40/2002

Raport Badawczy

Research Report

RB/66/2002

Adjustment Problem for Binary Constrained Linear Programming Problems

M. 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

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Krzysztof Kiwiel

Warszawa 2002

ADJUSTMENT PROBLEM FOR BINARY CONSTRAINED LINEAR PROGRAMMING PROBLEMS

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 the adjustment problem corresponding to linear programming problems with explicit or implicit binary constraints is considered. It consists in finding less costly perturbations of weights in the original problem, which guarantee that the optimal solution of the perturbed problem belongs to the specified subset of feasible solutions. We propose a method of solving problems of this type. The approach is based on using optimality conditions for corresponding linear programming relaxation.

1 Introduction

Let $X \subseteq \mathbb{R}^n$ and $c \in \mathbb{R}^n$. We will consider a mathematical programming problem with linear objective function

$$v(c, X) = \max\{c^T x : x \in X\}. \tag{P}$$

Let $F \subseteq X$ and $\Delta \subseteq \mathbb{R}^n$. The *adjustment problem* related to (P) is stated as follows:

$$a(F) = \min\{\|\delta\| : v(c+\delta, X) = v(c+\delta, F), \delta \in \Delta\}, \tag{A}$$

where $\|\delta\|$ denotes a norm of δ. In this paper we will consider mainly l_1 norm, i.e., $\|\delta\| = \|\delta\|_{l_1} = \sum_{i=1}^n |\delta_i|$.

The adjustment problem has been introduced in [9]. It can be interpreted in the following way: For a given problem (P) and its restriction defined by the solutions set F we want to find the less costly (in the sense of a given norm) and admissible (belonging to some specified set Δ) perturbations of coefficients in the objective function of (P) which guarantee that some optimal solution of the perturbed problem is also feasible (and thus – optimal) for this restriction of (P).

If for example the problem (P) is the maximum weight tree problem in a given graph we may look for such perturbations of lengths of edges, that an optimal solution of perturbed problem forms a Hamiltonian path in this graph. Similarly, if (P) is a linear programming problem, then we may be interested in such perturbations of the objective function coefficients, which would guarantee that there is an optimal solution of the perturbed problem, satisfying additional restrictions, e.g. integrality restrictions.

The adjustment problem may be infeasible, but - as we will see later - in some important cases its solution exists.

When the restricted solution set F contains only a single element, i.e., $F = \{x^o\}$, then the adjustment problem becomes the so called *inverse problem* with respect to x^o:

$$i(x^o) = a(\{x^o\}) = \min\{\|\delta\| : v(c+\delta, X) = (c+\delta)^T x^o,\ \delta \in \Delta\}, \tag{I}$$

The inverse problem (I) and some of its variants have attracted recently significant attention (see e.g. [1, 2, 3, 16, 17, 18]). Observe that immediately from the definitions of the adjustment problem and the inverse problem we obtain the following fact:

Proposition 1 *For $F \subseteq X$,*

$$a(F) = \min\{i(x) : x \in F\}. \tag{1}$$

2 The adjustment problem for linear programming problem

Let

$$X = \{x \in \mathbb{R}^n : Ax \leq b,\ x \geq 0\},$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. This case the problem (P) is a linear programming problem

$$v(c, X) = \max\{c^T x : Ax \leq b,\ x \geq 0\}. \tag{2}$$

The following lemma states optimality conditions for problem (2) (see e.g. [12]):

Lemma 1 *A feasible solution x^o is an optimal solution of the problem (2) if and only if there exists $y \in \mathbb{R}^m_+$ such that*

(i) $A^T y \geq c$,

(ii) $c^T x^o = b^T y$.

Given $\Delta \subseteq \mathbb{R}^n$ and a feasible solution x^o for (2), it follows from Lemma 1 that the inverse problem with respect to x^o can be stated as the following mathematical programming problem

$$\begin{aligned} i(x^o) \;=\; & \min \|\delta\| \\ & A^T y - \delta \geq c \\ & b^T y - \delta^T x^o = c^T x^o \\ & y \geq 0,\ \delta \in \Delta \end{aligned} \tag{ILP}$$

Observe that if Δ is a polyhedral convex set in $\mathbb{R}^n$, then for l_1 and l_∞ norms in $\mathbb{R}^n$ the problem (ILP) can be easily stated as a linear programming problem.

Let $F = \{x \in X : Dx \leq d,\ x \in S\}$, where $D \in \mathbb{R}^{p \times n}$, $d \in \mathbb{R}^p$ and S is some specified subset of $\mathbb{R}^n$. Thus the restriction of the original problem (P) is defined by adding new linear constraints

$$Dx \leq d$$

and requiring that solutions belong to the set S. In the following we will usually assume that $S = \mathbb{Z}^n$, where $\mathbb{Z}$ is the set of integers, or we will simply take $S = \mathbb{R}^n$.

The adjustment problem with respect to F can now be formulated as the following mathematical programming problem:

$$\begin{aligned} a(F) \;=\; & \min \|\delta\| \\ & A^T y - \delta \geq c \\ & b^T y - c^T x - \delta^T x = 0 \\ & Ax \leq b \\ & Dx \leq d \\ & x, y \geq 0,\ \delta \in \Delta,\ x \in S \end{aligned} \tag{ALP}$$

Observe, that even if Δ is a polyhedral convex set and $S = \mathbb{R}^n$, then (ALP) is no longer a linear programming problem due to nonlinear term $\delta^T x$.

3 Adjustment problem with binary restrictions

Consider now a special case of the adjustment problem of the form (ALP); namely, assume that $F \subseteq \mathbb{B}^n$. There are two important situations when problems of this type appear:

The first one is quite natural: we simply may require that in the restricted problem $S = \mathbb{B}^n$, or that $S = \mathbb{Z}^n$ and the linear constraints $Ax \leq b,\ Dx \leq d$ contain (or imply) inequalities $Ix \leq \mathbf{1}$, where $I \in \mathbb{R}^{n \times n}$ is an identity matrix and $\mathbf{1}$ denotes a vector of ones.

Another case, which is also important from the practical point of view, appears when $S = \mathbb{R}^n$ and the constraints matrix of the restricted problem, i.e., the matrix (A^T, D^T) is totally unimodular (see e.g. [11]).

If the set of feasible solutions of restricted problem fulfills the requirement $F \subseteq \mathbb{B}^n$, then nonlinear therm $\delta^T x$ in (ALP) may be formally linearized in a standard way using additional variables and constraints. To do this it will be convenient to express nonrestricted in sign vector $\delta \in \mathbb{R}^n$ as a difference of two nonnegative vectors. Let

$$\delta = \delta^+ - \delta^-,$$

where

$$\begin{aligned} \delta^+ &= (\delta_1^+, \ldots, \delta_n^+),\ \delta_i^+ = \max\{0, \delta_i\},\ i = 1, \ldots, n, \\ \delta^- &= (\delta_1^-, \ldots, \delta_n^-),\ \delta_i^- = \min\{0, \delta_i\},\ i = 1, \ldots, n. \end{aligned}$$

Thus we have

$$\delta^T x = \sum_{i=1}^{n} (\delta_i^- x_i - \delta_i^- x_i).$$

For $i = 1, \ldots, n$, we will introduce new variables $z_i^+, z_i^- \in \mathbb{R}_+$ satisfying the following conditions:

$$z_i^+ = \delta_i^+ x_i, \quad i = 1, \ldots, n, \tag{3}$$

$$z_i^- = \delta_i^- x_i, \quad i = 1, \ldots, n, \tag{4}$$

Constraint

$$b^T y - c^T x - \delta^T x = 0$$

in the formulation of problem (ALP) can be now replaced with a linear constraint

$$b^T y - c^T x - \mathbf{1}^T z^+ + \mathbf{1}^T z^- = 0,$$

where $z^+ = (z_1^+, \ldots, z_n^+)$ and $z^- = (z_1^-, \ldots, z_n^-)$.

For any new variable z_i^+, z_i^-, $i = 1, \ldots, n$, we have to add also constraints which would guarantee that equations (3) and (4) hold.

Let us take for example the equation $z_i^+ = \delta_i^+ x_i$ for some index i. It is equivalent to two implications

$$x_i = 0 \implies z_i^+ = 0,$$

$$x_i = 1 \implies z_i^+ = \delta_i^+,$$

which can be modeled in a standard way by adding the following new constraints:

$$\begin{aligned} z_i^+ - M x_i &\leq 0, \\ -\delta_i^+ + z_i^+ &\leq 0, \\ \delta_i^+ - z_i^+ + M x_i &\leq M, \end{aligned}$$

where M is sufficiently large constant satisfying the inequality $\delta_i^+ \leq M$ for any $i = 1, \ldots, n$.

If the set of admissible perturbations Δ is bounded, then the value of M can be calculated directly from the description of Δ. If $\Delta = \mathbb{R}^n$, then we can simply take $M = \|c\|_{l_1}$. Indeed, this case $y = 0$ and $\delta = -c$ provide a feasible solution of (ALP) for any $x \in F$ and thus there exists an optimal solution of (ALP), in which $|\delta_i| \leq \|c\|_{l_1}$.

Finally, the adjustment problem may be stated in the following form:

$$
\begin{aligned}
a(F) \;=\; \min\; & \|\delta^+\| + \|\delta^-\| \\
& A^T y - \delta^+ + \delta^- \geq c \qquad\qquad \text{(AP)} \\
& b^T y - c^T x - \mathbf{1}^T z^+ + \mathbf{1}^T z^- = 0 \\
& z^+ - Mx \leq 0, \\
& -\delta^+ + z^+ \leq 0, \\
& \delta^+ - z^+ + Mx \leq M, \\
& z^- - Mx \leq 0, \\
& -\delta^- + z^- \leq 0, \\
& \delta^- - z^- + Mx \leq M, \\
& x \in F \subseteq \mathbb{B}^n \\
& \delta^+ - \delta^- \in \Delta \\
& x,\, y,\, z^+, z^-, \delta^+, \delta^- \geq 0,
\end{aligned}
$$

Thus for initial linear programming problem (P) and the set Δ given as a polyhedral convex set, the adjustment problem for $F \subseteq \mathbb{B}^n$ and l_1 or l_∞ norms in $\mathbb{R}^n$ can be stated as a mixed integer linear programming problem. We will illustrate this fact with several examples.

Example 1 Consider a weighted digraph D shown in Figure 1.

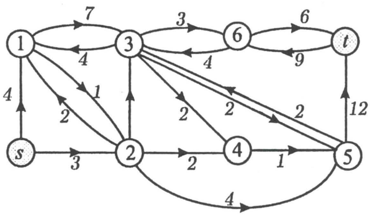

Figure 1: Digraph D from Example 1 with indicated lengths of arcs.

The following path of length 17 (given as a subset of arcs) is the shortest path from vertex s to vertex t in the digraph D:

$$p = \{(s,2), (2,4), (4,5), (5,3), (3,6), (6,t)\}.$$

This path is indicated with bold arcs in Figure 2.

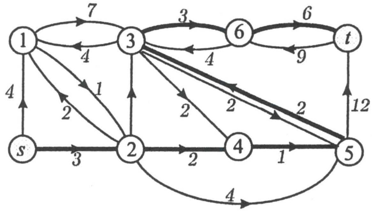

Figure 2: An optimal path in digraph D from Example 1.

Assume that we are interested in paths which pass through vertex 1 and we want to find the smallest possible modification of arcs lengths which would guarantee that there is such a path among optimal solutions of the modified problem. Therefore we have to solve the adjustment problem related to the original shortest path problem. In the restricted problem we have to consider additional requirement that a path contains the vertex 1.

It is well known that the shortest path problem in $D = (V, E)$ can be stated as a linear programming problem (see e.g. [11]). Namely, let $V = \{v_1, \ldots, v_8\} = \{s, 1, 2, 3, 4, 5, 6, t\}$ and $E = \{a_1, \ldots, a_{17}\} = \{(s, 1), (s, 2),$ (1,2), (1,3), (2,1), (2,4), (2,5), (3,1), (3,4), (3,5) (3,6), (4,5), (5,3), (5,t), (6,3), $(6, t), (t, 6)\}$. Denote by A the incidence matrix of digraph D. The initial vector of arcs lengths is given below:

$$c = (4, 3, 1, 7, 2, 2, 4, 4, 2, 3, 1, 2, 12, 4, 6, 9)^T.$$

Let $x = (x_1, \ldots, x_{17})^T \in \mathbb{R}^n_+$ denote the vector of decision variables. It is well known, that for $b = (1, 0, 0, 0, 0, 0, 0 - 1)^T$ the set of vertices of the polyhedron X, where

$$X = \{x \in \mathbb{R}^{17}_+ : Ax = b\},$$

forms a set of characteristic vectors of paths from s to t in digraph D. Any vertex of X is a binary vector, because the matrix A is totally unimodular. An optimal solution of the original problem, which corresponds to the path $p = \{(s, 2), (2, 4), (4, 5), (5, 3), (3, 6), (6, t)\}$ is given by the following vector:

$$x^o = (0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1)^T.$$

If we are interested in paths passing through the vertex 1 (observe that the path p does not fulfill this condition) we are faced with a restriction of the shortest path problem in which the set of feasible solutions contains additional constraints. For example we can require that at least one arc leaving the vertex 1 belongs to the feasible path. This leads to the following feasible set in a restriction of the original problem:

$$F = \{x \in X : x_3 + x_4 \geq 1\}.$$

Appendix 1 contains complete formulation of the corresponding adjustment problem (AP) with l_1 norm. Solving this problem we obtain the following optimal solution:

$$\delta^+ = (0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T,$$

$$\delta^- = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T,$$

$$x = (1,0,1,0,0,1,0,0,0,0,1,1,1,0,0,0,1)^T.$$

The optimal value $a(F)$ of the adjustment problem is equal to 2. The solution can be interpreted in the following way: To guarantee that the shortest path in the modified graph D pass through the vertex 1 we have to increase the weight of arc $a_2 = (s,2)$ by 2. Moreover, this is the smallest possible perturbation of lengths of arcs to achieve this goal (in the sense of l_1 norm). The optimal path in the digraph D with modified lengths of arcs is shown on Figure 3.

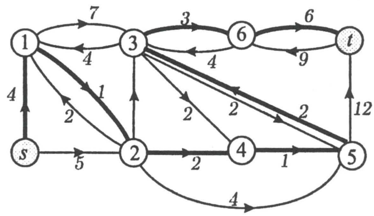

Figure 3: Digraph D from Example 1 with modified lengths of arc and indicated optimal path from s to t.

In a similar way we may solve the adjustment problem in the case when for example we require that the shortest path must not pass through specified subset of vertices, e.g., vertices 2 and 6. This case the restriction of the original problem corresponds to the set of feasible solutions F, where

$$F' = \{x \in X : x_5 + x_6 + x_7 + x_{17} = 0\}.$$

Solving the corresponding adjustment problem we obtain $a(F') = 9$ and

$$\delta'^{+} = (0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0),$$

$$\delta'^{-} = (2,0,0,1,0,0,0,0,5,0,0,0,0,0,0,0,0),$$

$$x' = (1,0,0,1,0,0,0,0,1,0,0,1,0,1,0,0,0).$$

Observe that this case the lengths of arcs in the modified digraph are not longer nonnegative. We could avoid this specifying the set Δ of admissible modifications of weights. To guarantee that weights in modified digraph are nonnegative it is enough to add to the adjustment problem inequalities $\delta^- \leq c$.

Example 2 Consider a continuous knapsack problem (P) in the form:

$$\begin{aligned} \max\ & 7x_1 + 4x_2 + 5x_3 + 2x_4 \\ & 3x_1 + 3x_2 + 4x_3 + 2x_4 \leq 7 \\ & 0 \leq x_1, x_2, x_3, x_4 \leq 1 \end{aligned}$$

An optimal solution of this problem has value $v(P) = 12.5$ an is given by the following vector:

$$x^o = (x_1^o, x_2^o, x_3^o, x_4^o)^T = (1, 1, 0.25, 0)^T.$$

Assume that we are interested in integer solutions of the problem (P) and that we want to modify coefficients of the original problem in such a way, that the set of optimal solutions of modified problem contains an integer vector. Thus we want to solve the adjustment problem corresponding to a restriction defined by $F = X \cap \mathbb{B}^4$, where

$$X = \{x \in \mathbb{R}^4 : 3x_1 + 3x_2 + 4x_3 + 2x_4 \leq 7,\ 0 \leq x_1, x_2, x_3, x_4 \leq 1\}.$$

The adjustment problem corresponding to this restriction is formulated in Appendix 2. An optimal solution of the adjustment problem is equal to 0.25. This means that the sum of absolute values of all perturbations of objective coefficients, which are necessary to guarantee integrality of solution is equal to 0.25. Optimal perturbations of the coefficients are given by the following vectors:

$$\delta^+ = (0,0,0,0)^T,$$

$$\delta^- = (0,-0.25,0,0)^T.$$

The optimal solution of the modified problem (P) is now an integral one:

$$x = (1,0,1,0)^T.$$

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.

[5] 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.

[6] 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.

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

[8] 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.

[9] M. Libura – On the inverse problem and some related problems in combinatorial optimization. Research Report RB/52/2001, Systems Research Institute, Polish Academy of Sciences, Warszawa 2001.

[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] M. Padberg – *Integer Programming and Extensions.* Springer-Verlag, Berlin, Heidelberg, New York 1995.

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

[14] 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.

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

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

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

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

4 Appendix 1

This section contains complete formulation of the adjustment problem (AP) from Example 1 in so-called LP format used in an optimization package CPLEX 6.5. Due to conventions used in the package the following notation is applied: Element x_a, which corresponds to edge $a = (i, j)$, is denoted by `xij`. Dual varibles y_v for $v \in V$ are denoted by `yv`. For elements δ_a^+ and δ_a^-, $a = (i, j) \in A$, we use symbols `uij` and `lij`, respectively. Elements z_a^+ and z_a^-, $a = (i, j) \in E$, are denoted by `zuij` and `zlij`. We take $M = 100$ as sufficiently large constant.

```
Minimize
   obj:  us1 + ls1 + us2 + ls2 + u12 + l12 + u21 + l21 + u13
         + l13 + u31 + l31 + u24 + l24 + u25 + l25 + u34 + l34
         + u35 + l35 + u36 + l36 + u63 + l63 + u45 + l45 + u53
         + l53 + u6t+ l6t + ut6 + lt6 + u5t + l5t
```

```
Subject To
   c1:  ys = 0
   c2:  - us1 + ls1 + ys - y1 <= 4
   c3:  - us2 + ls2 + ys - y2 <= 3
   c4:  - u12 + l12 + y1 - y2 <= 1
   c5:  - u21 + l21 - y1 + y2 <= 2
   c6:  - u13 + l13 + y1 - y3 <= 7
   c7:  - u31 + l31 - y1 + y3 <= 4
   c8:  - u24 + l24 + y2 - y4 <= 2
   c9:  - u25 + l25 + y2 - y5 <= 4
   c10:  - u34 + l34 + y3 - y4 <= 2
   c11:  - u35 + l35 + y3 - y5 <= 2
   c12:  - u36 + l36 + y3 - y6 <= 3
   c13:  - u63 + l63 - y3 + y6 <= 4
   c14:  - u45 + l45 + y4 - y5 <= 1
   c15:  - u53 + l53 - y3 + y5 <= 2
   c16:  - u6t + l6t + y6 - yt <= 6
   c17:  - ut6 + lt6 - y6 + yt <= 9
   c18:  - u5t + l5t + y5 - yt <= 12
   c19:  xs1 + xs2 = 1
   c20:  - xs1 + x13 + x12 - x21 - x31 = 0
   c21:  - xs2 - x12 + x21 + x24 + x25 = 0
   c22:  - x13 + x31 + x34 + x36 - x53 - x63 = 0
   c23:  - x24 - x34 + x45 = 0
   c24:  - x25 + x53 - x45 + x5t = 0
   c25:  - x36 + x63 + x6t - xt6 = 0
   c26:  - x5t - x6t + xt6 = -1
   cc:  ys - yt - 4 xs1 - 3 xs2 - 7 x13 - x12 - 2 x21 - 4 x31
        - 2 x24 - 4 x25 - 2 x34 - 3 x36 - 2 x53 - 4 x63
        - x45 - 12 x5t - 6 x6t - 9 xt6 - 2 x35 - zus1
        + zls1 - zus2 + zls2 - zu12 + zl12 - zu21 + zl21
        - zu13 + zl13 - zu31 + zl31 - zu24 + zl24 - zu25
        + zl25 - zu34 + zl34 - zu35 + zl35 - zu36 + zl36
        - zu63 + zl63 - zu45 + zl45 - zu53 + zl53 - zu6t
        + zl6t - zut6 + zlt6 - zu5t + zl5t = 0
   c28:  - 100 xs1 + zus1 <= 0
   c29:  - us1 + zus1 <= 0
   c30:  us1 + 100 xs1 - zus1 <= 100
   c31:  - 100 xs1 + zls1 <= 0
   c32:  - ls1 + zls1 <= 0
   c33:  ls1 + 100 xs1 - zls1 <= 100
   c34:  - 100 xs2 + zus2 <= 0
   c35:  - us2 + zus2 <= 0
```

```
c36:  us2 + 100 xs2 - zus2 <= 100
c37:  - 100 xs2 + zls2 <= 0
c38:  - ls2 + zls2 <= 0
c39:  ls2 + 100 xs2 - zls2 <= 100
c40:  - 100 x12 + zu12 <= 0
c41:  - u12 + zu12 <= 0
c42:  u12 + 100 x12 - zu12 <= 100
c43:  - 100 x12 + zl12 <= 0
c44:  - l12 + zl12 <= 0
c45:  l12 + 100 x12 - zl12 <= 100
c46:  - 100 x21 + zu21 <= 0
c47:  - u21 + zu21 <= 0
c48:  u21 + 100 x21 - zu21 <= 100
c49:  - 100 x21 + zl21 <= 0
c50:  - l21 + zl21 <= 0
c51:  l21 + 100 x21 - zl21 <= 100
c52:  - 100 x13 + zu13 <= 0
c53:  - u13 + zu13 <= 0
c54:  u13 + 100 x13 - zu13 <= 100
c55:  - 100 x13 + zl13 <= 0
c56:  - l13 + zl13 <= 0
c57:  l13 + 100 x13 - zl13 <= 100
c58:  - 100 x31 + zu31 <= 0
c59:  - u31 + zu31 <= 0
c60:  u31 + 100 x31 - zu31 <= 100
c61:  - 100 x31 + zl31 <= 0
c62:  - l31 + zl31 <= 0
c63:  l31 + 100 x31 - zl31 <= 100
c64:  - 100 x24 + zu24 <= 0
c65:  - u24 + zu24 <= 0
c66:  u24 + 100 x24 - zu24 <= 100
c67:  - 100 x24 + zl24 <= 0
c68:  - l24 + zl24 <= 0
c69:  l24 + 100 x24 - zl24 <= 100
c70:  - 100 x25 + zu25 <= 0
c71:  - u25 + zu25 <= 0
c72:  u25 + 100 x25 - zu25 <= 100
c73:  - 100 x25 + zl25 <= 0
c74:  - l25 + zl25 <= 0
c75:  l25 + 100 x25 - zl25 <= 100
c76:  - 100 x34 + zu34 <= 0
c77:  - u34 + zu34 <= 0
c78:  u34 + 100 x34 - zu34 <= 100
```

```
c79:  - 100 x34 + zl34 <= 0
c80:  - l34 + zl34 <= 0
c81:  l34 + 100 x34 - zl34 <= 100
c82:  - 100 x35 + zu35 <= 0
c83:  - u35 + zu35 <= 0
c84:  u35 + 100 x35 - zu35 <= 100
c85:  - 100 x35 + zl35 <= 0
c86:  - l35 + zl35 <= 0
c87:  l35 + 100 x35 - zl35 <= 100
c88:  - 100 x36 + zu36 <= 0
c89:  - u36 + zu36 <= 0
c90:  u36 + 100 x36 - zu36 <= 100
c91:  - 100 x36 + zl36 <= 0
c92:  - l36 + zl36 <= 0
c93:  l36 + 100 x36 - zl36 <= 100
c94:  - 100 x63 + zu63 <= 0
c95:  - u63 + zu63 <= 0
c96:  u63 + 100 x63 - zu63 <= 100
c97:  - 100 x63 + zl63 <= 0
c98:  - l63 + zl63 <= 0
c99:  l63 + 100 x63 - zl63 <= 100
c100:  - 100 x45 + zu45 <= 0
c101:  - u45 + zu45 <= 0
c102:  u45 + 100 x45 - zu45 <= 100
c103:  - 100 x45 + zl45 <= 0
c104:  - l45 + zl45 <= 0
c105:  l45 + 100 x45 - zl45 <= 100
c106:  - 100 x53 + zu53 <= 0
c107:  - u53 + zu53 <= 0
c108:  u53 + 100 x53 - zu53 <= 100
c109:  - 100 x53 + zl53 <= 0
c110:  - l53 + zl53 <= 0
c111:  l53 + 100 x53 - zl53 <= 100
c112:  - 100 x6t + zu6t <= 0
c113:  - u6t + zu6t <= 0
c114:  u6t + 100 x6t - zu6t <= 100
c115:  - 100 x6t + zl6t <= 0
c116:  - l6t + zl6t <= 0
c117:  l6t + 100 x6t - zl6t <= 100
c118:  - 100 xt6 + zut6 <= 0
c119:  - ut6 + zut6 <= 0
c120:  ut6 + 100 xt6 - zut6 <= 100
c121:  - 100 xt6 + zlt6 <= 0
```

```
    c122:  - lt6 + zlt6 <= 0
    c123:  lt6 + 100 xt6 - zlt6 <= 100
    c124:  - 100 x5t + zu5t <= 0
    c125:  - u5t + zu5t <= 0
    c126:  u5t + 100 x5t - zu5t <= 100
    c127:  - 100 x5t + zl5t <= 0
    c128:  - l5t + zl5t <= 0
    c129:  l5t + 100 x5t - zl5t <= 100
Bounds
    ys Free
    y1 Free
    y2 Free
    y3 Free
    y4 Free
    y5 Free
    y6 Free
    yt Free
    0 <= xs1 <= 1
    0 <= xs2 <= 1
    0 <= x13 <= 1
    0 <= x12 <= 1
    0 <= x21 <= 1
    0 <= x31 <= 1
    0 <= x24 <= 1
    0 <= x25 <= 1
    0 <= x34 <= 1
    0 <= x36 <= 1
    0 <= x53 <= 1
    0 <= x63 <= 1
    0 <= x45 <= 1
    0 <= x5t <= 1
    0 <= x6t <= 1
    0 <= xt6 <= 1
    0 <= x35 <= 1
All other variables are >= 0.
Binaries
    xs1 xs2 x13 x12 x21 x31 x24 x25 x34 x36 x53 x63 x45 x5t x6t xt6 x35
```

5 Appendix 2

This section contanins a formulation of the adjustment problem from Example 2. The problem is stated in CPLEX LP form. We use similar conventions in notation as in previous example.

```
Minimize
obj:  u1 + u2 + u3 + u4 + l1 + l2 + l3 + l4
Subject To
   c1:  3 x1 + 3 x2 + 4 x3 + 2 x4 <= 7
   c2:  - u1 + l1 + 3 y0 + y1 >= 7
   c3:  - u2 + l2 + 3 y0 + y2 >= 4
   c4:  - u3 + l3 + 4 y0 + y3 >= 5
   c5:  - u4 - l4 + 2 y0 + y4 >= 2
   vp:  7 x1 + 4 x2 + 5 x3 + 2 x4 + zu1 + zu2 + zu3 + zu4 - zl1 - zl2
- zl3 - zl4
   - p = 0
   vd:  - 7 y0 - y1 - y2 - y3 - y4 + d = 0
   cc:  p - d = 0
   r:  x2 + x3 >= 1
   c10:  - 10 x1 + zu1 <= 0
   c11:  - u1 + zu1 <= 0
   c12:  u1 + 10 x1 - zu1 <= 10
   c13:  - 10 x2 + zu2 <= 0
   c14:  - u2 + zu2 <= 0
   c15:  u2 + 10 x2 - zu2 <= 10
   c16:  - 10 x3 + zu3 <= 0
   c17:  - u3 + zu3 <= 0
   c18:  u3 + 10 x3 - zu3 <= 10
   c19:  - 10 x4 + zu4 <= 0
   c20:  - u4 + zu4 <= 0
   c21:  u4 + 10 x4 - zu4 <= 10
   c22:  - 10 x1 + zl1 <= 0
   c23:  - l1 + zl1 <= 0
   c24:  l1 + 10 x1 - zl1 <= 10
   c25:  - 10 x2 + zl2 <= 0
   c26:  - l2 + zl2 <= 0
   c27:  l2 + 10 x2 - zl2 <= 10
   c28:  - 10 x3 + zl3 <= 0
   c29:  - l3 + zl3 <= 0
   c30:  l3 + 10 x3 - zl3 <= 10
   c31:  - 10 x4 + zl4 <= 0
```

```
   c32:  - l4 + zl4 <= 0
   c33:  l4 + 10 x4 - zl4 <= 10
Bounds
   0 <= x1 <= 1
   0 <= x2 <= 1
   0 <= x3 <= 1
   0 <= x4 <= 1
All other variables are >= 0.
Binaries
   x1 x2 x3 x4
```