On the Complexity of Postoptimality Analysis of 0/1 Programs

Stan Van Hoesel and Albert Wagelmans

OR 259-91 September 1991

ON THE COMPLEXITY OF POSTOPTIMALITY ANALYSIS OF 0/1 PROGRAMS

Stan Van Hoesel[1]
Albert Wagelmans[2]

September 1991

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 can not be performed by polynomial methods, unless P = 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 $\mathcal{P} = \mathcal{NP}$.

OR/MS subject classification: Analysis of algorithms, computational complexity: postoptimality analysis of 0/1 programs; Analysis of algorithms, suboptimal algorithm: sensitivity analysis of approximate solutions of 0/1 programs

1) Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

2) Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands; currently on leave at the Operations Research Center, Room E 40-164, Massachusetts Institute of Technology, Cambridge, MA 02139; financial support of the Netherlands Organization for Scientific Research (NWO) is gratefully acknowledged.

0 Introduction

Whereas sensitivity analysis is a well-established topic in linear programming (see Gal, 1979, for a comprehensive review), its counterpart in integer programming is a much less developed research area (reviews are given by Geoffrion and Nauss, 1977, and Jenkins, 1990). Maybe more surprisingly, hardly any attention has been paid to sensitivity analysis of specific NP-hard problems, neither from an algorithmic nor from a theoretical point of view. The results in this paper are of a theoretical nature and relate to many well-known NP-hard problems.

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. In Section 1 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 can not be performed by polynomial methods, unless $\mathcal{P} = \mathcal{NP}$. 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, Section 2 deals with the complexity 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 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 $\mathcal{P} = \mathcal{NP}$.

Several concluding remarks are given in Section 3.

1 Postoptimality analysis of optimal solutions

Consider an optimization problem of the following form

$$\min \sum_{i=1}^{n} c_i x_i$$

s.t.

$$x \in X \subset \mathbb{R}^n \qquad (P)$$

$$x_i \in \{0,1\} \quad \text{for all } i=1,\ldots,n$$

with $c \in \mathbb{Q}_+^n$.

We will first prove three propositions with respect to (P) and then discuss their implications. In the first proposition we consider decreasing cost coefficients.

Proposition 1 (P) is polynomially solvable if

(a) for every instance of (P) it takes polynomial time to determine a feasible solution $x \in \mathbb{R}^n$ such that every $x' \in \mathbb{R}^n$ with $x' \neq x$ and $x' \leq x$, is infeasible, and

(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.

Proof 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. We start with an arbitrary feasible solution and define a cost vector $c' \in \mathbb{Q}_+^n$, $c' \geq \bar{c}$, such that the solution is optimal with respect to c'. Then we modify c' systematically until further changes will render the current solution non-optimal. We will show how to determine another feasible solution that is optimal if the intended modification of c' is actually carried out and continue in this way. The vector c' will be monotonically non-increasing, and we terminate as soon as $c' = \bar{c}$.

We define $M \equiv 1 + \sum_{i=1}^{n} \bar{c}_i$ and for every n-dimensional 0/1-vector x we let

$$I_1(x) \equiv \{i \mid 1 \leq i \leq n \text{ and } x_i = 1\}$$

and

$$I_0(x) \equiv \{i \mid 1 \leq i \leq n \text{ and } x_i = 0\}$$

Furthermore, assume that a polynomial procedure $LOW_i(c,x)$ calculates l_i, $i \in \{1,...,n\}$, as defined under (b) of the proposition with respect to the cost vector c and a given corresponding optimal solution x.

The initial vector c' is constructed as follows.

<u>Initialization</u> Let x be an arbitrary feasible solution with the property mentioned under (a) of the proposition. Set the entries of vector c' as follows:

$$c_i' := \bar{c}_i \quad \text{for all } i \in I_1(x)$$

and

$$c_i' := M \quad \text{for all } i \in I_0(x)$$

Because there is no feasible $x' \neq x$ with $x' \leq x$, x is clearly optimal with respect to c'.

The initialization step is followed by a number of steps which we call the *major iterations*. In each major iterations one entry of the vector c' is decreased and, if necessary, a new optimal solution is determined. Define $I_M \equiv \{i \mid 1 \leq i \leq n \text{ and } c_i' = M\}$. In the major iterations it will always hold that $c_i' = \bar{c}_i$ for all $i \notin I_M$. Note that when entries of c' are decreased, the value of every feasible solution does not increase. In particular this means that the optimal value is is non-increasing. Because we start with a vector c' that has an optimal solution with value less than M (see the initialization step), it follows that in every major iteration $i \in I_M$ implies $x_i = 0$, where x is the optimal solution at the end of that iteration. When the major iterations stop, it holds that $c' = \bar{c}$. Hence, the current optimal solution solves problem (P) for cost vector $\bar{c}$.

<u>Major iterations</u> We are given c' and a corresponding optimal solution x. Furthermore, it holds that $I_M \subseteq I_o(x)$. Pick any $j \in I_M$ and execute $LOW_j(c',x)$. If $c_j' - l_j \leq \bar{c}_j$, then x remains optimal if c_j' is set to $\bar{c}_j$. If $c_j' - l_j > \bar{c}_j$, then determine (by the procedure described below) a new feasible solution x' that is optimal when c_j' is set to $\bar{c}_j$; set $x := x'$. In any case set $c_j' := \bar{c}_j$ and delete j from I_M. Stop if $I_M = \emptyset$; otherwise, repeat.

Suppose $c'_j - l_j > \bar{c}_j$ for some $j \in \{1,\ldots,n\}$. We will show how to determine in this case a feasible solution x' that it is optimal after c'_j has been decreased by more than l_j. To facilitate the exposition we consider the optimal value as a function of c_j, while all other cost coefficients are fixed. Assume for the moment that $c_i = c'_i$ for all $i \neq j$. The following observations are crucial for the correctness of the procedure that we are going to describe. If c_j is decreased, then the value of every solution $\bar{x}$ with $\bar{x}_j = 0$ does not change, while the value of every solution $\bar{x}$ with $\bar{x}_j = 1$ decreases by the same amount as c_j does (see Figure 1). Therefore, any solution x' that is optimal if $c_j < c'_j - l_j$ must have $x'_j = 1$. Because x' is optimal on the interval $(0, c'_j - l_j)$, it is certainly optimal if for $c_j = \bar{c}_j$.

Also note that changing c_i, $i \neq j$, will have no effect on the value of a solution $\bar{x}$ as function of c_j if $\bar{x}_i = 0$; however, if $\bar{x}_i = 1$ then the value function will shift horizontally by the same amount as the change of c_i (see Figure 2).

INSERT FIGURES 1 AND 2 HERE

<u>Determination of a new optimal solution</u>

We are given a solution x that is optimal with respect to c' and an index j such that $c'_j - l_j > \bar{c}_j$ and $j \in I_M \subseteq I_0(x)$. Note that x is still optimal for $c_j = c'_j - l_j$. The procedure that we are going to describe finds a feasible solution x' with $x'_j = 1$ that is optimal for $c_j = c'_j - l_j$. It determines the elements of the set $I_1(x')$ for some solution x' with the desired properties. Clearly, $j \in I_1(x')$ and $i \notin I_1(x')$ for all $i \in I_M$.

Initialize $I_1 := \{j\}$; this set will eventually become equal to $I_1(x')$. Furthermore, set $c''_i := c'_i$, for all $i \neq j$, and $c''_j := c'_j - l_j$. Note that $LOW_j(c'', x)$ will output 0.

To determine a solution x' with the desired property we modify c''. It will always hold trivially that x is optimal with respect to this cost vector. For x' we will accomplish the same, because if at any point a modification of c'' is such that all solutions x' with the desired property turn out to become non-optimal, then the change will be made undone.

First we determine which elements of $I_0(x) \setminus I_M$ will appear in $I_1(x')$. To this end we carry out the so-called *minor iterations of type* 1.

Minor iterations type 1 Pick any $k \in I_0(x) \backslash I_M$ that has not yet been considered and set $c''_k := M$.

If the output of $LOW_j(c'',x)$ is now positive, then every optimal solution with $x'_i = 1$ for all $i \in I_1$ must also have $x'_k = 1$ (see Figure 3a: the value functions of all solutions with the desired property have shifted). In this case we reset $c''_k := c'_k$ and add k to I_1, because we know now that, given earlier choices of variable values, we are searching for solutions x' that must have $x'_k = 1$.

If on the other hand the output of $LOW_j(c'',x)$ is still 0, then there is still an optimal solution x' with $x'_i = 1$ for all $i \in I_1$ and $x'_k = 0$ (see Figure 3b: there is at least one solution with the desired property for which the value function has not shifted). In this case we maintain the change in c'', which means that from now on we restrict our search to solutions x' with $x'_k = 0$.

Repeat unless all elements of $I_0(x) \backslash I_M$ have been considered.

INSERT FIGURES 3A AND 3B HERE

After the minor iterations of type 1 it holds that $I_1 = I_0(x) \cap I_1(x')$. To determine the indices $i \in I_1(x) \cap I_1(x')$ we carry out the so-called *minor iterations of type* 2.

Minor iterations type 2 Pick any $k \in I_1(x)$ that has not yet been considered and set $c''_k := \frac{1}{2} c''_k$. This decreases the value of the optimal solution x by $\frac{1}{2} c''_k$.

If the output of $LOW_j(c'',x)$ is now a positive number, then every solution x' with $x'_i = 1$ for all $i \in I_1$ has $x'_k = 0$ (see Figure 4a: the value functions of all solutions with the desired property remain the same). In this case we reset $c''_k := c'_k$.

If on the other hand the output of $LOW_j(c'',x)$ is still 0, then there exists a solution x' with $x'_i = 1$ for all $i \in I_1$ and $x'_k = 1$ (see Figure 4b: there is at least one solution with the desired property for which the optimal value function has also shifted). In this case we add k to I_1 and maintain the change in c'', which means that from now on we will restrict our search to solutions x' with $x'_k = 1$.

Repeat unless all elements of $I_1(x)$ have been considered.

INSERT FIGURES 4A AND 4B HERE

The solution x' is now defined by $I_1(x'):=I_1$. Note that I_1, and therefore x', may depend on the order in which indices are considered in the above procedures. However, x' found in this way clearly has the desired properties.

As far as the total complexity of calculating an optimal solution for the problem instance with cost vector $\bar{c}$ is concerned, we first note that under assumption (a) of the proposition the initialization step takes polynomial time. Furthermore, it can easily be verified that each of the procedures LOW_i, $i=1,\ldots,n$, is executed at most n times. Hence, under assumption (b) of the proposition the major iterations can be carried out in polynomial time. This completes the proof. □

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

***Proposition* 2** (P) is polynomially solvable if

(a) for every instance of (P) it takes polynomial time to determine a feasible solution $x\in\mathbb{R}^n$ such that every $x'\in\mathbb{R}^n$ with $x'\neq x$ and $x'\leq x$, is infeasible, and

(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 of Proposition 1. For the initial feasible solution x, we define the cost vector c' for which x is optimal as follows: set $c'_i:=c_i$ for all $i\in I_0(x)$; define $c_{min}\equiv\min\{c_i|i\in I_0(x)\}$, $\varepsilon\equiv c_{min}/|I_1(x)|$ and set $c'_i:=\min\{\varepsilon,c_i\}$ for all $i\in I_1(x)$. In the major iterations we pick any $j\in I_1(x)$ for which $c'_j<c_j$ and check whether c'_j can be increased to c_j without rendering the solution x non-optimal. If this is not the case then we determine a new optimal solution x' with $x'_j=0$, using a polynomial procedure that calculates u_j

as a subroutine. □

The following proposition relates the preceding results to the complexity of the question whether a given solution is still optimal after an arbitrary change of the cost vector.

***Proposition* 3** Suppose that an optimal solution is known for the instance of (P) corresponding to an arbitrary 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 another arbitrary cost vector $c' \in \mathbb{Q}_+^n$, then the values l_i and u_i, $i=1,\dots,n$, as defined in Propositions 1 and 2 can be determined in polynomial time.

Proof The idea is to find the values l_i and u_i, $i=1,\dots,n$, by binary search. For details we refer to the proof of Proposition 6 (with $\varepsilon = 0$). □

***Remark* 1** Results similar to Propositions 1, 2 and 3 hold if the objective function of (P) is to be maximized instead of minimized.

The three propositions above have implications for many well-known NP-hard problems. For instance, we are able to conclude that, unless $\mathcal{P} = \mathcal{NP}$, it is impossible to determine in polynomial time the maximal ranges in which the distances of a traveling salesman problem can vary individually without affecting the optimality of a given tour. A similar conclusion can be drawn with respect to checking whether an optimal tour is still optimal after an arbitrary change of the distances. Note that we may only draw such conclusions if the NP-hard problem can be formulated in polynomial time as a suitable 0/1 program.

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

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s.t.

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \text{for all } i = 1, \ldots, m$$

$$\sum_{i=1}^{m} a_{ij} x_{ij} \le b_j \qquad \text{for all } j = 1, \ldots, n$$

$$x_{ij} \in \{0,1\} \qquad \text{for all } i = 1, \ldots, m,\ j = 1, \ldots, n$$

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

$$\min \sum_{i=1}^{m} \left[\sum_{j=1}^{n} c_{ij} x_{ij} + M x_{i,n+1} \right]$$

s.t.

$$\sum_{j=1}^{n+1} x_{ij} = 1 \qquad \text{for all } i = 1, \ldots, m$$

$$\sum_{i=1}^{m} a_{ij} x_{ij} \le b_j \qquad \text{for all } j = 1, \ldots, n \qquad (P)$$

$$\sum_{i=1}^{m} x_{i,n+1} \le m$$

$$x_{ij} \in \{0,1\} \qquad \text{for all } i = 1, \ldots, m,\ j = 1, \ldots, n+1$$

This formulation has a trivial feasible solution that satisfies condition (a) in the first two propositions. The constant M should be chosen such that in case the first formulation has a feasible solution, then $x_{i,n+1} = 0$ for all $i = 1, \ldots, m$, in any optimal solution of formulation (P).

***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, $i = 1, \ldots, n$, can be computed in polynomial time, even if (P) is NP-hard and $\mathcal{P} \neq \mathcal{NP}$. 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 $x_i = 0$

respectively $x_i = 1$. Subsequently, it is easy to determine l_i and u_i for all $i = 1, \ldots, n$.

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* 4** 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 be the maximal allowable increase of $\bar{c}_i$, $i \in \{1, \ldots, n\}$. If it is possible to compute in polynomial time a value $\tilde{u}_i$ such that $(1-\varepsilon)u_i \leq \tilde{u}_i \leq u_i$, 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 $\bar{c} \in \mathbb{N}_+^n$. 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 $\tilde{u}_i^k$, $k \geq 1$, recursively as follows:

$\bar{c}_i^k \equiv \bar{c}_i^{k-1} + \tilde{u}_i^{k-1}$,

$\bar{c}_j^k = \bar{c}_j$ 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 i-th 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 $\tilde{u}_i^k \geq (1-\varepsilon)(c_i^* - \bar{c}_i^k)$ for all $k \geq 1$. Using induction it is easy to verify that $c_i^* - \bar{c}_i^k \leq \varepsilon^{k-1} u_i$ for all $k \geq 1$. Therefore, $c_i^* - \bar{c}_i^k < 1$ for all $k > {}^{1/\varepsilon}\log u_i$. 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 $u_i \leq \sum_{j=1}^n \bar{c}_j$. Hence, c_i^* is found after calculating $O(\log(\sum_{j=1}^n \bar{c}_j))$ 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 an optimization problem which can be formulated in polynomial time as a binary program of the following form

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

with $c \in \mathbb{Q}_{\geq 0}^n$.

We will prove two propositions with respect to (P), which can be used to show that, unless $\mathcal{P} = \mathcal{NP}$, 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 all cost coefficients of a min-knapsack problem, the maximal increase such that an ε-optimal solution maintains this property, would imply $\mathcal{P} = \mathcal{NP}$.

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 Proposition 5).

Proposition 5 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 $\bar{c} \in \mathbb{Q}^n_{\geq 0}$. 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 $z^H \leq (1+\varepsilon)z^*$. 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

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

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 $\bar{X}_1 \neq X_1$ 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 $\delta \geq 0$. 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 Figure 5). 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 $i \in X_1 \backslash \bar{X}_1$.

INSERT FIGURE 5 HERE

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 $i\in X_1\backslash\bar{X}_1$. This is equivalent to the following statement:

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

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. □

***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.

***Proposition* 6** 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 $\bar{c}\in\mathbb{N}^n$. If it can be checked in polynomial time whether the heuristic solution is also ε-optimal with respect to another arbitrary cost vector $c'\in\mathbb{N}^n$, then the optimal value of the problem instance can be determined in polynomial time.

Proof We use Proposition 5 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 $\varepsilon\bar{c}_i\in\mathbb{N}$ for all $i\in X_1$. 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 $0\le u_i\le(1+\varepsilon)\sum_{j=1}^{n}\bar{c}_j$. In this case the exact 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 6 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. 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, the results are clearly not relevant for those problems. It seems that establishing similar complexity results calls for a much more problem specific approaches than those used in this paper.

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 (1985) propagates 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.

Acknowledgement

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.

References

Gal, T. (1979), *Postoptimal Analysis, Parametric Programming and Related Topics*, McGraw-Hill, New York

Geoffrion, A.M. and R. Nauss (1977), "Parametric and Postoptimality Analysis in Integer Linear Programming", *Management Science* 23, 453 – 466

Jenkins, L. (1990), "Parametric Methods in Integer Programming", *Annals of Operations Research* 27

Wendell, R.E. (1985), "The Tolerance Approach to Sensitivity Analysis in Linear Programming", *Management Science* 31, 564–578

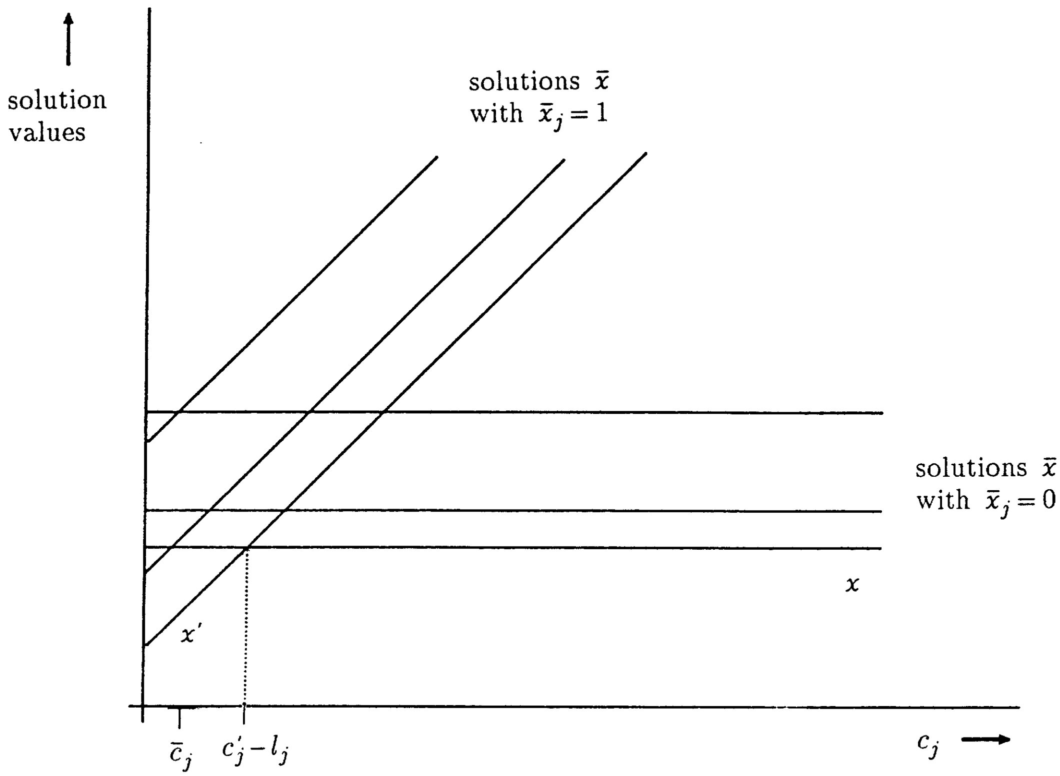


Figure 1

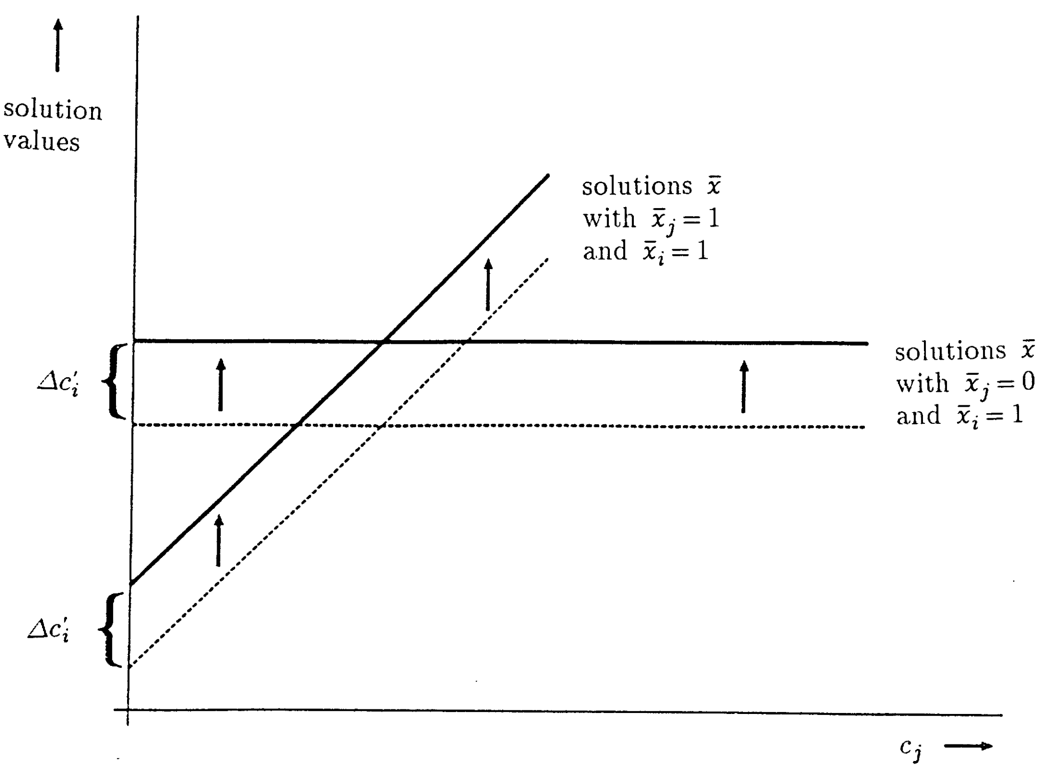


Figure 2

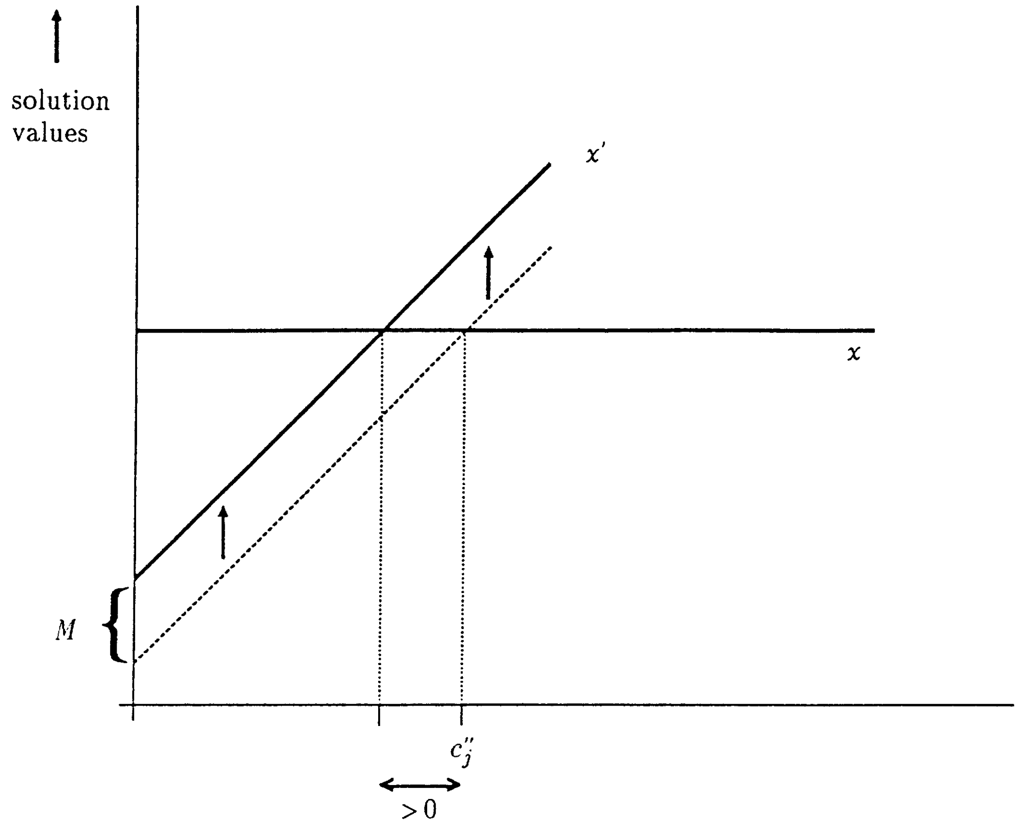


Figure 3a

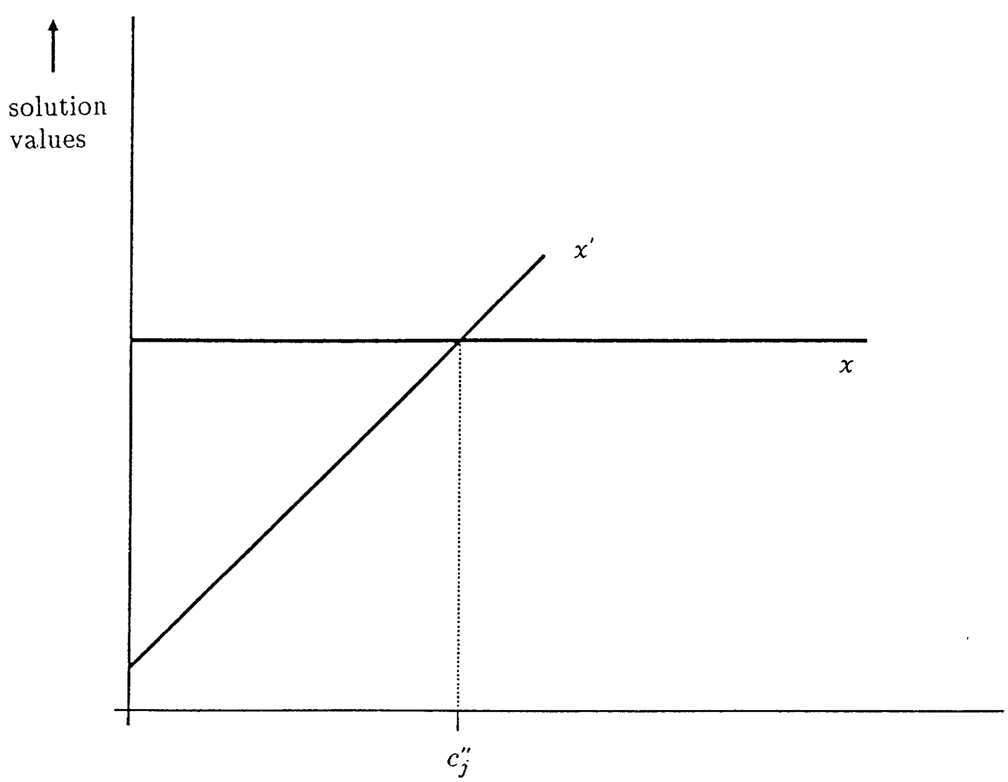


Figure 3b

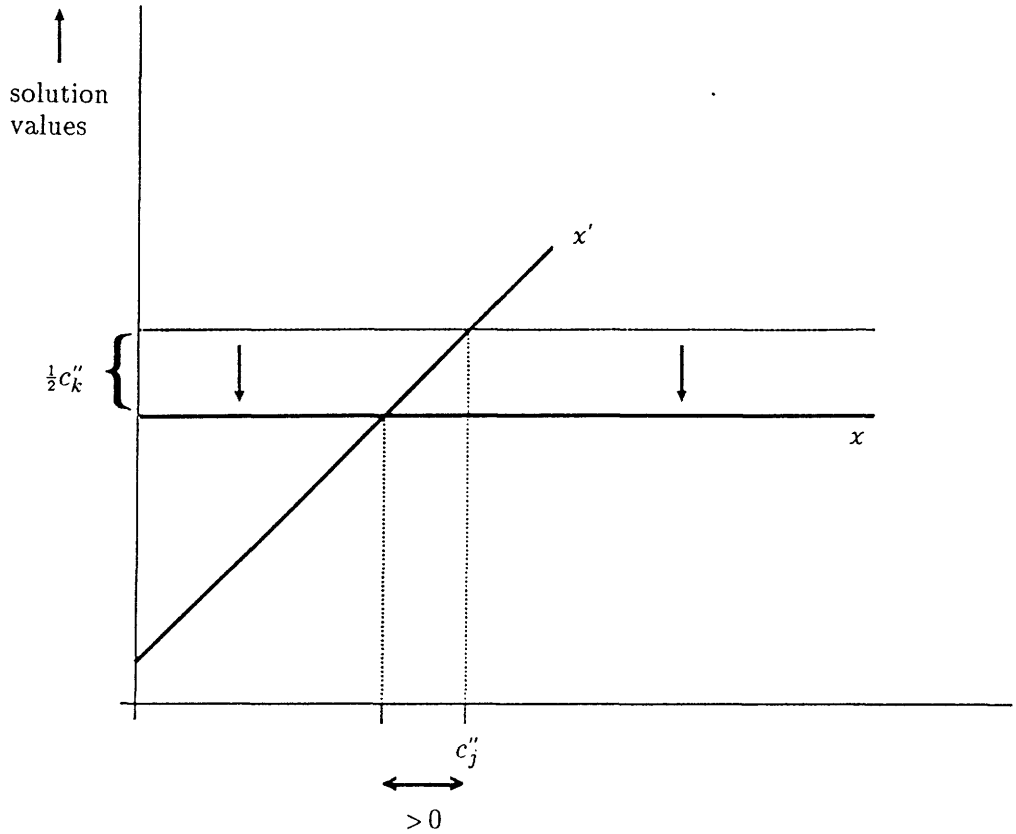


Figure 4a

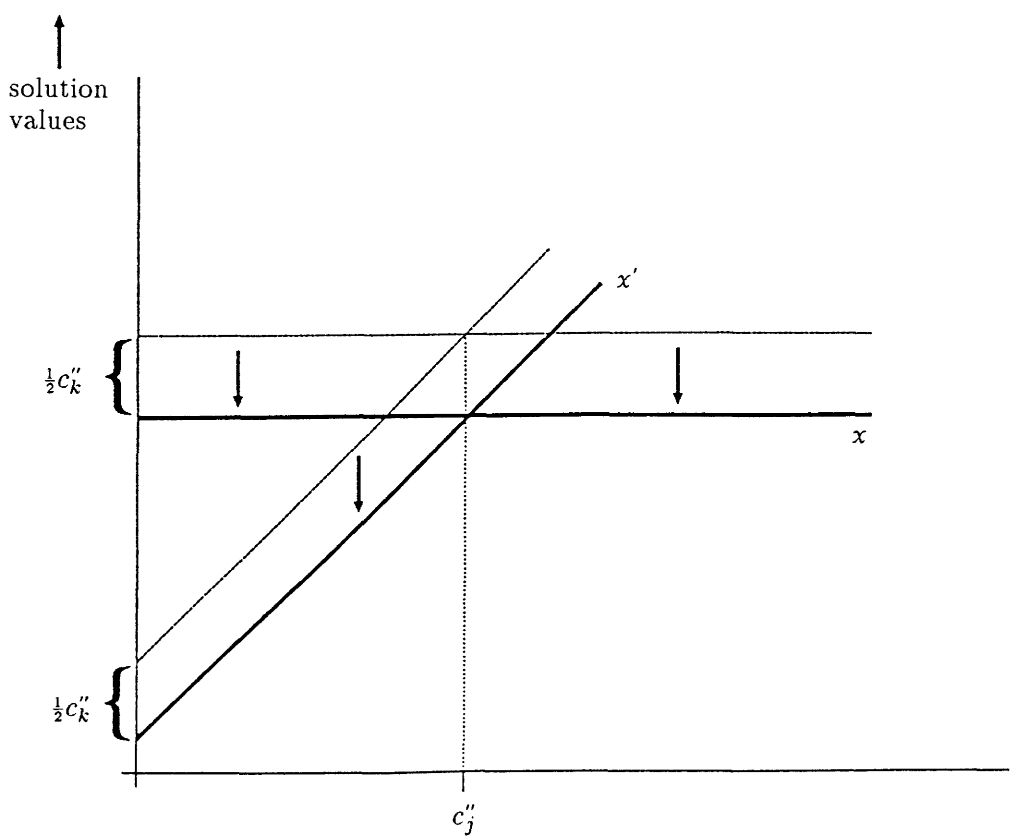
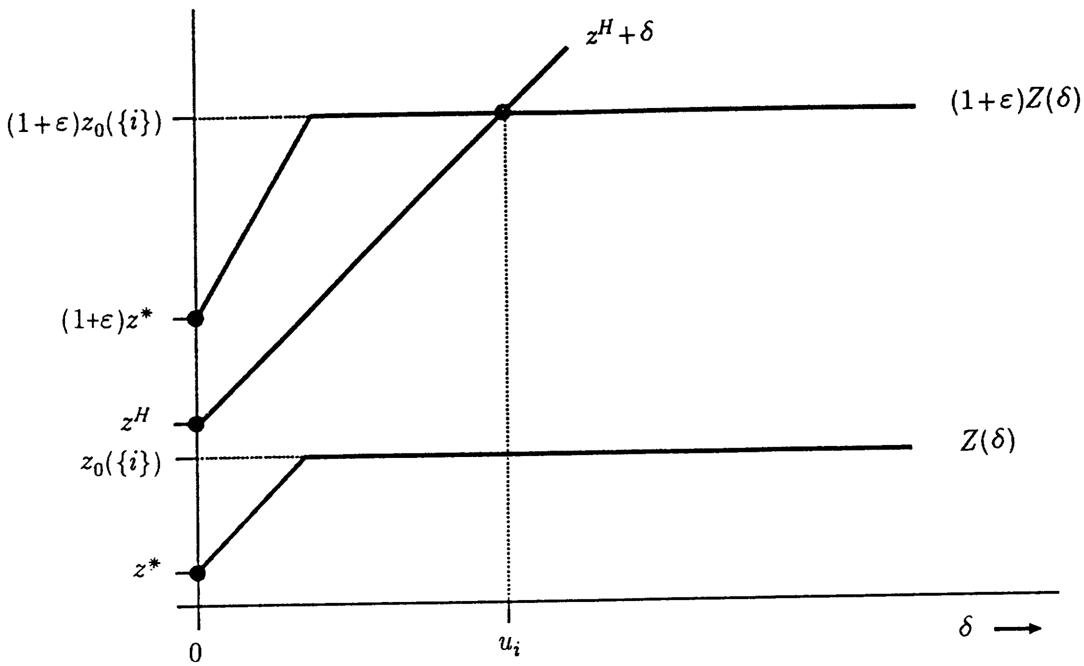
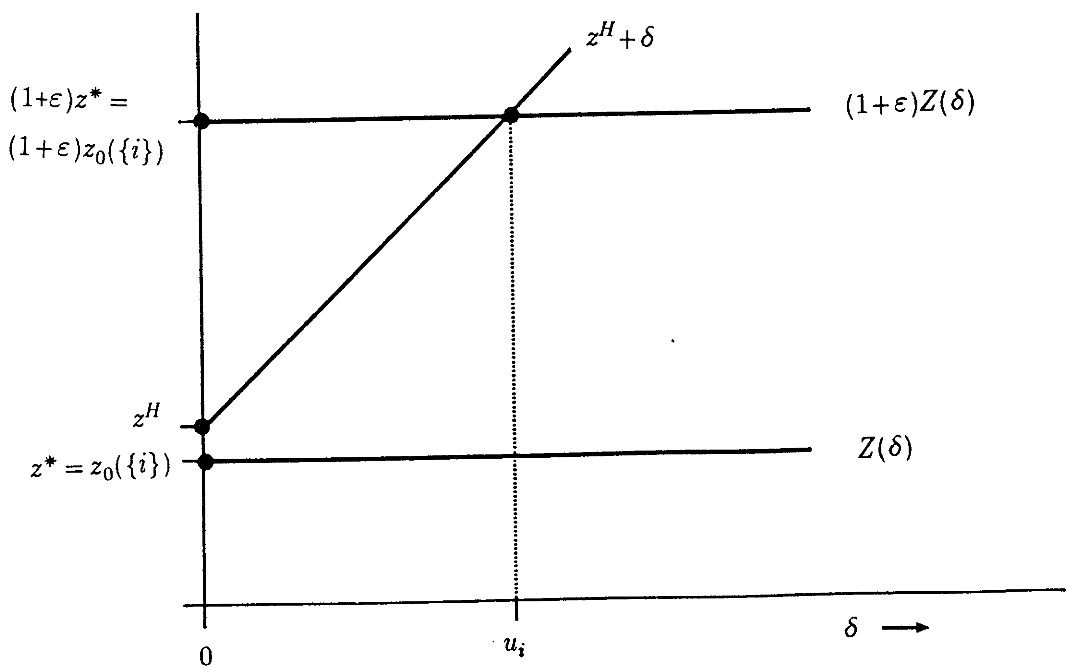


Figure 4b



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



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

Figure 5