrete Mathematics*, **25**:27–45 (1985).

[9] B. Berger and P. W. Shor, "Approximation algorithms for the maximum acyclic subgraph problem", *Proc. 1st Ann. ACM-SIAM Symp. on Discrete Algorithms*, 236–243 (1990).

[10] R. Boppana and M. M. Halldórsson, "Approximating maximum independent sets by excluding subgraphs", *BIT* **32**:180–196 (1992).

[11] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University (1976).

[12] T. H. Cormen, C. E. Leiserson and R. L. Rivest, "Introduction to Algorithms", *MIT Press* (1989).

[13] G. Corneujols, M. Fisher and G. Nemhauser, "Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms", *Management Science*, **23**(8):789–810 (1977).

[14] M. Demange and V. T. Paschos, "On an approximation measure founded on the links between optimization and polynomial approximation theory", *Theoretical Computer Science*, **158**:117–141 (1996).

[15] M. Demange, P. Grisoni and V. T. Paschos, "Differential approximation algorithms for some combinatorial optimization problems", *Theoretical Computer Science*, **209**(1-2):107–122 (1998).

[16] R. Duh and M. Fürer, "Approximation of k-set cover by semi-local optimization", *Proc. of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, 256–264 (1997).

[17] G. N. Frederickson, M. S. Hecht and C. E. Kim, "Approximation algorithms for some routing problems," *SIAM J. Comput.* **7**:178–193 (1978).

[18] M. Goemans and D. Williamson, "0.878 approximation algorithms for MAX-CUT and MAX-2-SAT", *Proc. of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, 256–264 (1994).

[19] L. K. Grover, "Local search and the local structure of NP-complete problems", *Operations Research Letters* **12**:235-243 (1992).

[20] G. Gutin and A. Yeo, "Polynomial algorithms for the TSP and the QAP with a factorial domination number", manuscript (1998).

[21] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$", *37th Annual Symposium on Foundations of Computer Science*, 627–636 (1996).

[22] M. M. Halldórsson, "Approximating k-set cover and complementary graph coloring", *IPCO 96*, (1996).

[23] D. Hartvigsen, "Extensions of Matching Theory", Ph.D. Thesis, Carnegie-Mellon University (1984).

[24] R. Hassin and S. Lahav, "Maximizing the number of unused colors in the vertex coloring problem", *Information Processing Letters* **52**:87–90 (1994).

[25] R. Hassin and S. Rubinstein, "Approximation for the maximum acyclic subgraph problem", *Information Processing Letters* **51**:133–140 (1994).

[26] R. Hassin and S. Rubinstein, "Better approximations for Max TSP", **67**(3):125–130 (1998).

[27] D. Hochbaum, "Efficient bounds for the stable set, vertex cover, and set packing problems", *Discrete Applied Mathematics*, **6**:243–254 (1983).

[28] D. S. Hochbaum (Ed), *Approximation algorithms for NP-hard problems.* PWS Publishing Co., Boston, MA, (1996).

[29] H. Kise, T. Ibaraki and H. Mine, "Performance analysis of six approximation algorithms for the one-machine maximum lateness scheduling problem with ready times", *Journal of the Operations Research Society of Japan* **22**:205–223 (1979).

[30] R. Kosaraju, J. Park and C. Stein, "Long tours and short superstrings", *35th Annual Symposium on Foundations of Computer Science*, (1994), 166–177.

[31] G. L. Nemhauser and L. E. Trotter, Jr., "Vertex packing: structural properties and algorithms", *Mathematical Programming*, **8**:232–248 (1975).

[32] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization* Wiley & Sons (1988).

[33] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley, Chichester (1983). [Translated by E. R. Dawson from *Slozhnost' Zadach i Effektivnost' Metodov Optimizatsii* (1979).]

[34] C. H. Papadimitriou and M. Yannakakis, "The traveling salesman problem with distances one and two", *Mathematics of Operations Research*, **18**(1):1–11 (1993).

[35] S. Sahni and T. Gonzalez, "P-complete approximation problems", *JACM*, **23**:555–565 (1976).

[36] W. D. Tzeng and G. H. King, "Three-quarter approximation for the number of unused colors in graph coloring", *Information Sciences*, **114**:105–126 (1999).

[37] S. A. Vavasis, "Approximation algorithms for indefinite quadratic programming", *Mathematical Programming* **57**:279-311 (1972).

[38] E. Zemel, "Measuring the quality of approximate solutions to zero-one programming problems", *Mathematics of Operations Research* **6**:319–332 (1981).