ELSEVIER

Discrete Applied Mathematics 91 (1999) 251–263

DISCRETE APPLIED MATHEMATICS

On the complexity of postoptimality analysis of 0/1 programs

Stan Van Hoesel[a], Albert Wagelmans[b,*,1]

[a] *Department of Quantitative Economics, University of Limburg, P.O. Box 616, NL-6200 MD, Maastricht, Netherlands*

[b] *Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, Netherlands*

Received 5 November 1996; received in revised form 15 September 1998; accepted 12 October 1998

Abstract

In this paper we address the complexity of postoptimality analysis of 0/1 programs with a linear objective function. After an optimal solution has been determined for a given cost vector, one may want to know how much each cost coefficient can vary individually without affecting the optimality of the solution. We show that, under mild conditions, the existence of a polynomial method to calculate these maximal ranges implies a polynomial method to solve the 0/1 program itself. As a consequence, postoptimality analysis of many well-known NP-hard problems cannot be performed by polynomial methods, unless $\mathscr{P} = \mathscr{NP}$. A natural question that arises with respect to these problems is whether it is possible to calculate in polynomial time reasonable approximations of the maximal ranges. We show that it is equally unlikely that there exists a polynomial method that calculates conservative ranges for which the relative deviation from the true ranges is guaranteed to be at most some constant. Finally, we address the issue of postoptimality analysis of ε-optimal solutions of NP-hard 0/1 problems. It is shown that for an ε-optimal solution that has been determined in polynomial time, it is not possible to calculate in polynomial time the maximal amount by which a cost coefficient can be increased such that the solution remains ε-optimal, unless $\mathscr{P} = \mathscr{NP}$.

Keywords: Postoptimality analysis; Computational complexity

0. Introduction

Whereas sensitivity analysis is a well-established topic in linear programming (see [2] for a comprehensive review), its counterpart in mixed integer programming and

* Corresponding author. Fax: +31 10 408 9162; e-mail: wagelmans@few.eur.nl.

[1] Part of this research was carried out at the Operations Research Center, Massachusetts Institute of Technology; financial support of the Netherlands Organization for Scientific Research (NWO) is gratefully acknowledged.

PII: S0166-218X(98)00151-6

combinatorial optimization is a much less developed research area. The excellent annotated bibliography by Greenberg [4] shows that in the last 20 years results have appeared more or less isolated in the literature, but that quite recently there seems to be an increased interest.

In this paper we address the complexity of postoptimality analysis of 0/1 programs with a linear objective function. The first complexity results with respect to stability analysis of such problems have appeared in the Russian literature. We refer to Sotskov et al. [10] for a review of these results, which mainly relate to situations in which several problem parameters may vary simultaneously. Our results concern individual changes of parameters. To be more precise, we consider the situation in which an optimal solution has been determined with respect to a given cost vector and one wants to know how much each cost coefficient can vary individually without affecting the optimality of the solution. We show that, under mild conditions, the existence of a polynomial method to calculate these maximal ranges implies a polynomial method to solve the 0/1 program itself. As a consequence, postoptimality analysis of many well-known NP-hard problems cannot be performed by polynomial methods, unless $\mathscr{P}=\mathscr{NP}$. A natural question that arises with respect to these problems is whether it is possible to calculate in polynomial time reasonable approximations of the maximal ranges. We show that it is equally unlikely that there exists a polynomial method that calculates conservative ranges for which the relative deviation from the true ranges is guaranteed to be at most some constant.

Of course, one is not always willing or able to compute an optimal solution of an NP-hard problem and much research has been devoted to the design of fast heuristics. The performance of these heuristics can either be evaluated experimentally or theoretically. In the latter case one often tries to prove that the heuristic always produces ε-optimal solutions, i.e., the relative deviation of the solution value from the optimal value is less than some constant ε. This means that we have a guarantee on the quality of the solution that the heuristic produces and we may be interested to know under which changes of the cost coefficients this guarantee still holds. Therefore, we also study the complexity of postoptimality analysis of ε-optimal solutions of NP-hard 0/1 problems. Our result is that for an ε-optimal solution that has been determined in polynomial time, it is impossible to calculate in polynomial time the maximal amount by which a cost coefficient can be increased such that the solution remains ε-optimal, unless $\mathscr{P}=\mathscr{NP}$.

Despite these negative results, one may still want to calculate (approximations to) the stability measures mentioned above. Algorithms to do so have been proposed in several papers. The interested reader is referred to Gordeev et al. [3], Sotskov [9], Kravchenko et al. [6], Sotskov et al. [11], Libura et al. [7] and Chakravarti and Wagelmans [1].

Finally, we should mention that results quite similar to our main results (Theorems 1 and 2 in this paper) have independently been obtained by Ramaswamy and Chakravarti [8]. The difference between their and our results is discussed in Section 1. Ramaswamy and Chakravarti have also studied problems with a min–max objective function. They show that for these problems it is again unlikely that the maximal ranges can be computed in polynomial time if the problem itself is NP-hard. Furthermore, they also show positive results: both for linear and min–max objective functions, the maximal ranges can be computed in polynomial time if the problem itself is polynomial solvable.

This paper is organized as follows. In Section 1 we prove our main results with respect to optimal solutions. The results on ε-optimal solutions are presented in Section 2. Section 3 contains concluding remarks.

1. Postoptimality analysis of optimal solutions

Consider an optimization problem of the following form:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & x \in X \subset \{0,1\}^n \end{array} \qquad \text{(P)}$$

with $c \in \mathbb{Q}_+^n$, i.e., the cost coefficients are positive rationals. Throughout this paper we will only consider rational cost coefficients, because computational complexity theory only concerns such problems (see, for instance, [5]). Note that irrational values can be approximated by rationals and that we may as well assume that all cost coefficients are integers. Furthermore, throughout this paper we assume that X, the set of feasible solutions, does not depend on the cost coefficients.

We will prove two theorems with respect to (P) and discuss their implications. The first theorem concerns decreasing cost coefficients.

Theorem 1. *Consider* (P) *for a fixed set $X \subset \{0,1\}^n$. This problem is polynomially solvable for any $c \in \mathbb{Q}_+^n$, if the following two conditions are both satisfied.*

(a) *For every problem instance it takes polynomial time to determine a feasible solution $x \in X$ which is minimal, i.e., there does not exist another feasible solution $x' \in X$ with $x' \leqslant x$.*

(b) *For every cost vector $c' \in \mathbb{Q}_+^n$ and for every optimal solution x of the problem instance defined by c', the maximal value l_i by which the cost coefficient of x_i, $i = 1, \ldots, n$, may be decreased such that x remains optimal, can be determined in polynomial time. Here $l_i \equiv c'_i$ if x remains optimal for arbitrarily small positive cost coefficients of x_i.*

This theorem has implications for many well-known NP-hard problems. For instance, we are able to conclude that, unless $\mathscr{P} = \mathscr{NP}$, it is impossible to determine in polynomial time the maximal amounts by which the distances in a traveling salesman problem (TSP) can be decreased individually without affecting the optimality of a given tour.

The proof of the theorem makes use of four lemmas which we will prove first.

Assume that a polynomial procedure $LOW(i,c,x)$ calculates l_i, $i \in \{1, \ldots, n\}$, as defined under (b) of the theorem with respect to the cost vector c and a given corresponding optimal solution x. Furthermore, define $N(j,c,\delta)$ to be the vector obtained from c by replacing c_j by $c_j - \delta$, and let $A(j,c,\delta)$ be the set of optimal solutions to (P), when c is replaced by $N(j,c,\delta)$.

Lemma 1. *If $LOW(j,c,x) = 0$ for some $j \in \{1,\dots,n\}$, then $x_j = 0$. Furthermore, $A(j,c,\delta)$ is the same for all $0 < \delta < c_j$ and this set consists of exactly those solutions x' which are optimal with respect to c and have $x'_j = 1$.*

Proof. The key observation is that replacing c_j by $N(j,c,\delta)$, $0 < \delta < c_j$, does not change the value of any solution $\bar{x}$ with $\bar{x}_j = 0$, whereas the values of all solutions $\bar{x}$ with $\bar{x}_j = 1$ decrease by δ. Hence, if $x_j = 1$, x would remain optimal for every $0 < \delta < c_j$ and this means that $LOW(j,c,x) = c_j > 0$. Therefore $x_j = 0$ must hold. Furthermore, from the key observation and the fact that $LOW(j,c,x) = 0$ it follows that there exists at least one solution x' with $x'_j = 1$ which is optimal with respect to c. Since the value of such a solution decreases by δ, whereas the value of any other solution decreases by at most δ, it now follows that for any $0 < \delta < c_j$, $A(j,c,\delta)$ consists of exactly those solutions x' which are optimal with respect to c and have $x'_j = 1$. □

Lemma 2. *Suppose that $LOW(j,c,x) = 0$ for a certain $j \in \{1,\dots,n\}$. Let $i \neq j$ and $\delta < 0$. If $x_i = 0$, then*

(i) x is optimal with respect to $N(i,c,\delta)$, and

(ii) $LOW(j,N(i,c,\delta),x) = 0$ if and only if there is at least one solution x' which is optimal with respect to c and has $x'_j = 1$ and $x'_i = 0$.

Proof. To prove (i), we note that the values of all solutions $\bar{x}$ with $\bar{x}_i = 0$, remain unchanged when c is replaced by $N(i,c,\delta)$, whereas the values of all solutions $\bar{x}$ with $\bar{x}_i = 1$ increase by $|\delta|$. Hence, x remains optimal.

$LOW(j,c,x) = 0$ means again that there exists at least one solution x' with $x'_j = 1$ which is optimal with respect to c. If c is replaced by $N(i,c,\delta)$, these solutions x' have their value increased by $|\delta|$ if they have $x'_i = 1$, whereas their value remains unchanged if $x'_i = 0$.

To prove (ii), first assume $LOW(j,N(i,c,\delta),x) = 0$. Then there exists at least one solution x'' with $x''_j = 1$ which is optimal with respect to $N(i,c,\delta)$. Since, the value of x does not change when c is replaced by $N(i,c,\delta)$ and the value of every other solution does not decrease, x'' must also be optimal with respect to c. Furthermore, because its value does not increase, $x''_j = 0$ must hold. We now have that, if $LOW(j,N(i,c,\delta),x) = 0$, then $x'_j = 0$ and 1 for at least one solution x' which is optimal with respect to c. On the other hand, if $LOW(j,N(i,c,\delta),x) > 0$ then every solution x' with $x'_j = 1$ which is optimal with respect to c, is no longer optimal when c is replaced by $N(i,c,\delta)$. This implies that each such solution x' has $x'_i = 1$. □

Lemma 3. *Suppose that $LOW(j,c,x) = 0$ for a certain $j \in \{1,\dots,n\}$. Let $i \neq j$ and $0 < \delta < c_i$. If $x_i = 1$, then*

(i) x is optimal with respect to $N(i,c,\delta)$, and

(ii) $LOW(j, N(i, c, \delta), x) = 0$ *if and only if there is at least one solution* x' *which is optimal with respect to* c *and has* $x'_j = 1$ *and* $x'_i = 1$.

Proof. Analogous to the proof of Lemma 2. □

Lemma 4. *Given an optimal solution* x *with respect to* c *and a value* δ, $0 < d < c_j$, *an element of* $A(j, c, \delta)$ *can be found in polynomial time.*

Proof. If $LOW(j,c,x) \geqslant \delta$, then $x \in A(j,c,\delta)$ and we are done. Otherwise, the definition of $LOW(j,c,x)$ implies that $LOW(j, N(i, c, LOW(j,c,x)), x) = 0$. Using Lemma 1, it follows that $x_j = 0$ and that every solution x' which belongs to $A(j,c,\delta)$ has $x'_j = 1$. Furthermore, it suffices to determine some solution x' which is optimal with respect to $N(i, c, LOW(j,c,x))$ and has $x'_j = 1$. We will describe a procedure, based on Lemmas 2 and 3, to construct such a solution x' in polynomial time.

Initially, we set $S := \{j\}$ and $c' := N(i, c, LOW(j,c,x))$. At termination of our procedure, S will contain the indices i for which $x'_i = 1$, where x' is some solution with the desired properties. To determine S, we modify c'. It will always hold trivially that x is optimal with respect to this cost vector. We will also make sure that at least one solution x' with the desired properties remains optimal with respect to c'.

Lemma 2 can be used to determine which indices $i \neq j$ with $x_i = 0$ will appear in S, as follows. Consider these indices one by one in some arbitrary order. If $LOW(j, N(i, c', \delta'), x) = 0$ for some arbitrary $\delta' < 0$, then there exists – among the solutions still under consideration – an optimal solution x' with $x'_i = 0$. In this case we set $c' := N(i, c', \delta')$. Note that this renders any solution x'' with $x''_i = 1$ non-optimal. Therefore, from this point on, we will only consider solutions x' with $x'_i = 0$.

If $LOW(j, N(i, c', \delta'), x) > 0$, then all solutions x' still under consideration must have $x'_i = 1$. Only in this case we add i to S.

We repeat the above until all indices $i \neq j$ with $x_i = 0$ have been considered. Then we consider the indices i with $x_i = 1$ and we use Lemma 3. If $LOW(j, N(i, c', \delta'), x) = 0$ for some arbitrary $0 < \delta' < c_i$, then there exists – among the solutions still under consideration – a solution x' with $x'_i = 1$. In this case we add i to S and set $c' := N(i, c', \delta')$, which means that from now on we will restrict our search to solutions x' with $x'_i = 1$, because solutions with $x'_i = 0$ are no longer optimal. If $LOW(j, N(i, c', \delta'), x) > 0$, then all solutions x' which are still under consideration must have $x'_i = 0$. In this case we do not update c'.

After all indices have been considered, x' is defined as the solution which has exactly the components in S equal to 1. Note that S, and therefore x', may depend on the order in which indices are considered in the above procedure. However, x' found in this way clearly has the desired properties. Furthermore, it is easily seen that the procedure is polynomial. □

We are now able to prove Theorem 1.

Proof of Theorem 1. Let $\bar{c} \in \mathbb{Q}_+^n$ be a given cost vector. We will show that the corresponding problem instance can be solved in polynomial time by solving a sequence of reoptimization problems.

Define $M \equiv 1 + \sum_{i=1}^{n} \bar{c}_i$ and let x be an arbitrary feasible solution with the property mentioned under (a) of the theorem. Initialize the entries of cost vector c' as follows: $c_i' := \bar{c}_i$ if $x_i = 1$ and $c_i' := M$ if $x_i = 0$. Because there is no feasible $x' \neq x$ with $x' \leqslant x$, x is clearly optimal with respect to c'.

Now replace the value of any cost coeffcient c_i' which is equal to M by the value $\bar{c}_i$ and compute a new optimal solution using the polynomial procedure described in the proof of Lemma 4. Repeat this until $c' = \bar{c}$. At this point we have determined a solution which is optimal with respect to $\bar{c}$.

Since the above boils down to executing a polynomial procedure at most n times, the overall running time is polynomially bounded. □

The following theorem states a similar result with respect to increasing cost coefficients.

Theorem 2. *Consider* (P) *for a fixed set $X \subset \{0,1\}^n$. This problem is polynomially solvable for any $c \in \mathbb{Q}_+^n$, if the following two conditions are satisfied.*

(a) *For every problem instance it takes polynomial time to determine a feasible solution $x \in X$ which is minimal.*
(b) *For every cost vector $c' \in \mathbb{Q}_+^n$ and for every optimal solution x of the problem instance defined by c', the maximal value u_i by which the cost coefficient of x_i, $i = 1, \ldots, n$, may be increased such that x remains optimal, can be determined in polynomial time. Here $u_i \equiv \infty$ if x remains optimal for arbitrarily large cost coefficients of x_i.*

Proof. Analogous to the proof of Theorem 1. Given a minimal feasible solution x, initialize the cost vector c' as follows: set $c_i' := \bar{c}_i$ if $x_i = 0$; define $\bar{c}_{\min} \equiv \min\{\bar{c}_i | i = 1, \ldots, n$: $x_i = 0\}$, $\varepsilon \equiv \bar{c}_{\min}/n$ and set $c_i' := \min\{\varepsilon, \bar{c}_i\}$ for all i with $x_i = 1$. Then solution x is optimal with respect to c'. Now, for each i with $c_i' < \bar{c}_i$, increase the value of c_i' to $\bar{c}_i$ and compute a new optimal solution after each change of c'. □

Ramaswamy and Chakravarti [8] have independently obtained results which are quite similar to Theorems 1 and 2. The difference is that we consider the situation in which the cost coefficients may change, but will always remain positive, whereas Ramaswamy and Chakravarti study the case that the cost coefficients are not restricted in sign. (This allows them to prove their results under conditions which are milder than our condition (a).) Hence, these results should be viewed as being complementary, rather than identical.

The following result relates Theorems 1 and 2 to the complexity of the question whether a given solution is still optimal after an arbitrary change of (one or more components of) the cost vector.

Proposition 1. *Suppose that an optimal solution is known for the instance of* (P) *corresponding to a certain cost vector* $\bar{c} \in \mathbb{Q}_+^n$. *If it can be checked in polynomial time whether this solution is also optimal with respect to an arbitrary cost vector* $c' \in \mathbb{Q}_+^n$, *then the values* l_i *and* u_i, $i = 1, \ldots, n$, *as defined in Theorems* 1 *and* 2 *can be determined in polynomial time.*

Proof. The idea is to find the values l_i and u_i, $i = 1, \ldots, n$, by binary search. For details we refer to the proof of Proposition 3 (with $\varepsilon = 0$). $\square$

This proposition implies, for instance, that it is not possible to check in polynomial time whether an optimal TSP tour is still optimal after an arbitrary change of the distance matrix, unless $\mathscr{P} = \mathscr{NP}$.

Remark 1. Results similar to Theorems 1 and 2 and Proposition 1 hold if the objective function of (P) is to be maximized instead of minimized.

Remark 2. Condition (a) in Theorems 1 and 2 is less strong than may seem at first sight. Consider the following well-known formulation of the generalized assignment problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } i = 1, \ldots, m, \\ & \sum_{i=1}^{m} a_{ij} x_{ij} \leqslant b_j \quad \text{for all } j = 1, \ldots, n, \\ & x_{ij} \in \{0, 1\} \quad \text{for all } i = 1, \ldots, m, \ j = 1, \ldots, n. \end{aligned}$$

It is NP-hard to determine a feasible solution for this formulation, and therefore the theorems do not immediately apply. However, by introducing an additional agent which can handle all jobs at very large costs the following suitable formulation (P) is obtained.

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \left(\sum_{j=1}^{n} c_{ij} x_{ij} + M x_{i,n+1} \right) \\ \text{s.t.} \quad & \sum_{j=1}^{n-1} x_{ij} = 1 \quad \text{for all } i = 1, \ldots, m, \\ & \sum_{i=1}^{m} a_{ij} x_{ij} \leqslant b_j \quad \text{for all } j = 1, \ldots, n, \\ & \sum_{i=1}^{m} x_{i,n+1} \leqslant m, \\ & x_{ij} \in \{0, 1\} \quad \text{for all } i = 1, \ldots, m, \ j = 1, \ldots, n+1. \end{aligned}$$

This formulation has a trivial feasible solution that satisfies condition (a) in Theorems 1 and 2. The constant M is chosen to be equal to $\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}+1$, which implies that $x_{i,n+1}=0$ for all $i=1,\dots,m$ in any optimal solution of (P), if the original formulation has a feasible solution. Hence, if both formulations are feasible, they have the same optimal solutions. Note that the size of the two formulations is of the same order. Since (P) has the same structure as the original formulation, polynomial algorithms to compute maximal ranges associated with individually changing cost coefficients of any formulation with this structure, would imply a polynomial algorithm to solve (P), and therefore also the original formulation of the generalized assignment problem.

Remark 3. We have assumed that the only available information is the optimality of a given solution for a particular problem instance. If additional information is available, then it is possible that the values l_i and u_i, $i=1,\dots,n$, can be computed in polynomial time, even if (P) is NP-hard and $\mathscr{P}\neq\mathscr{NP}$. Typically, solution methods for NP-hard problems generate useful information as an inexpensive byproduct. As an extreme example, we can simply use complete enumeration to find an optimal solution and store at the same time for every variable x_i the optimal values under the restrictions $x_i=0$ or $x_i=1$. Subsequently, it is easy to determine l_i and u_i for all $i=1,\dots,n$.

Knowing that it is unlikely that the maximal allowable increases and decreases of the cost coefficients can be determined exactly in polynomial time, a natural question that arises is whether it is possible to calculate reasonable approximations of these values in polynomial time. In particular, we are interested in underestimates that are relatively close to the true values. We would then obtain for every cost coefficient a range in which it can be varied individually without affecting the optimality of the solution at hand. These are not necessarily the maximal ranges, but hopefully they are not too conservative. Therefore, one would like to have some guarantee that the approximations are reasonable. For instance, this is the case if the estimate is known to be at least $(1-\varepsilon)$ times the true value for some ε, $0<\varepsilon<1$. However, we have the following result.

Proposition 2. *Let $\bar{c}\in\mathbb{Q}_+^n$ be an arbitrary cost vector. Consider an optimal solution with respect to the cost vector $\bar{c}$ and let $u_i<\infty$ be the maximal allowable increase of $\bar{c}_i$, $i\in\{1,\dots,n\}$. If it is possible to compute in polynomial time a value $\tilde{u}_i$ such that $(1-\varepsilon)u_i\leqslant\tilde{u}_i\leqslant u_i$, for some $\varepsilon\in\mathbb{Q}$, $0<\varepsilon<1$, then u_i can be determined in polynomial time.*

Proof. Without loss of generality, we may assume that $\bar{c}\in\mathbb{N}_+^n$, i.e., all cost coefficients are positive integers. Then all solutions have an integer value and this implies that $u_i\in\mathbb{N}$. Let $\bar{c}^1\equiv\bar{c}$ and $\tilde{u}_i^1\equiv\tilde{u}_i$. For $k>1$ we define $\bar{c}^k\in\mathbb{Q}_+^n$ and $\tilde{u}_i^k$, $k\geqslant 1$, recursively as follows:

$$\bar{c}_i^k\equiv\bar{c}_i^{k-1}+\tilde{u}_i^{k-1},$$
$$\bar{c}_j^k=\bar{c}_j \quad \text{if } j\neq i,$$

and $\tilde{u}_i^k$ is the approximation of the maximal allowable increase of cost coefficient $\bar{c}_i^k$ which is calculated analogously to $\tilde{u}_i$ with respect to $\bar{c}^k$ and the original optimal solution.

Hence, we are considering a sequence of cost vectors for which only the ith entry is changing. Note that the original solution remains optimal, because the approximations are underestimates of the maximal allowable increases.

Let us define $c_i^* \equiv \bar{c}_i + u_i$, then $c_i^* \in \mathbb{N}$ and $\tilde{u}_i^k \geqslant (1-\varepsilon)(c_i^* - \bar{c}_i^k)$ for all $k \geqslant 1$. Using induction it is easy to verify that $c_i^* - \bar{c}_i^k \leqslant \varepsilon^{k-1} u_i$ for all $k \geqslant 1$. Therefore, $c_i^* - \bar{c}_i^k < 1$ if $\varepsilon^{k-1} u_i < 1$ or, equivalently, $(1/\varepsilon)^{1-k_1} u_i < 1$. The latter holds for all $k > {}^{1/\varepsilon}\log u_i$. (Note that $1/\varepsilon > 1$.)

Because $c_i^* \in \mathbb{N}$, it is easy to see that $c_i^* - \bar{c}_i^k < 1$ implies $c_i^* = \lceil \bar{c}_i^k \rceil$. If $u_i < \infty$, then clearly $u_i \leqslant \sum_{j=1}^{n} \bar{c}_j$. Hence, c_i^* is found after calculating $O({}^{1/\varepsilon}\log u_i) = O(\log(\sum_{j=1}^{n} \bar{c}_j))$ times an approximation of an allowable increase. If the latter calculations can be done in polynomial time, a polynomial method to calculate $u_i = c_i^* - \bar{c}_i$ results. □

Remark 4. A similar result holds with respect to maximal allowable decreases.

2. Postoptimality analysis of ε-optimal solutions

Consider a binary program of the following form:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & x \in X \subset \{0,1\}^n \end{aligned} \tag{P}$$

with $c \in \mathbb{Q}^n_{\geqslant 0}$. Note that, contrary to the preceding section, we now allow zero cost coefficients.

We will prove two results with respect to (P), which can be used to show that, unless $\mathscr{P} = \mathscr{NP}$, several sensitivity questions related to ε-optimal heuristics for NP-hard problems cannot be answered by polynomial algorithms. For instance, we will be able to conclude that existence of a polynomial algorithm to determine, for any cost coefficient of a min-knapsack problem, the maximal increase such that an ε-optimal solution maintains this property, would imply $\mathscr{P} = \mathscr{NP}$. (As before, we may only draw such conclusions if the NP-hard problem can be formulated in polynomial time as a suitable 0/1 program, but again this is the case for many well-known NP-hard problems.)

As another example, suppose that an ε-optimal tour has been obtained, for an instance, of the traveling salesman problem which obeys the triangle inequality. We will be able to conclude that it is unlikely that there exists a polynomial algorithm to determine whether after a change of the distance matrix (not necessarily maintaining the triangle inequality) the tour is still ε-optimal. Similar results can be derived for other NP-hard problems (see also Remark 5 after Theorem 3).

Theorem 3. *Suppose that H is a polynomial ε-approximation algorithm ($\varepsilon \in \mathbb{Q}_+$) for* (P) *that has been applied to the instance corresponding to an arbitrary cost vector $\bar{c} \in \mathbb{Q}^n_{\geqslant 0}$. Let u_i, $i = 1, \ldots, n$, be the maximal value by which $\bar{c}_i$ can be increased such that the heuristic solution remains ε-optimal. If u_i can be determined in polynomial time for all $i = 1, \ldots, n$, then the optimal value of the problem instance can be determined in polynomial time.*

Proof. Let z^* and z^H denote, respectively, the value of the optimal and heuristic solution. Because H is ε-optimal it holds that $z^H \leqslant (1+\varepsilon)z^*$. We will show that once the values u_i, $i = 1, \ldots, n$, have been calculated it is possible to calculate z^* after a polynomial number of additional operations.

For every $S \subseteq \{1, \ldots, n\}$ we define $z_0(S)$ as the optimal value under the condition that $x_i = 0$ for all $i \in S$, and analogously we let $z_1(S)$ denote the optimal value under the condition that $x_i = 1$ for all $i \in S$. Furthermore, define

$$X_1 \equiv \{i \mid 1 \leqslant i \leqslant n \text{ and } x_i = 1 \text{ in the heuristic solution}\}$$

and

$$\bar{X}_1 \equiv \{i \in X_1 \mid u_i = \infty\}.$$

Suppose $i \in X_1$, then increasing $\bar{c}_i$ will increase the value of the heuristic solution, whereas the value of any feasible solution with $x_i = 0$ will remain constant. Hence, if there exists a feasible solution with $x_i = 0$, then the heuristic solution can not remain ε-optimal when $\bar{c}_i$ is increased by arbitrarily large values. It is now easy to see that $\bar{X}_1$ is the set of variables that are equal to 1 in every feasible solution. Thus, if $X_1 = \bar{X}_1$ then it follows from the non-negativity of the cost coefficients that $z^* = z^H$.

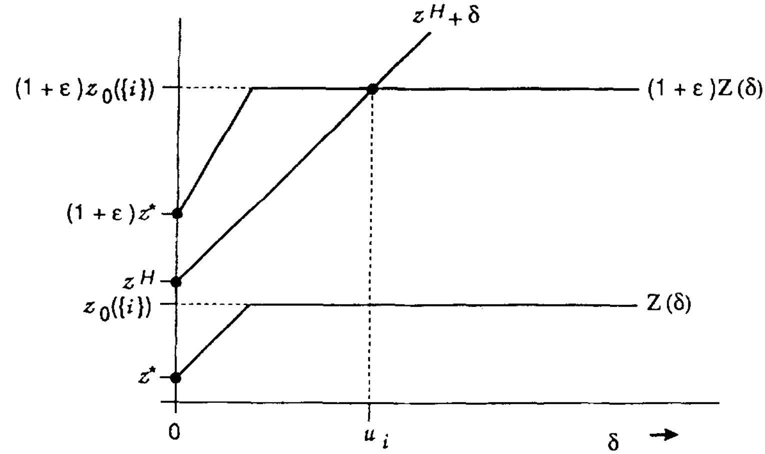
Now suppose that $\bar{X}_1 \neq \bar{X}_1$ and $i \in X_1 \backslash \bar{X}_1$. Let $Z(\delta)$ denote the optimal value of the problem instance that is obtained if $\bar{c}_i$ is increased by $\delta \geqslant 0$. Hence, $Z(0) = z^*$ and on $[0, \infty)$ the function Z is either constant or linear with slope 1 up to a certain value of δ and constant afterwards. If $\bar{c}_i$ is increased by u_i, then the value of the heuristic solution becomes equal to $z^H + u_i$. From the definition of u_i it follows that $z^H + u_i = (1+\varepsilon)Z(u_i)$ (see Figs. 1 and 2). Moreover, if $\delta = u_i$ then $x_i = 0$ in an optimal solution. Hence, $Z(u_i) = z_0(\{i\})$ and therefore $z^H + u_i = (1+\varepsilon)z_0(\{i\})$. It follows that $z_0(\{i\})$ can be easily calculated for all $i \in X_1 \backslash \bar{X}_1$.

In an optimal solution of the original problem instance either $x_i = 1$ for all $i \in X_1 \backslash \bar{X}_1$ or $x_i = 0$ for at least one $i \in X_1 \backslash \bar{X}_1$. Therefore, we have the following equality:

$$z^* = \min[z_1(X_1 \backslash \bar{X}_1), \min\{z_0(\{i\}) \mid i \in X_1 \backslash \bar{X}_1\}].$$

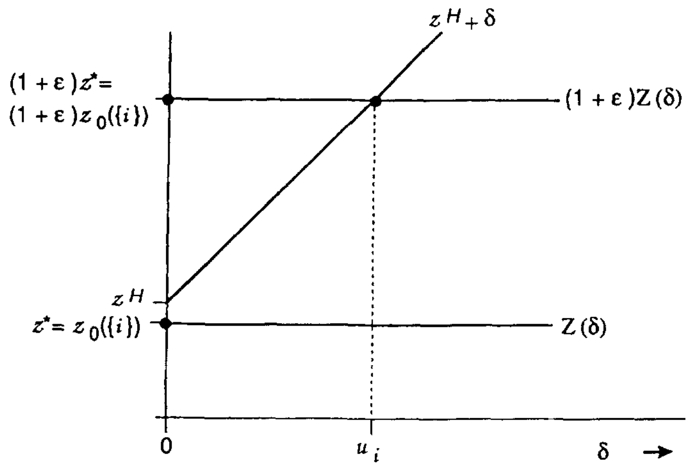
Finally, note that $z_1(X_1 \backslash \bar{X}_1) = z_1(X_1)$ and $z_1(X_1) = z^H$ because of the non-negativity of the cost coefficients. Therefore, z^* can now easily be calculated. $\square$

Remark 5. If the objective function of (P) is to be maximized instead of minimized, then a similar result holds with respect to maximal allowable decreases of objective coefficients.



Case A: 0 *lies in the interval on which* $Z(\delta)$ *is strictly increasing*

Fig. 1.



Case B: 0 *lies in the interval on which* $Z(\delta)$ *is constant*

Fig. 2.

Proposition 3. *Suppose that H is a polynomial* ε*-approximation algorithm* ($\varepsilon \in \mathbb{Q}_+$) *for* (P) *that has been applied to the instance corresponding to an arbitrary cost vector* $\bar{c} \in \mathbb{Q}^n_{\geqslant 0}$. *If it can be checked in polynomial time whether the heuristic solution is also* ε*-optimal with respect to another arbitrary cost vector* $c' \in \mathbb{Q}^n_{\geqslant 0}$, *then the optimal value of the problem instance can be determined in polynomial time.*

Proof. We use Theorem 3 and its proof. It suffices to show that the values u_i, $i = 1, \ldots, n$, can be calculated in polynomial time for all $i \in X_1$ if there exists a polynomial algorithm to check ε-optimality of the heuristic solution. The idea is to use this algorithm in a binary search for u_i, $i \in X_1$.

First note that we may assume that $\bar{c} \in \mathbb{N}^n$ and $\varepsilon\bar{c} \in \mathbb{N}^n$. This implies that if $u_i < \infty$, then $u_i \in \mathbb{N}$.

Suppose $\bar{c}_i$, $i \in X_1$, is increased to a value greater than $(1+\varepsilon)\sum_{j=1}^n \bar{c}_j$, then the value of the heuristic solution also becomes greater than this value. Therefore, the heuristic solution can only stay ε-optimal if the optimal solution value is greater than $\sum_{j=1}^n \bar{c}_j$. Clearly, every feasible solution with $x_i = 0$ will have a value at most $\sum_{j=1}^n \bar{c}_j$ and if such a solution exists, then $u_i < \infty$. We conclude that $u_i = \infty$ if and only if the heuristic solution stays ε-optimal and by assumption this can be checked in polynomial time.

The above implies that $u_i < \infty$ is equivalent to $0 \leqslant u_i \leqslant (1+\varepsilon)\sum_{j=1}^n \bar{c}_j$. In this case the exact (integer) value of u_i can be found in polynomial time by a binary search among the integers in this range, where in each iteration ε-optimality of the heuristic solution is checked. □

Remark 6. Note that ε in Proposition 3 may depend on the size of the problem instance, but not on the values of the cost coefficients.

3. Concluding remarks

We think that the results in this paper are particularly interesting because of their generality. Many well-known NP-hard optimization problems can be put in the form to which the results apply. Note, however, that we have only considered the cost coefficients of the 0/1 formulation. For instance, although many min–max problems can be formulated as 0/1 problems with a linear objective function, viz., as the minimization of a single variable, our results are clearly not relevant for those problems. For complexity results on min–max problems we refer to Ramaswamy and Charkravarti [8].

The kind of postoptimality analysis considered in this paper corresponds to the classical way of performing sensitivity analysis in linear programming: only one cost coefficient is assumed to change, the other coefficients remain fixed. Of course, one may also be interested in simultaneous changes. For instance, for linear programming Wendell [12] propounds the so-called *tolerance approach* which allows for such changes. However, given our results, we do not expect that a similar approach to NP-hard 0/1 problems leads to subproblems that are polynomially solvable, even if ε-optimal solutions are considered instead of optimal ones.

The results in this paper can be viewed as being negative, because they state that certain polynomial algorithms are unlikely to exist. On the other hand, Ramaswamy and Charkravarti [8] show that if we are dealing with a polynomially solvable problem (P), then it is always possible to compute the maximal ranges of each individual cost coefficient in polynomial time. Recently, Chakravarti and Wagelmans [1] have generalized this result to the calculation of the stability radius of a solution, which is a measure for maximal simultaneous changes of cost coefficients. They also discuss a further generalization to Wendell's tolerance approach.

Acknowledgements

The authors would like to thank Dr. Charles Blair from the University of Illinois and two anonymous referees, whose comments on an earlier version of this paper have been extremely helpful in improving the presentation of the results. Part of this research was carried out while the second author was visiting the Operations Research Center at the Massachusetts Institute of Technology with financial support of the Netherlands Organization for Scientific Research (NWO). He would like to thank the students, staff and faculty affiliated with the ORC for their kind hospitality. Special thanks to Kerry M. Malone, without whom writing this paper would have been so much easier.

References

[1] N. Chakravarti, A.P.M. Wagelmans, Calculation of stability radii for combinatorial optimization problems, Oper. Res. Lett. 23 (1999) 1–7.

[2] T. Gal, Postoptimal Analysis, Parametric Programming and Related Topics, McGraw-Hill, New York, 1979.

[3] E.N. Gordeev, V.K. Leontev, I.Kh. Sigal, Computational algorithms for finding the radius of stability in selection problems, USSR Comput. Math. Math. Phys. 23 (1983) 973–979.

[4] H. Greenberg, An annotated bibliography for post-solution analysis in mixed integer and combinatorial optimization, in: D.L. Woodruff (Ed.), Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search, Kluwer Academic Publishers, Dordrecht, 1998, pp. 97–148.

[5] D.S. Johnson, C.H. Papadimitriou, Computational complexity, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds.), The Traveling Salesman Problem, Wiley, New York, 1985, pp. 37–85.

[6] S.A. Kravchenko, Yu.N. Sotskov, F. Werner, Optimal schedules with infinitely large stability radius, Optimization 33 (1995) 271–280.

[7] M. Libura, E.S. Van der Poort, G. Sierksma, J.A.A. Van der Veen, Stability aspects of the traveling salesman problem based on the k-best solutions, Discrete Appl. Math. (1999), to appear.

[8] R. Ramaswamy, N. Chakravarti, Complexity of determining exact tolerances for min-sum and min–max combinatorial optimization problems, Working Paper WPS-247/95, Indian Institute of Management, Calcutta, India, 1995, pp. 34.

[9] Yu.N. Sotskov, The stability of the approximate boolean minimization of a linear form, USSR Comput. Math. Math. Phys. 33 (1993) 699–707.

[10] Yu.N. Sotskov, V.K. Leontev, E.N. Gordeev, Some concepts of stability analysis in combinatorial optimization, Discrete Appl. Math. 58 (1995) 169–190.

[11] Yu.N. Sotskov, A.P.M. Wagelmans, F. Werner, On the calculation of the stability radius of an optimal or an approximate schedule, Ann. Oper. Res. 83 (1998) 213–252.

[12] R.E. Wendell, The tolerance approach to sensitivity analysis in linear programming, Management Sci. 31 (1985) 564–578.