12/2001 A0213

Raport Badawczy
Research Report

RB/52/2001

On the inverse problem and some related problems in combinatorial optimization

Marek Libura

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Pracę zgłosił: prof. dr hab. K.Kiwiel

Warszawa 2001

ON THE INVERSE PROBLEM AND SOME RELATED PROBLEMS IN COMBINATORIAL OPTIMIZATION

Marek Libura
Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warszawa, Poland
e-mail: libura@ibspan.waw.pl

Abstract

In this paper we introduce the *adjustment problem* corresponding to the generic combinatorial optimization problem. It consists in finding less costly perturbations of weights in the original problem, which guarantee that the opimal solution of the perturbed problem belong to tha specified subset of feasible solutions.

We study the properties of the adjustment problem an its relations to the standard inverse problem in combinatorial optimization.

1 Introduction

Let $E = \{e_1, \ldots, e_n\}$ be an arbitrary finite set called the *ground set.* For any subset $F \subseteq E$, $\xi(F) = (\xi_1(F), \ldots, \xi_n(F))^T \in \mathbb{B}^n$ denotes the characteristic vector of F, i.e., $\xi_i(F) = \lfloor e_i \in F \rceil$, $i = 1, \ldots, n$, where for any sentence Q, $\lfloor Q \rceil = 1$ if and only if the logical value of Q is *truth.*

Let $w : \mathbb{R}^n \times 2^E \to \mathbb{R}$ denote the real value function, which we will call the *weight function.* In this paper we assume, that for $F \subseteq E$

$$w(c, F) = c^T \cdot \xi(F), \tag{1}$$

where $c \in \mathbb{R}^n$ is a vector of so-called *weights* of elements of the ground set. For a family of subsets $\mathcal{G} \subseteq 2^E$ and $c \in \mathbb{R}^n$ let

$$\mu(c, \mathcal{G}) = \min\{w(c, F) : \ F \in \mathcal{G}\},$$

with standard convention, that for arbitrary vector $c \in \mathbb{R}^n$, $\mu(c, \mathcal{G}) = \infty$ if $\mathcal{G} = \emptyset$.

Given the weight vector $c \in \mathbb{R}$ and a family $\mathcal{F} \subseteq 2^E$ of so-called *feasible subsets* (*feasible solutions*) , the generic *combinatorial optimization problem* is defined as follows:

$$\text{Find } F^* \in \mathcal{F} \text{ such that } w(c, F^*) = \mu(c, \mathcal{F}).$$

In tis paper we will use also a more standard notation for the combinatorial optimization problem:

$$\min_{F \in \mathcal{F}} w(c, F). \tag{P}$$

Sometimes it is required to find not only a single set F^* satisfying the condition $w(c, F^*) = \mu(c, \mathcal{F})$, but the family of all such sets. Given $\mathcal{F} \subseteq 2^E$ and $c \in \mathbb{R}^n$, we will denote this family by $\Omega(c, \mathcal{F})$ and we will call any of its element an *optimal solution* of the problem (P).

For $\mathcal{F} \subseteq E$ and arbitrary family $\mathcal{G} \subseteq \mathcal{F}$ we will define the set $S(\mathcal{G})$ of all weight vectors, for which any solution belonging to $\mathcal{G}$ is an optimal solution of the problem (P). Namely,

$$S(\mathcal{G}) = \{c \in \mathbb{R}^n : w(c, F) = \mu(c, \mathcal{F}) \ \text{ for any } F \in \mathcal{G}\}.$$

The set of vectors $S(\mathcal{G})$ is called *the optimality region* with respect to the family $\mathcal{G}$.

The optimality region with respect to the family of feasible solutions generalizes in a natural way the notion of so called *stability region* with respect to a single solution $f^o \in \mathcal{F}$ (see e.g. [6, 8]). It is well known (see [8]) that for any $F^o \in \mathcal{F}$ the stability region $S(\{F^o\})$ is a polyhedral convex cone in $\mathbb{R}^n$. This implies that also an optimality region with respect to the family $\mathcal{G}$ forms a polyhedral convex cone in $\mathbb{R}^n$ which is simply an itersection of stability regions with respect to all solutions belonging to the family $\mathcal{G}$.

Most of discrete opimization problems can be stated in the above form or – at least – reformulated to problem (P). In this paper we will frequently use as an example the following combinatorial optimization problem:

Example
Consider the symmetric undirected graph G shown in Figure 1. Let E be the set of all edges of the graph G, i.e., $E = \{e_1, \ldots, e_7\}$, and let $\mathcal{T}$ denote the set of all spanning trees in the graph G. From the theorem by Kirchhoff (see e.g. [5]) it is easy to calculate that $|\mathcal{T}| = 21$. Figure 2 presents all spanning trees belonging to $\mathcal{T}$.

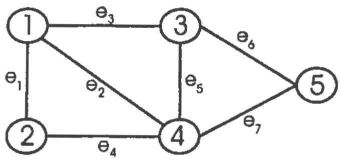

Figure 1: Graph $G = (V, E)$ from the Example.

Assume now that in the formulation of the combinatorial optimization problem (P) we take $\mathcal{F} = \mathcal{T}$ and $c = c^o = (4, 4, 1, 5, 3, 7, 8)^T$. Thus we are faced with a well known (see e.g. [11]) *minimum spanning tree problem* on the graph G with lengts of edges given by the vector c^o. This problem has a single optimal solution $F^* = \{e_1, e_3, e_5, e_6\}$ so we have $\Omega(c^o, F) = \{F^*\}$ and $w(c^o, F^*) = 15$. □

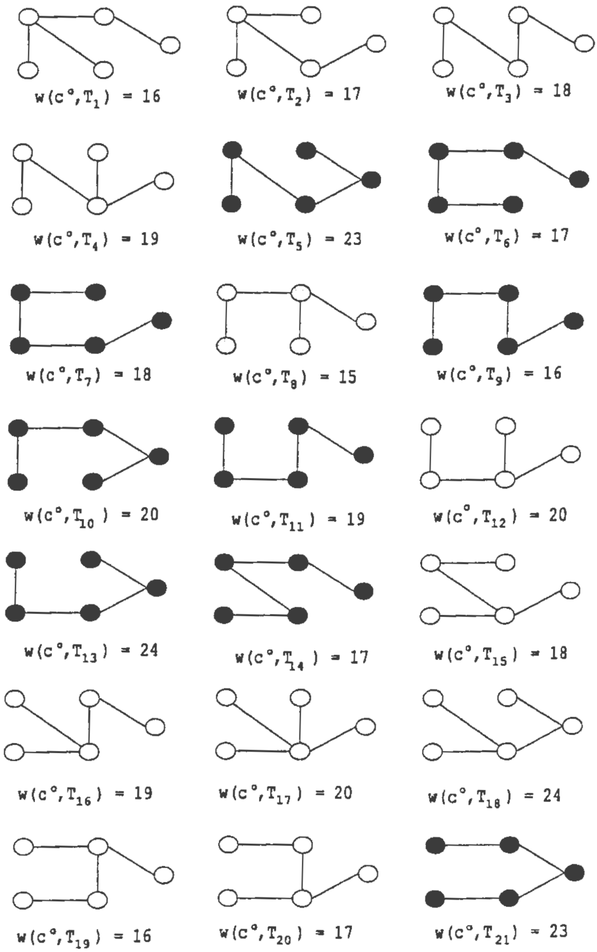

Figure 2: All spanning trees in the graph G and its weights for $c = c^o$.

2 The adjustment problem

Consider the combinatorial optimization problem (P) stated for $\mathcal{F} \subseteq 2^E$ and $c^o \in \mathbb{R}^n$

$$\min_{F \in \mathcal{F}} w(c^o, F) \tag{2}$$

Given an arbitrary subset of feasible solutions $\overline{\mathcal{F}} \subseteq \mathcal{F}$, a set of vectors of weights $\mathcal{C} \subseteq \mathbb{R}^n$, and a real value function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, we define the *adjustment problem* related to the problem (2) in the following form:

Find $c^* \in \mathcal{C}$ such that

$$f(c^*, c^o) = \min\{f(c, c^o) : \ c \in \mathcal{C}\}$$

and

$$\mu(c^*, \overline{\mathcal{F}}) = \mu(c^*, \mathcal{F}).$$

We can shortly denote the adjustment problem as follows:

$$\begin{aligned} \min_{c \in \mathcal{C}} \quad & f(c, c^o) \\ & \mu(c, \overline{\mathcal{F}}) = \mu(c, \mathcal{F}). \end{aligned} \tag{3}$$

Let $a(\overline{\mathcal{F}})$ denote the optimal value of the problem (3); we will call this value the *adjustment cost* related to the subset $\overline{\mathcal{F}}$.

The adjustment problem may be interpretted in the following way:

For a given combinatorial optimization problem (P) and an initial vector of weights c^o we want to find a new vector of weights c^*, belonging to a specified set $\mathcal{C}$, and such that some optimal solution of the modified this way problem (P), belongs to the set $\overline{\mathcal{F}}$. Moreover, we want to minimize the adjustment cost equal to $f(c^*, c^o)$.

The set $\mathcal{C}$ in the formulation of the adjustment problem is called the *restriction set* and the function f is called the *cost function.* Frequently,

$$f(c, c^o) = ||c - c^o||,$$

where $||\cdot||$ denotes some norm in $\mathbb{R}^n$. Moreover, typically $\mathcal{C} = \mathbb{R}^n$ or $\mathcal{C} = \mathbb{R}^n_+$. The subset $\overline{\mathcal{F}}$ of the family of feasible solutions $\mathcal{F}$ is called *required solutions set.*

One can iterpret the adjustment problem as an optimization problem consisting in finding the ,,cheapest" (measured by the value of cost function) and admissible (i.e., belonging to the set $\mathcal{C}$) perturbation of the original vector of weights, which guarantee that the solution of the original problem becomes a solution of the restricted problem with the feasible set $\overline{\mathcal{F}}$.

Observe that when the restriction set $\mathcal{C}$ contains a vector of weights c^c in which all components are equal, then the adjustment problem (3) has a feasible solution c^c. This follows simply from the fact that this case all feasible solutions of the problem (P) have the same weight. In particular, a solution of the adjustment problem always exists if $\mathcal{C} = \mathbb{R}^n$ or $\mathcal{C} = \mathbb{R}^n_+$.

Example (continued)

We will formulate an example of the adjustment problem related to the minimum spanning tree problem defined for the graph G shown in Figure 1. Let us take, as before, $\mathcal{F} = \mathcal{T}$ and $c = c^o = (4,4,1,5,3,7,8)^T$. Thus the initial combinatorial optimization problem (P) is stated as follows:

$$\min_{F \in \mathcal{T}} w(c^o, F) \tag{4}$$

Assume now that we are interested in such a solution of the problem (4) which is not only a spanning tree, but also forms a path in the graph G. This means that we are looking for a solution which is a Hamiltonian path in G. Denote by $\mathcal{H}$ the set of all Hamiltonian paths in G. Obviously, $\mathcal{H} \subseteq \mathcal{T}$. In our very small example it is easy to see from Figure 2, that $\mathcal{H} = \{T_3, T_5, T_6, T_7, T_9, T_{10}, T_{11}, T_{13}, T_{14}, T_{21}\}$ (spanning trees belonging to this subset are distinguished on Figure 2).

Our goal is to make the less costly adjustment of the initial weight vector c^o, which would guarantee, that the solution of the modified problem (4) is a Hamiltonian path in G.

Assume that the cost of an adjustment is measured by the l_1 norm in $\mathbb{R}^7$ and that $\mathcal{C} = \mathbb{R}^7$. Thus we have the following adjustment problem related to the minimum spanning tree problem (4):

$$\begin{aligned} \min \quad & \sum_{i=1}^{7} |c(e_i) - c^o(e_i)| \\ & \mu(c, \mathcal{H}) = \mu(c, \mathcal{T}). \end{aligned} \tag{5}$$

We will show later that this problem has the following optimal solution:

$$c^* = (4,4,1,5,3,8,8)^T.$$

We have $f(c^*, c^o) = \sum_{i=1}^{7} |c(e_i) - c^o(e_i)| = 1$. Thus the optimal value of the adjustment problem is equal to 1. Comparing the initial vector of weights

c^o and a solution c^* of the problem (5) it is easy to see that it is enough to adjust the initial vector of weights $c^o = (4,4,1,5,3,7,8)^T$ by increasing the weight $c^o(e_6)$ by 1 in order to guarantee that the optimal solution of the modified minimum spanning tree problem becomes a Hamiltonian path in the graph G.

□

The adjustment problem is closely related to so-called *inverse problem*, which attract recently significant attention (see e.g. [1]-[6], [12]-[17]).

3 The inverse problem and the adjustment problem

Given the combinatorial optimization problem (P) we will will define the *inverse optimization problem* (I) in the following general form:

For $c^o \in \mathbb{R}^n$, $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $\mathcal{F}^r \subseteq \mathcal{F}$ and $\mathcal{C} \in \mathbb{R}^n$, find $c^* \in \mathbb{R}^n$ such that

$$c^* \in \arg \min_{c \in S(\mathcal{F}^r) \cap \mathcal{C}.} f(c, c^o) \qquad \text{(I)}$$

As before, the function f in the statement of the problem (I) is called *the cost function.* The family of subsets $\mathcal{F}^r$ is called *the reference solutions set* and the vector c^o – *the reference weight vector.* The set $\mathcal{C}$ is called *the restrictions region.*

The inverse optimization problem may be interpreted as follows:

For an initial combinatorial optimization problem (P) we want to find a weight vector c^*, belonging to the restrictions region $\mathcal{C}$, for which any solution from the reference set $\mathcal{F}^r$ is optimal in the problem (P) and, moreover, the cost of changing weights vector from the reference value c^o to c^*, measured by the cost function f, is minimum.

Thus the only difference in statements of the inverse problem and the adjustment problem is that in the inverse problem we require thal *all* solutions belonging to reference solutions set became optimal after changes of weights, whereas in the adjustment problem we require that *at least one* solution from this set becomes the optimal one.

If the set $\mathcal{F}^r$ contains a single element, i.e., $\mathcal{F}^r = \{F^o\}$, then the inverse problem is stated as follows:

$$\begin{array}{ll} \min & f(c, c^o) \\ & c \in S(\{F^o\}) \cap \mathcal{C}. \end{array} \tag{6}$$

Assume now that the initial data of the inverse problem are fixed, i.e., the initial vector of weights c^o, the function f as well as the set $\mathcal{C}$ are given.

Let $i(F^o)$ denote the optimal value of the problem (6). We will call this value the *inverse cost* with respect to the feasible solution F^o. The inverse cost is simply the minimum adjustment cost necessary to make the feasible solution F^o an optimal solution of the problem (P).

It is now easy to see that the adjustment cost related to an arbitrary subset $\overline{\mathcal{F}}$ is equal to the minimum of the inverse costs with respect to all solutions belonging to the set $\overline{\mathcal{F}}$, i.e.,

$$a(\overline{\mathcal{F}}) = \min\{i(F) : F \in \overline{\mathcal{F}}\}.$$

This fact gives, in principle, the way of solving the adjustment problem through solving a sequence of the inverse problems for all elements of the set $\overline{\mathcal{F}}$.

In such a 'brut force' solution one could incorporate simple bounds for the optimal value of the inverse problem. Such bounds can be deriver for various functions f.

In the following we will consider the most typical formulation of the inverse problem. Namely we will assume that the adjustment cost is measured by l_1 norm and that the restriction set $\mathcal{C}$ is taken as $\mathbb{R}^n$. Observe that this case for $F \in \mathcal{F}$,

$$i(F) \geq w(c^o, F) - \mu(c^o, \mathcal{F})$$

and

$$i(F) \leq u(c^o),$$

where $u(c^o)$ denotes l_1 distance of the vector c^o from the line $c(e_i) =$ const, $i = 1, \ldots, n$.

4 Optimality conditions

It is easy to see that both defined problems: the adjustment problem and the inverse problem, are closely related to the optimality condition for the initial combinatorial optimization problem. In fact, this is also the reason for difficulty of solving these problems, because such optimality conditions are rather seldom available in combinatorial optimization (see e.g. [8]).

Assume for simplicity that $\mathcal{C} = \mathbb{R}^n$. Then the inverse problem (I) is stated as follows:

$$c^* \in \arg \min f(c, c^o) \qquad c \in S(\mathcal{F}^r). \tag{7}$$

The set $S(\mathcal{F}^r)$ in the formulation of the above problem is an itersection of stability regions $S(\{F\})$ with respect to all solutions $F \in \mathcal{F}^r$.

For some combinatorial optimization problems (see [8]) we can provide a complete description of the cone $S(\{F\})$ and this leads to efficient algorithms for corresponding inverse optimization problems (see e.g. [1, 3, 12, 16]).

However in general, the only available optimality conditions are so-called trivial optimality conditions (see [8]). One can hardly expext that these optimality conditions might lead to efficient algorithms for the inverse or adjustment problems, but they are usefull in understending some properties of the problems.

We will state the optimality conditions with respect to some specified feasible solution F^o in the context of the simplest inverse optimization problem (6) assuming that $\mathcal{C} = \mathbb{R}^n$ (see [8]).

Let for $F \in \mathcal{F}$, $I' = \{i : e_i \in F^o \setminus F\}$ and $I'' = \{i : e_i \in F \setminus F^o\}$. Then

$$\begin{aligned} S(F^o) = \{ & c \in \mathbb{R}^n : c_i = c_i^o + \delta_i^+ - \delta_i^- \\ & \text{and} \quad \sum_{i \in I'} (\delta_i^+ - \delta_i^-) - \sum_{i \in I''} (\delta_i^+ - \delta_i^-) \leq \sum_{i \in I'} c_i^o - \sum_{i \in I''} c_i^o \\ & \text{for any} \quad F \in \mathcal{F} \setminus \{F^o\} \, \}. \end{aligned} \tag{8}$$

Thus the inverse optimization problem may be, in principle, formulated as a (large) linear programming problem. We will illustrate this possibility on the following example.

Example (continued)

Consider again the minimum spanning tree problem defined for graph G shown on Figure 1. Appendix contains a *Mathematica* programm which generates linear programming problem for calculating the inverse cost for any spanning tree belonging to the set $\mathcal{T}$.

All spanning trees in graf G are given explicitly in table T by the incidence vectors of corresponding subsets of edges. Vector c^o denotes an initial vector of weights.

First the set of spanning trees is sorted according to the nondecreasing weights and a table S of sorted weights is produced. Then for any spanning tree the linear programming problem defined by matrix A and right-hand-side vector rhs, corresponding to inequalities (8) is solved. Last line calculates the inverse costs for all spanning trees, sorted according to nondecreasing weights.

In the table shown in Figure 3 optimal solutions for all spanning trees are given. Each row of the table contains values of the perturbations δ_i^+, $i = 1, \ldots, 7$, δ_i^-, $i = 1, \ldots, 7$, for weights of graph edges, which correspond to the minimum inverse costs.

Below all inverse costs are given. Observe that the spanning trees are ordered according to the nonincreasing weights and that there is no corresponding monotonicity in corresponding inverse costs.

$$\{0, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 4, 8, 5, 5, 9, 11, 12, 12\}$$

□

References

[1] D. Burton, P.L. Toint – On an instance of the inverse shortest path problem. *Mathematical Programming*, 53 (1992) 45-61.

[2] D. Burton, P.L. Toint – On the use of an inverse shortest paths algorithms for recovering linearly correlated costs. *Mathematical Programming*, 63 (1994) 1-22.

[3] M. Cai, X. Yang, Y. Li – Inverse polymatroidal flow problem. *Journal of Combinatorial Optimization*, 3 (1999) 115-126.

[4] R. Diestel – *Graph Theory*. Springer-Verlag, New York, Berlin, Heidelberg 2000.

	δ_1^+	δ_2^+	δ_3^+	δ_4^+	δ_5^+	δ_6^+	δ_7^+	δ_1^-	δ_2^-	δ_3^-	δ_4^-	δ_5^-	δ_6^-	δ_7^-
T_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T_1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
T_7	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T_{16}	1	0	0	0	0	0	0	0	0	0	0	0	0	0
T_2	0	0	0	0	1	1	0	0	0	0	0	0	0	0
T_6	0	0	0	0	1	0	0	0	0	0	1	0	0	0
T_{14}	1	0	0	0	1	0	0	0	0	0	0	0	0	0
T_{17}	1	0	0	0	0	1	0	0	0	0	0	0	0	0
T_4	0	0	0	0	1	1	0	0	0	0	1	0	0	0
T_8	0	3	0	0	0	0	0	0	0	0	0	0	0	0
T_{15}	1	0	0	0	1	1	0	0	0	0	0	0	0	0
T_9	0	2	0	0	0	1	0	0	0	1	0	0	0	0
T_{11}	0	3	0	0	0	0	0	0	0	0	1	0	0	0
T_{19}	1	3	0	0	0	0	0	0	0	0	0	0	0	0
T_3	0	0	0	0	1	0	0	0	0	0	0	0	3	4
T_{12}	0	3	0	0	0	1	0	0	0	0	1	0	0	0
T_{20}	1	3	0	0	0	1	0	0	0	0	0	0	0	0
T_{18}	1	0	1	0	2	0	0	0	0	0	0	0	2	3
T_{10}	0	3	0	0	1	0	0	0	0	0	0	0	3	4
T_{13}	0	6	1	0	4	0	0	0	0	0	0	0	0	1
T_{21}	1	6	0	0	4	0	0	0	0	0	0	0	0	1

Figure 3: Solutions of the inverse problems for all spanning trees from the Example.

[5] R.L. Graham, M. Grötsschel, L. Lovász –*Handbook of Combinatorics*, Elsevier, Amsterdam 1995.

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

[7] E.L. Lawler, J.K. Lenstra, A.G. Rinnoy Kan, D. Shmoys – *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* John Wiley & Sons, New York 1985.

[8] M. Libura – Optimality conditions and sensitivity analysis for combinatorial optimization problems. *Control and Cybernetics*, 25 (1996) 1165-1180.

[9] M. Libura – On accuracy of solutions for combinatorial optimization problems with perturbed coefficients of the objective function. *Annals of Operations Research*, 86 (1999) 53–62.

[10] M. Libura, E.S. van der Poort, G. Sierksma, and J.A.A. van der Veen – Stability aspects of the traveling salesman problem based on k-best solutions. *Discrete Applied Mathematics* 87 (1998) 159-185.

[11] G.L. Nemhauser, L.A. Woolsey – *Integer and Combinatorial Optimization.* John Wiley & Sons, New York 1988.

[12] P.T. Sokkalingam, R.K. Ahuja, J.B. Orlin – Solving inverse spanning tree problems through network flow techniques. *Operations Research*, 47 (1999) 291-298.

[13] Y.N. Sotskov, V.K. Leontev, and E.N. Gordeev – Some concepts of stability analysis in combinatorial optimization. *Discrete Applied Mathematics*, 58 (1995) 169-190.

[14] S. van Hoesel, A. Wagelmans – On the complexity of postoptimal analysis of 0/1 programs. *Discrete Applied Mathematics*, 91 (1999) 251-263.

[15] S. Xu, J. Zhang – An inverse problem of weighted shortest path problems. *Japanese Journal of Applied and Industrial Mathematics*, 12 (1995) 47-59.

[16] J. Zhang, M. Cai – Inverse problem of minimum cuts. *Mathematical Methods of Operations Research*, 47 (1998) 51-58.

[17] J. Zhang, Z. Ma – Solution structure of some inverse combinatorial optimization problems. *Journal of Combinatorial Optimization*, 3 (1999) 127-139.

5 Appendix

$$T = \{\{1,1,1,0,0,1,0\},\\ \{1,1,1,0,0,0,1\},\\ \{1,1,0,0,0,1,1\},\\ \{1,1,0,1,0,0,1\},\\ \{1,1,0,0,1,1,0\},\\ \{1,1,0,1,0,1,0\},\\ \{1,1,0,0,1,0,1\},\\ \{1,0,1,0,1,1,0\},\\ \{1,0,1,0,1,0,1\},\\ \{1,0,1,0,0,1,1\},\\ \{1,0,0,1,1,1,0\},\\ \{1,0,0,1,1,0,1\},\\ \{1,0,0,1,0,1,1\},\\ \{0,1,1,1,0,1,0\},\\ \{0,1,1,1,0,0,1\},\\ \{0,1,0,1,1,1,0\},\\ \{0,1,0,1,1,0,1\},\\ \{0,1,0,1,0,1,1\},\\ \{0,0,1,1,1,1,0\},\\ \{0,0,1,1,1,0,1\},\\ \{0,0,1,1,0,1,1\}\}$$

$$c^o = \{4,1,4,5,3,7,8\};$$

```
S = Sort[T,
    OrderedQ[{Inner[Times, #1, c], Inner[Times, #2, c]}]&];

b = Table[Table[w[[i]] - w[[j]], {j, 21}], {i, 21}];

A = Table[Table[Join[(1 - S[[i]])S[[j]] - S[[i]](1 - S[[j]]),
        -(1 - S[[i]])S[[j]] + S[[i]](1 - S[[j]])], {j, 1, 21}],
    {i, 1, 21}];

Inv = Table[constr = Delete[A[[i]], i];
    rhs = Delete[b[[i]], i]; LinearProgramming[
      {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}, constr, rhs],
    {i, 1, 21}];

InverseCost = Apply[Plus, Inv,{1}]
```