VU University Amsterdam
Department of Mathematics

University of Silesia in Katowice
Institute of Mathematics

Joint Master Programme of Mathematics

Arkadiusz Hulbój
Student no. 218258 (UŚ), 2002450 (VU)

Sensitivity Analysis of MIPs with an Application to Call Center Shift Scheduling

Master's Thesis in Mathematics

Supervisors:
- Dr. Sandjai Bhulai
  VU University Amsterdam
  Department of Mathematics
- Dr. Sebastian Sitarz
  University of Silesia
  Institute of Mathematics

June 2010

Supervisor's statement

Hereby I confirm that the present thesis was prepared under my supervision and that it fulfills the requirements for the degree of Master of Science in Mathematics.

Date Supervisor's signature

Author's statement

Hereby I declare that the present thesis was prepared by me and none of its contents was obtained by means that are against the law.

I also declare that the present thesis is a part of the Joint Master Programme of Mathematics of the University of Silesia and VU University in Amsterdam. The thesis has never before been a subject of any procedure of obtaining an academic degree.

Moreover, I declare that the present version of the thesis is identical to the attached electronic version.

Date Author's signature

Contents

Introduction

Through the years several methods have been proposed for managing optimization issues. In plenty of different areas such as finance, logistics, telecommunications, industrial production or personal planning, problems may take a form of Mathematical Programming.

The optimization problems mentioned above are frequently classified based on the types of the decision variables, constraints, and the objective function. The most classical one, with efficient solution algorithms, is called Linear Programming (LP), where all constraints, and the objective function have a linear character, while all variables are usually non-negative and real-valued, thus it is said that the variables have continuous domains. Many real-life problems fit this category very well. Problems with even millions of variables and constraints are routinely solved with commercial mathematical programming software like Xpress-Optimizer, which will be used to do calculations in our models.

It has quickly occurred that continuous variables are not sufficient to represent decisions of a discrete nature. This brought researchers to the development of Mixed Integer Programming (MIP). This is a very similar technique to the LP, but all variables may have either discrete or continuous domains. The other technique is called Integer Programming (IP), with all integer (discrete) variables. A method of solution of such problems contains LP techniques coupled with Branch-and-Bound (an enumeration technique). However, using these methods even for small problem instances might lead to a "computational explosion".

In recent years, algorithmic improvements, growing computer speed have enabled tackling even larger problems, so that modeling real-world situations can be done with greater accuracy. Though solution techniques of mathematical programming is probably the best researched area of optimization, postoptimality analysis has obtained much fewer attention. However, it deals with much more information about a single optimal solution, than just a solution itself.

Coefficients, right-hand side values within LP problems, are assumed to be "real" numbers, in practise they commonly occur to fluctuate in a certain range. There are several ways to cope with such a problem. One of them is the Stochastic Programming (SP) approach, where based on the historical data the parameter distribution should be estimated. However, it may still not provide us with a guarantee, that the current solutions behave as the estimated distribution does.

The other way to assess all possible ranges of the uncertain parameters is Sensitivity Analysis. It is simply an analysis of changes of an optimal solution, resulted by alterations of the input data. This is mostly done by computation of parameters changes, for which an optimal solution stays unchanged.

Sensitivity anaysis constitutes perhaps the most important issue of nowadays applied modelling. Knowledge about perturbations in the problem data is necessary to effectively manage uncertainty. Since sensitivity analysis is used to reveal factors, on which the solution depends mostly, it is essential to know what to focus on during collecting the information, and it indicates issues that should be closely watched by decision makers.

In this paper we will introduce some optimization methods and techniques dealing with sensitivity analysis for Linear Programming (Chapter 1), Integer and Mixed-Integer Programming (Chapter 2), and Multicriteria Optimization (Chapter 3), that have been developed in previous years by researchers. These considerations will lead us to an application of MIP sensitivity analysis within a call center shift-scheduling problem (Chapter 4).

It is commonly known that the solution of the LP dual, points out the sensitivity of the optimal objective function value to alterations in the right-hand sides of the inequality constraints. Results obtained in LP theory stands a core of the development for other optimization branches. Influenced by LP, duality theories were introduced for discrete optimization problems, which became the basis for the sensitivity analysis. However, due to the problem growth, dual solutions in Integer Programming are very complex, and rarely used.

A new approach was introduced by Dawande and Hooker. They have developed the inference dual theory of an optimization problem, which might be used effectively for sensitivity analysis both in LP and IP theory.

Inferring the best possible bound on the optimal value from the constraint set is an assignment of the inference dual. Since the solution of a dual is a proof, obtained by usage of inference method, sensitivity analysis might be regarded as a contribution analysis of each constraint in the proof.

There are two conditions of the inference dual solution that should be satisfied. First, all inference rules stemming from the constraint set in the problem should be recognized, and secondary those rules should be used to prove optimality.

As the paper was built on the knowledge basically introduced by scientists in the past, there are few areas of the author's contribution. First of all, it is shown in Chapter 4, by constructing an example, that in general the set $T_{x^*}^W$ of all weak efficient solutions is not equal to the closure of the set of all efficient solutions clT_{x^*} in the Multi-Objective Linear Programming (MOLP) problem:

$$clT_{x^*} \neq T_{x^*}^W$$

Furthermore, inspired by an approach of Mavrotas & Diakoulaki of generating the so-called incumbent list of all efficient solutions within Multicriteria Branch & Bound Method (MCBB), and an approach of Dawande & Hooker within a MIP primal sensitivity analysis, we propose a MIP analogous with usage of usual Branch & Bound (B&B) technique to generate all possible solutions, by solving the main MIP problem in the first step, and afterwards deriving at each feasible node of a B&B search tree, all other possible efficient solutions. In that way the decision maker is provided with the list of all efficient solutions, not only with the first one achieved by the software.

Notation

Since there are some special abbreviations, which are referred to throughout the entire paper, we introduce them now:

LP	–	Linear Programming
IP	–	Integer Programming
MIP	–	Mixed-Integer Programming
CP	–	Constraint Programming
MOLP	–	Multiple Objective Linear Programming
MOIP	–	Multiple Objective Integer Programming
MOMIP	–	Multiple Objective Mixed-Integer Programming
B&B	–	Branch-and-Bound Method
MCBB	–	Multicriteria Branch-and-Bound Method
SA	–	Sensitivity Analysis

Chapter 1

Linear Programming

There are a lot of reasons, why Linear Programming (LP) still is a strongly considerable field of optimization. Many real-life problems can be directly solved by LP, or has its own importance in solving sub-problems (Branch & Bound).

A vast number of the central concepts (duality, decomposition, importance of convexity, etc) of optimization theory have been influenced by LP.

Furthermore, we may find application of LP in industrial environments, economics or company management. Moreover, the presence of LP algorithms is noticed in planning, scheduling, technology and other branches.

In this chapter we present only the most important facts from LP theory. All theory is introduced without a proof as a commonly well-known theory, which stands as the foundation for further considerations. This chapter is entirely based on the book of Robert J. Vanderbei called *Linear Programming: Foundations and Extensions* [26].

1.1. Preliminaries

Primal Problem

Each linear problem of the following form is commonly referred to as the *primal* problem:

$$\begin{array}{ll} \min & c^T x \\ & Ax \geq b \\ & x \in \mathbb{R}^n_+ \end{array} \tag{1.1}$$

$$\text{where:} \quad c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \text{ and } b \in \mathbb{R}^m.$$

This can be written equivalently in a standard form as a:

$$\begin{array}{lll} \min & \sum_{j=1}^{n} c_j x_j & \\ & & \\ & \sum_{j=1}^{n} a_{ij} x_j \geq b_i & i = 1, \ldots, m \\ & x_j \geq 0 & j = 1, \ldots, n. \end{array}$$

Dual Problem

Associated with the primal problem is the so-called *dual* problem:

$$\begin{array}{ll} \max & y^T b \\ & \\ & y^T A \leq c^T \\ & y \in \mathbb{R}^m \\ & \\ \text{where:} & c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \text{ and } b \in \mathbb{R}^m, \end{array} \tag{1.2}$$

which analogously has its own standard form:

$$\begin{array}{lll} \max & \sum_{i=1}^{m} y_i b_i & \\ & & \\ & \sum_{i=1}^{m} y_i a_{ij} \leq c_j & j = 1, \ldots, n. \end{array}$$

Slack Variables

Let us now introduce slack variables:

$$x_{n+i} = w_i = \sum_{j=1}^{n} a_{ij} x_j - b_i \quad i = 1, \ldots, m.$$

and:

$$y_{m+j} = z_j = c_j - \sum_{i=1}^{m} y_i a_{ij} \quad j = 1, \ldots, n.$$

This transfers primal problem (1.1) into:

$$\begin{aligned} \min \quad & c^T x \\ & \\ & A - w = b \\ & x, w \in \mathbb{R}^n_+, \end{aligned}$$

and dual problem (1.2) into:

$$\begin{aligned} \max \quad & y^T b \\ & \\ & y^T A + z = c^T \\ & y, z \in \mathbb{R}^m. \end{aligned}$$

Basic notation

Let us introduce:

$\mathcal{B} = \{j_1, \ldots, j_m\}$	$-$the set of basis indexes, we call this the base
$\mathcal{N} = \{j_{m+1}, \ldots, j_{m+n}\}$	$-$the set of non-basis indexes
$\overline{c} = c - c_B A_B^{-1} A$	$-$vector of optimality indicators

moreover:

A_B	–	submatrix of A, associated with $\mathcal{B}$
A_N	–	submatrix of A without basis columns
x_B	–	vector of basis variables of primal (1.1)
x_N	–	vector of non-basis variablesof primal (1.1)
y_B	–	vector of basis variables of dual (1.2)
y_N	–	vector of non-basis variables of dual (1.2)
c_B	–	basis columns of a cost vector c
c_N	–	non-basis columns of a cost vector c

Definition 1. *We call the vector [x_B, x_N] the basis feasible solution with respect to $\mathcal{B}$ of a primal problem (1.1) if:*

$$x_B = A_B^{-1} b \geq 0,$$
$$x_N = 0.$$

We call the vector [y_B, y_N] the basis feasible solution with respect to $\mathcal{B}$ of a dual problem (1.2) if:

$$y_B = 0,$$
$$y_N = (A_B^{-1} A_N)^T c_B - c_N.$$

1.2. Convex Analysis

Before we will go further in our considerations, let us step back to the beginning of optimality theory, namely convex analysis.

Definition 2 (Convex Combination). *Given a finite set of points: $x_1, x_2, \ldots, x_n$, in $\mathbb{R}^n$, a point $x \in \mathbb{R}^n$ is called a convex combination of these points if:*

$$x = \sum_{j=1}^{n} \lambda_j x_j \quad \textit{and} \quad \sum_j \lambda_j = 1$$

Furthermore, it is called a strict convex combination if none of the $\lambda_j^{\prime}s$ vanish.

Definition 3 (Convex Set). *If for every x and y in a subset $S \in \mathbb{R}^m$, this subset also contains all points on the line connecting x and y, then S is convex. That is:*

$$\theta x + (1 - \theta) y \in S, \textit{for every } 0 < \theta < 1.$$

Theorem 1. *A set C contains all convex combinations of points in C, if and only if the set C is convex as well.*

For each set $S \in \mathbb{R}^m$, there exists a smallest convex set containing S, called the *convex hull* of S and is denoted by $conv(S)$.

Definition 4 (*Convex hull* of S)**.** *The intersection of all convex sets containing S.*

This definition is equivalent to the following theorem:

Theorem 2. *The convex hull conv(S) of a set S in $\mathbb{R}^m$ consists precisely of the set of all convex combinations of finite collections of points from S.*

1.2.1. Carathéodory's Theorem

In 1907 Carathéodory showed that to define the convex hull of S, it is sufficient to use $m+1$ points:

Theorem 3 (Carathéodory's Theorem)**.** *The convex hull conv(S) of a set $S \in \mathbb{R}^m$ consists of all convex combinations of $m+1$ points from S:*

$$conv(S) = \left\{ z = \sum_{j=1}^{m+1} \lambda_j z_j : \; z_j \in S \text{ and } \lambda_j \geq 0 \text{ for all } j, \; \sum_j \lambda_j = 1 \right\}.$$

1.2.2. The Separation Theorem

Definition 5 (Halfspace)**.** *Any set given by a single (nontrivial) linear inequality in $\mathbb{R}^n$ is a halfspace of $\mathbb{R}^n$:*

$$x \in \mathbb{R}^n : \sum_{j=1}^{n} a_j x_j \leq b, \quad b \in \mathbb{R}, \quad (a_1, \ldots, a_n) \neq 0.$$

Definition 6 (Generalized halfspace)**.** *If the coefficients $(a_1, a_2, \ldots, a_n)$ vanish in the definition of a halfspace, then such a set is called a generalized halfspace.*

Every generalized halfspace is simply a halfspace, all of $\mathbb{R}^n$, or the empty set, and thus it is convex as well.

Definition 7 (Polyhedron)**.** *An intersection of a finite collection of generalized halfspaces is called a polyhedron. A polyhedron is any set of the form:*

$$x \in \mathbb{R}^n : \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \dots, m$$

As every polyhedron is an intersection of a collection of convex sets, it is obviously convex.

Theorem 4 (The Separation Theorem for polyhedra)**.** *Let P and $\bar{P}$ be two disjoint nonempty polyhedra in $\mathbb{R}^n$. Then there exist disjoint halfspaces H and $\bar{H}$ such that $P \subset H$ and $\bar{P} \subset \bar{H}$.*

1.2.3. Farkas' Lemma

Farkas' Lemma plays a fundamental role in the proof of the separation theorem, and constitutes one of the basic theorems of optimality theory.

Theorem 5 (Farkas' Lemma)**.** *The system $Ax \leq b$ has no solutions if and only if there is a y such that:*

$$A^T y = 0, \quad y \geq 0, \quad b^T y < 0.$$

1.3. The Simplex Method

The simplex method, developed by George Dantzig, constitutes one of the most important algorithms in optimization theory nowadays. It solves LP problems by constructing a feasible solution at a vertex of the polyhedron and then walking along a path on the edges of the polyhedron to vertices with non-decreasing values of the objective function until an optimum is reached. In other words, the simplex method produces a sequence of steps to "adjacent" bases such that the current value of the objective function increases at each step.

We begin with the so-called *starting dictionary*, which initially has m basis and n nonbasis variables. As the simplex method goes further, it moves from one dictionary to another whilst searching for an optimal solution.

During iterations of the simplex method, exactly one variable enters $\mathcal{B}$, chosen from $\mathcal{N}$ as to increase the objective function value. Usually it means picking a variable of index k, which has the largest coefficient. However, if the set $\mathcal{N}$ is empty, we have obtained an optimal solution. Furthermore, at the same time there is one variable that leaves the basis $\mathcal{B}$, chosen to preserve nonnegativity of the current basis variables.

It is required that:

$$\bar{b}_i - \bar{a}_{ik} x_k \geq 0, \quad i \in \mathcal{B}.$$

This means that the only variables that can become negative while x_k increases are those for which $\bar{a}_{ik}$ is positive. As we do not want any of the variables to become negative, the following equality seems to be reasonable:

$$x_k = \min_{\{i \in \mathcal{B} : \bar{a}_{ik} > 0\}} \frac{\bar{b}_i}{\bar{a}_{ik}}$$

The above mentioned variable selection rule might be altered, since we wish to take the largest possible increase in x_k:

$$x_k = \left(\max_{i \in \mathcal{B}} \frac{\bar{a}_{ik}}{\bar{b}_i} \right)^{-1}.$$

As we have chosen the leaving and entering variables to achieve an interchange, there must be some appropriate row operations involved, called pivot rules.

Degeneracy, cycling, unboundedness

When a denominator in one of the above mentioned ratios vanishes and the numerator is nonzero, then the ratio should be regarded as $+\infty$ or $-\infty$. Furthermore, if the numerator is positive we deal with a degenerate pivot.

The most undesirable issue is cycling, when the simplex method makes a sequence of degenerate pivots and eventually returns to an initial feasible solution. Such infinite loops never find an optimal solution. In such a situation all the pivots within the cycle must be degenerate, since the objective function value never decreases. However, in practice, degeneracy is very common, but cycling is rare.

When all of the ratios are nonpositive, as the entering variable increases, none of the basis variables will become zero. Then the problem becomes unbounded.

Auxiliary problem

When we set each x_j to zero, then we obtain the solution, which is feasible if and only if all the right-hand sides are nonnegative. If they are not, an auxiliary problem must be introduced :

$$\begin{array}{lll} \textit{maximize} & -x_0 & \\ & \sum_{j=1}^{n} a_{ij}x_j - x_0 \leq b_i & i = 1, \ldots, m \\ & x_j \geq 0 & j = 1, \ldots, n. \end{array} \tag{1.3}$$

If now we set $x_j = 0$, for $j = 1, \ldots, n$, and then pick x_0 sufficiently large, then we obtain a feasible solution to the auxiliary problem. Additionally the objective value of this solution is equal to zero, if the original problem has a feasible solution. The solving process of the auxiliary problem to find an initial feasible solution is commonly called *Phase I.*

Now we are able to continue applying the simplex method until an optimal solution is reached, this process is called *Phase II.*

Fundamental Theorem of Linear Programming

Theorem 6 (Fundamental Theorem of Linear Programming)**.** *For an arbitrary linear program, the following statements are true:*
— *If there is no optimal solution, then the problem is either infeasible or unbounded.*
— *If a feasible solution exists, then a basis feasible solution exists.*
— *If an optimal solution exists, then a basis optimal solution exists.*

1.4. Duality theory

One of the most important issues in optimization is duality theory. In LP, the primal problem (1.1) and the dual problem (1.2) are complementary. Thanks to duality a solution to either one determines a solution to both. In other words, it means that the dual of the dual is simply the primal problem. Each feasible solution for one of these two, gives a bound on the optimal objective function value for the other. These bounds, however, leave a gap (within which the optimal solution lies), but they are better than nothing.

The Weak Duality Theorem

The dual problem in general provides tighter bounds for the primal objective function value, which is summarized in the following theorem:

Theorem 7 (Weak Duality Theorem). *Let $x = (x_1, \ldots, x_n)$ be a feasible solution of primal problem (1.1), analogously let $y = (y_1, \ldots, y_m)$ be a feasible solution of dual problem (1.2). Then:*

$$y^T b \leq c^T x.$$

If we imagine the real line consisting of all possible values for the primal objective function, and visualize analogous situations for the dual problem, we would see that the set of primal values lie entirely to the right of the set of dual values.

The Strong Duality Theorem

In fact, for linear programming there is no gap between the optimal objective function value for the primal and for the dual, which stands as a convenient tool for verifying optimality.

Theorem 8 (Strong Duality Theorem). *If $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is an optimal solution of primal problem (1.1),then dual problem (1.2) also has an optimal* *solution* $y^* = (y_1^*, \ldots, y_m^*)$*, such that:*

$$y^{*T} b = c^T x^*.$$

A very important issue raises when primal problem (1.1) does not have an optimal solution. Let us suppose it is unbounded. This fact together with the weak duality theorem shows that dual problem (1.2) must be infeasible.

Analogously, when dual problem (1.2) is unbounded, then primal problem (1.1) is infeasible. However sometimes it occurs that both the primal and the dual problems are infeasible. These considerations lead to the following theorem:

Theorem 9 (Duality Theorem)**.** *Only one of the following conditions occurs:*

1. *Both the primal (1.1) and dual problem (1.2) have optimal solutions and:*

$$c^T x = y^T b.$$

2. *The primal problem does not have any feasible solution,* $X = \emptyset$*, the dual problem has a feasible solution, but not optimal as:*

$$\sup\{y^T b \quad : y \in Y\} = +\infty.$$

3. *The dual problem does not have any feasible solution,* $Y = \emptyset$*, the primal problem has a feasible solution, but not optimal as:*

$$\inf\{c^T x \quad : x \in X\} = -\infty.$$

4. *Neither the primal, nor the dual problem has a feasible solution:*

$$X = \emptyset, Y = \emptyset.$$

Thanks to duality theory it is easy to provide a certificate of optimality. It must be checked that the primal solution is feasible for the primal problem, the dual solution is feasible for the dual problem, and that the primal and dual objective values agree.

The simplex method applied to primal problem (1.1) in fact solves both the primal and the dual (1.2). As the dual of the dual is the primal, during solution procedure with respect to the dual, the simplex method solves both the primal and the dual problem.

When we notice that the dual has an obvious basis feasible solution, it might be easier to apply the simplex method to the dual problem, instead of the primal one.

1.4.1. Complementary Slackness

Now we will consider primal problem (1.1) with its slack variables:

$$\begin{aligned} \min \quad & c^T x \\ & Ax - w = b \\ & x, w \geq 0. \end{aligned} \tag{1.4}$$

The dual (1.2) can be written as follows:

$$\begin{aligned} \max \quad & y^T b \\ & y^T A + z = c^T \\ & y, z \geq 0. \end{aligned} \tag{1.5}$$

Sometimes there might occur a situation, when it is necessary to obtain an optimal solution for the dual problem, when only an optimal solution for the primal one is known. This issue is a subject of the following theorem.

Theorem 10 (The Complementary Slackness Theorem)**.** *Suppose that $x = (x_1, \ldots, x_n)$ is a primal feasible solution and that $y = (y_1, \ldots, y_m)$ is a dual feasible solution. Let $w = (w_1, w_2, \ldots, w_m)$ denote the corresponding primal slack variables, and let $z = (z_1, z_2, \ldots, z_n)$ denote the corresponding dual slack variables. Then x and y are optimal for their respective problems if and only if:*

$$x_j z_j = 0, \quad \text{for } j = 1, \ldots, n$$

$$w_i y_i = 0, \quad \text{for } i = 1, \ldots, m.$$

Suppose that a nondegenerate basis optimal solution for the primal is known:

$$x^* = (x_1^*, x_2^*, \ldots, x_n^*),$$

and we want to obtain a corresponding optimal solution for the dual.

Let:

$$w^* = (w_1^*, w_2^*, \ldots, w_m^*),$$

denote the corresponding primal slack variables, which might be easily obtained from their definition as slack variables:

$$w_i^* = \sum_j a_{ij} x_j^* - b_i.$$

The dual (1.2) constraints are:

$$\sum_i y_i a_{ij} + z_j = c_j, \quad j = 1, \ldots, n$$

where we have written the inequalities in equality form by introducing slack variables $z_j, j = 1, 2, \ldots, n$.

These constraints form n equations with $m + n$ unknown variables. However, the basis optimal solution (x^*, w^*) is a collection of $n + m$ variables, many of which are positive. In fact, since the solution of the primal (1.1) is assumed to be nondegenerate, it follows that the m basic variables will be strictly positive.

Then from the Complementary Slackness Theorem it is known that the corresponding dual variables must vanish. As we may set m of $m + n$ variables to zero, there are just n equations left with n unknown variables. Thus there should be a unique solution, and all its components should be nonnegative, since we assumed optimality of x^*.

Thus if (x^*, w^*) denotes an optimal solution to the primal and (y^*, z^*) denotes an optimal solution to the dual, then the *Complementary Slackness Theorem* says that, for each $j = 1, \ldots, n$, either $x_j^* = 0$ or $z_j^* = 0$ (or both).

Strict Complementarity

As a matter of fact, there are optimal pairs of solutions, for which exactly one member of each pair (x_j^*, z_j^*) vanishes and exactly one member from each pair (y_j^*, w_j^*) vanishes.

Then the optimal solutions are *strictly complementary* to each other. This is often expressed by $x^* + z^* > 0$ and $y^* + w^* > 0$.

Theorem 11. *If both the primal (1.1) and the dual (1.2) have feasible solutions, then there exists a primal feasible solution* $(\bar{x}, \bar{w})$ *and a dual feasible solution* $(\bar{y}, \bar{z})$ *such that* $\bar{x} + \bar{z} > 0$ *and* $\bar{y} + \bar{w} > 0$.

As to make a linear programming problem feasible there must exist a variable x_j that vanishes, the so-called null variable. If such a variable exists, then its dual slack is not null.

Theorem 12 (The Strict Complementary Slackness Theorem)**.** *If a linear programming problem has an optimal solution, then there is an optimal solution* (x^*, w^*) *and an optimal dual solution* (y^*, z^*) *such that:*

$$x^* + z^* > 0 \text{ and } y^* + w^* > 0.$$

1.5. Sensitivity Analysis

The dual solution provides a partial sensitivity analysis. Let y^{*T} be the optimal solution of the dual problem (1.2), which makes $f(y^*) = y^{*T}b$ the optimal value for both the primal (1.1) and the dual. If $Ax \geq b + \Delta b$ in (1.1), then the dual becomes:

$$\begin{aligned} \max \quad & y^T(b + \Delta b) \\ & y^T A \leq c^T \\ & y \in \mathbb{R}^m \\ \text{where:} \quad & c \in \mathbb{R}^n, A \in \mathbb{R}^{n \times m}, \text{ and } b \in \mathbb{R}^m. \end{aligned} \tag{1.6}$$

As only the objective function changes, y^{*T} is feasible as well in (1.6). Therefore $y^T(b + \Delta b)$ is a lower bound to the optimal value of the perturbed primal problem. The possibly negative change in the optimal value $y^{*T}b$ of the original problem is bounded below by $y^{*T}\Delta b$. The change is in fact equal to $y^{*T}\Delta b$, if the perturbation Δb lies within easy computational ranges.

Let us assume that a problem has been solved to optimality, therefore we have at our disposal the final optimal dictionary. Suppose now we wish to change the objective coefficients from c to, say, $\bar{c}$. It is natural to ask how the dictionary at hand could be adjusted to become a valid dictionary for the new problem. That is, we want to maintain the current classification of the variables into basis and nonbasis variables and simply adjust $f(x^*), x^*_B$ and y^*_N appropriately.

Recall that:

$$
\begin{array}{lcl}
x_B^* & = & A_B^{-1} b, \\
y_N^* & = & (A_B^{-1} A_N)^T c_B - c_N, \\
f(x^*) & = & c_B^T A_B^{-1} b.
\end{array}
\tag{1.7}
$$

Hence, the change from c to $\overline{c}$ requires us to recompute $y_N^*, f(x^*)$, but x_B^* remains unchanged. Therefore, after recomputing the new dictionary the primal is still feasible, and so there is no need for a Phase I procedure: we can jump straight into the primal simplex method, and if $\overline{c}$ is not too different from c, we can expect to get to the new optimal solution in a relatively small number of steps.

Now suppose that instead of changing c, we wish to change only the right-hand side b. In this case, we see that we need to recompute $x_B^*, f(x^*)$, but y_N^* remains unchanged. Hence, the new dictionary will be dual feasible, and so we can apply the dual simplex method to arrive at the new optimal solution fairly directly.

Therefore, changing just the objective function or just the right-hand side results in a new dictionary having nice feasibility properties.

What if we need/want to change some (or all) entries in both the objective function and the right-hand side and maybe even the constraint matrix too?

In this case, everything changes: $y_N^*, f(x^*)$, and x_B^*. Even the entries in $\mathcal{B}$ and $\mathcal{N}$ change.

Nonetheless, as long as the new basis matrix A_B is nonsingular, we can make a new dictionary that preserves the old classification into basic and nonbasic variables. The new dictionary will most likely be neither primal feasible nor dual feasible, but if the changes in the data are fairly small in magnitude, one would still expect that this starting dictionary will get us to an optimal solution in fewer iterations than simply starting from scratch.

While there is no guarantee that any of these so-called warm-starts will end up in fewer iterations to optimality, extensive empirical evidence indicates that this procedure often makes a substantial improvement: sometimes the warm-started problems solve in as little as one percent of the time it takes to solve the original problem.

Ranging

Often one does not wish to solve a modification of the original problem, but instead just wants to ask a hypothetical question:

If one is to change the objective function by increasing or decreasing one of the objective coefficients a small amount, how much could it increase/decrease without changing the optimality of the current basis?

To study this question, let us suppose that c gets changed to $c + t\Delta c$, where t is a real number and c is a given vector (which is often all zeros except for a one in a single entry, but we do not need to restrict the discussion to this case). It is easy to see that y_N^* gets incremented by

$$t\Delta y_N,$$

where:

$$\Delta y_N = (A_B^{-1} A_N)^T \Delta c_B - \Delta c_N.$$

Hence, the current basis will remain dual feasible as long as

$$y_N^* + t\Delta y_N \geq 0.$$

We have manipulated this type of inequality many times before, and so it should be clear that, for $t > 0$, this inequality will remain valid as long as:

$$t \leq \left(\max_{j \in N} -\frac{\Delta y_j}{y_j^*} \right)^{-1}.$$

Similar manipulations show that, for $t < 0$, the lower bound is:

$$t \geq \left(\min_{j \in N} -\frac{\Delta y_j}{y_j^*} \right)^{-1}.$$

Combining these two inequalities, we see that t must lie in the interval:

$$\left(\min_{j\in N} -\frac{\Delta y_j}{y_j^*}\right)^{-1} \leq t \leq \left(\max_{j\in N} -\frac{\Delta y_j}{y_j^*}\right)^{-1}.$$

Now suppose we change b to $b + t\Delta b$ and ask how much t can change before the current basis becomes nonoptimal. In this case, y_N^* does not change, but x_B^* gets incremented by $t\Delta x_B$, where:

$$\Delta x_B = A_B^{-1}\Delta b.$$

Hence the current basis remains optimal as long as t lies in interval:

$$\left(\min_{i\in B} -\frac{\Delta x_i}{x_i^*}\right)^{-1} \leq t \leq \left(\max_{i\in B} -\frac{\Delta x_i}{x_i^*}\right)^{-1}.$$

Chapter 2

Integer Programming

The LP models are continuous, in the meaning that all decision variables are allowed to be fractional. However, fractional solutions sometimes are undesirable, and we must consider the following problem:

$$
\begin{aligned}
\min \quad & c^T x \\
& Ax \geq b \\
& x \in \mathbb{Z}_+^n \\
\text{where:} \quad & c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \text{ and } b \in \mathbb{R}^m.
\end{aligned}
\tag{2.1}
$$

This problem is called an *integer-programming problem* (IP). When at least one of the decision variables becomes fractional, the problem becomes a *mixed integer programming* (MIP) model. Without loss of generality, it might be assumed that the variables with indexes from 1 to $p \leq n$ are integer, thus the MIP primal problem takes the form of:

$$
\begin{aligned}
\min \quad & c^T x \\
& Ax \geq b \\
& x \in X' = \mathbb{Z}_+^p \times \mathbb{R}^{n-p} \\
\text{where:} \quad & c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \text{ and } b \in \mathbb{R}^m
\end{aligned}
\tag{2.2}
$$

Integer programming is present in all areas of decision making nowadays. It has particular contribution to optimization problems as: facility location, all kinds of scheduling problems, routing, communication networks, capital budgeting, project selection, or analysis of capital development alternatives. As a matter of fact, there are several applications of IP in science: physics, genetics, medicine, and even in engineering.

2.1. Solution Methods

Sometimes the solution of the primal IP problem (2.1) might be obtained, by ignoring the integrality restrictions and deriving the solution to the resulting linear program (1.1), the so-called *linear relaxation* of IP. However it is possible only when all the basis solutions of (1.1) are integer.

The feasible constraint set $Ax \geq b,\ x \geq 0$, for an arbitrary integer vector b have only integer basic solutions, if and only if the matrix A is totally *unimodular*, which means that all square submatrices of A have a determinant equal to 0, 1 or -1.

In general some of the variables in the LP solution will still be fractional, thus other methods should be undertaken to obtain a pure IP solution.

Since LP is less constrained than IP, there are some conclusions:

— If we consider primal IP (2.1), then the optimal objective value for the primal LP (1.1) is less than or equal to the optimal objective for a primal IP (an upper bound).

— If there is considered dual IP, then the optimal objective value for dual LP problem (1.2) is greater than or equal to that of the primal IP (a lower bound).

— If the LP primal (dual) is infeasible, then so is the IP primal (dual).

— If the LP primal (dual) is optimized by integer variables, then that solution is feasible and optimal for the IP primal (dual).

— If the objective function coefficients are integer, then the optimal objective for primal IP (2.1) is greater than or equal to the "round up" of the optimal objective for the primal LP (1.1). The optimal objective for dual IP (unknown) is less than or equal to the "round down" of the optimal objective for the dual LP (1.2).

In fact, solving the LP relaxation does give some information: it gives a bound on the optimal value, and, if we are lucky, it may give the optimal solution to the IP. However, rounding the solution of the LP will not in general give the optimal solution of the IP. In fact, for some problems it is difficult to round and even get a feasible solution.

There is no single technique for solving integer programs. Instead, a lot of procedures have been developed. We indicate three possible approaches:

1. enumeration techniques, including the branch-and-bound procedure;
2. cutting-plane techniques; and
3. group-theoretic techniques.

There exists plenty of other procedures, which are simply a mixed-modification of above mentioned approaches.

The cutting-plane technique was the first one introduced for IP, for which the convergence in a finite number of moves, was proved[29]. The algorithm solves integer programs by modifying LP solutions until the integer one is reached. By adding new constraints a single LP is being redefined, which successively reduces the feasible region until an integer optimal solution is found.

However, in practice, the Branch & Bound procedure almost always outperform the cutting-plane algorithm. Therefore in this paper we will deal with the much more efficient one.

2.2. Branch & Bound

Branch & Bound might be seen as a strategy of “divide and conquer.” The main issue is to divide the feasible region to develop bounds $LB \leq z^* \leq UB$ on z^*, and if it is still required, further partitioning.

Basic Intuitions

Let us consider primal IP problem (2.1), corresponding to the so-called *root node* of a tree. Let X denote the constraint set of this problem:

$$X = \{x \in \mathbb{Z}^n_+ : \; Ax \geq b\}.$$

The search tree is constructed in an iterative way. *By branching* on an existing node, for which the optimal solution of the LP relaxation is fractional, which means that some of the integer restricted variables have fractional values, the new node is formed.

Next by selection of the fractional valued variable and adding proper constraints in each child subproblem, two child nodes are formed.

The essence of the B&B algorithm is as follows:

1. Solve the linear relaxation of the problem. If the solution is integer, then we are done. Otherwise create two new subproblems by branching on a fractional variable.

2. A subproblem is not active when any of the following occurs:

   a) The subproblem was used to branch on,

   b) All variables in the solution are integer,

   c) The subproblem is infeasible,

   d) The subproblem might be fathomed by a bounding argument.

3. Choose an active subproblem and branch on a fractional variable. Repeat until there are no active subproblems.

The B&B algorithm is a search tree, of which leaves represent a partition, and their nodes represent subpolyhedra that were further subdivided. Consider now a partition of X into the subpolyhedra $L_1, \dots, L_s$ in such way that $X \subset \sum_{i=1}^{s} L_i$ and assume that these subpolyhedra are nonempty. Let LP_i be the linear program $\min_{x^i \in X_i} c^T x^i$ associated with subpolyhedron L_i. The B&B optimality conditions are presented formally by the following theorem:

Theorem 13 (Ralphs, Guzelsoy, 2004)**.** *Let B^i be the optimal basis for LP^i. Let*

$$U = \min \{c_{B^i}(A_{B^i})^{-1}b + \beta_i | 1 \le i \le s, \hat{x}^i \in X'\}$$

and

$$L = \min \{c_{B^i}(A_{B^i})^{-1}b + \gamma_i | 1 \le i \le s\},$$

where β_i and γ_i are constant factors associated with nonbasis variables fixed at nonzero bounds and $\hat{x}^i$ is the BFS corresponding to basis B^i. If $U = L$, then $z^* = U$ and for each $1 \le j \le s$ such that $\hat{x}^j \in X'$ and $c_{B^j}(A_{B^j})^{-1}b = z^*$, $\hat{x}^j$ is an optimal solution.

Fathoming

There are some criteria with respect to which the L_j region need not to be subdivided. This occurs when:

1. *Fathoming by infeasibility*
   the LP over L_j is infeasible;
2. *Fathoming by integrality*
   the optimal LP over L_j is integer;
3. *Fathoming by bounds*
   the value of the LP solution z^j over L_j satisfies $z^j \leq \underline{z}$ (while maximizing.)

Mathematical Algorithm

We introduced some intuitive rules of the B&B algorithm, now let us introduce the general mathematical idea of it [20]. First of all, let us introduce some useful notation. Let L denote the list of active subproblems $\{IP\}^i$, where $IP^0 = IP$ denotes the original integer program. Let $\overline{z}_i$ denote an upper bound on the optimal objective value of IP^i, and let z_{IP} denote the *incumbent* objective value (the objective value corresponding to the current best integral feasible solution of IP). The algorithm is as follows:

1. (*Initialization*): Set $L = \{IP^0\}$, $\overline{z}_0 = -\infty$, and $z_{IP} = +\infty$.

2. (Termination): If $L = \emptyset$, then the solution z^*, which yielded the incumbent objective value z_{IP} is optimal. If no such z^* exists, then IP is infeasible.

3. (Problem selection and relaxation): Select and delete a problem IP^i from L. Solve a relaxation of IP^i. Let z_i^R denote an optimal objective value of the relaxation, and let x^{iR} be an optimal solution if one exists, which means that: $z_i^R = C^T x^{iR}$ or $z_i^R = \infty$.

4. (Fathoming and Pruning):
   — If $z_i^R \leq z_{IP}$ go to Step 2.

   — If $z_i^R > z_{IP}$ and x^{iR} is integral feasible, update $z_{IP} = z_i^R$. Delete from L all problems with $\overline{z}_i \leq z_{IP}$. Go to Step 2.

5. (Partitioning): $\{X^{i,j}\}_{j=1}^{k}$ be a partition of the constraint set X^i of problem IP^i. Add problems $\{IP^{ij}\}_{j=1}^{k}$ to L, where IP^{ij} is IP^i with feasible region restricted to X^{ij} and $\overline{z}_{ij} = z_i^R$ for $j = 1, \ldots, k$. Go to Step 2.

When we go from an original region to one of its subdivisions, we need to add one constraint. This constraint is not satisfied by an optimal solution of the LP relaxation over the original region, and this brings a motivation for the dual simplex algorithm use in here.

When a new constraint is added with its own slack basic variable, we begin with a starting solution for the dual-simplex algorithm, which actually has only one negative basis variable.

Usually, only a few dual-simplex computations are needed to reach an optimal solution. The primal-simplex algorithm would require many more pivoting operations.

2.2.1. Partitioning Strategies within the B&B Algorithm

Suppose x^R is an optimal solution to the linear relaxation of a tree node. Throughout the years there were some partitioning strategies proposed:

— *Variable dichotomy.*

When x_j^R is fractional, two new nodes are created in following way:

$$x_j \leq \lfloor x_j^R \rfloor \quad \text{and} \quad \lceil x_j^R \rceil \leq x_j,$$

where $\lfloor \cdot \rfloor$, $\lceil \cdot \rceil$ denote, respectively, the floor and the ceiling of the real number.

— *Generalized-Upper-Bound Dichotomy (GUB Dichotomy).*

When in the original IP program there is a following constraint $\sum_{j\in Q} x_j = 1$, and $x_i^R, i \in Q$ are fractional. The following partition:

$$Q = Q_1 \cup Q_2,$$

might occur, such that $\sum_{j\in Q_1} x_j^R$ and $\sum_{j\in Q_2} x_j^R$ are approximately of equal value.

Then by setting respectively:

$$\sum_{j \in Q_1} x_j = 0 \quad \text{and} \quad \sum_{j \in Q_2} x_j = 0,$$

two branches can be formed.

— *Multiple branches for bounded integer variable.*

When x_j^R is fractional and $x_j \in \{0, \ldots, l\}$, new nodes might be created $l + 1$ with:

$$x_j = k \quad \text{for node} \quad k = 0, \ldots, l.$$

2.2.2. Branching Variable Selection

During the partitioning process, there must be selected a branching variable to create the children nodes. There are some common approaches:

— *Most/least infeasible integer variable.*

The integer variable, whose fractional value is the farthest from (closest to) an integral value is chosen.

— *Driebeck-Tomlin penalties.*

The penalties are defined as the cost of the dual pivot, which are necessary to remove the fractional variable from the basis. When many pivots are required to restore primal feasibility, these penalties may not be very informative.

When forcing the value of the kth basic variable *up*, the *up* penalty is:

$$u_k = \min_{j : a_{kj} < 0} \frac{(1 - f_k)\overline{c}_j}{-a_{kj}},$$

where: f_k is the fractional part of x_k, $\overline{c}_j$ is the reduced cost of the variable x_j, and the a_{kj} are the transformed matrix coefficients from the kth row of the optimal dictionary for the LP relaxation.

The *down penalty* d_k is calculated as:

$$d_k = \min_{j:a_{kj}>0} \frac{f_k \bar{c}_j}{a_{kj}}.$$

As the penalties are computed, there are several rules of the branching variable selection. For instance:

$$\max_k \max_k (u_k, d_k) \quad \text{or} \quad \max_k \min_k (u_k, d_k).$$

When the LP objective value for the parent node minus the penalty is worse than the incumbent integer solution, a penalty might be used to eliminate a branch.

— *Pseudocost estimate.*

Not just the cost of the first pivot, as with the penalties, but the total cost of all pivots, is reflected by pseudocost. When x_k is a candidate on a branching variable, the pseudocosts are computed as:

$$U_k = \frac{\overline{z}_k - z_k^u}{1 - f_k}$$

and

$$D_k = \frac{\overline{z}_k - z_k^d}{f_k},$$

where $\overline{z}_k$ is the objective value of the parent, z_k^u is the objecting value from forcing up, and z_k^d is the objective value from forcing down.

If a variable has been branched upon repeatedly, an average may be used.

The maximum degradation is computed as follows:

$$D_k f_k + U_k(1 - f_k).$$

and determines the selection of the branching variable.

If the problem has a large percentage of integer variables the pseudocosts are not regarded as beneficial.

— *Pseudoshadow prices.*

Pseudoshadow prices calculate the total cost of forcing a fractional variable to become an integral one. For each constraint and for each integer variable pseudoshadow prices are either given an initial value or specified by the user.

The branching variable is chosen using criteria similar to penalties and pseudocosts.

— *Strong branching.*

Strong branching is applicable to 0-1 IP programs with simplex-based branch and cut settings. The algorithm works as follows:

Let $N, K \in \mathbb{Z}^n_+$. Having a solution for some LP relaxation, make a list of N binary variables that are fractional and close to 0.5.

Let I be an index set of this list. Then, for each $i \in I$ fix x_i first to 0, then to 1 and perform K iterations of the dual simplex method with the steepest edge pricing.

Let $L_i,\ U_i, i \in I$, be the objective values that result from these calculations, where L_i corresponds to fixing x_i to 0, and U_i to fixing it to 1.

Then the selection of a branching variable might be based on the best weighted-sum of both values.

— *Priorities selection.*

Selection of the variables is based on user-assigned priorities, furthermore, priorities may stem from objective function coefficients, or pseudocosts.

2.2.3. Node Selection

There are several strategies of the node selection, which, in fact, may affect the improvement possibilities of an incumbent, the chance of node fathoming, and the total number of problems necessary to solve before optimality is obtained.

— *Depth-first-search with backtracking.*

Choose a child of the previous node as the next node, when it is pruned select the other child. When both of them are pruned, select the most recently created unexplored node, which will be the other child node of the last successful node.

— *Best bound.*

Select the node, with the best LP objective value, among all unexplored nodes. Since nodes might be pruned only if its relaxed objective value is less than the current incumbent, thus the node with the larger LP objective value cannot be pruned.

— *Sum of integer infeasibilities.*

The sum of infeasibilities at a node is calculated as a:

$$s = \sum_j \min(f_j, 1 - f_j).$$

Select the node with either maximum or minimum sum of integer infeasibilities.

— *Best estimate using pseudocosts.*

The individual pseudocost might be used to estimate the resulting integer objective value achievable from node k:

$$\epsilon_k = \overline{z}_k - \sum_i \min(D_i f_i, U_i(1 - f_i)),$$

where $\overline{z}_k$ is the LP relaxation value at node k.

The best estimate indicates which node should be chosen.

— *Best estimate using pseudoshadow prices.*

Pseudoshadow prices can also be used to derive an estimate of the resulting integer objective value achievable from the node, and the best estimate becomes an indicator for node selection.

— *Best projection.*

The projection is defined as an estimate of the objective function value, which is associated with an integer solution, attained by following the subtree that starts at this node.

In particular the projection p_k associated with node k defintion is as follows:

$$p_k = \overline{z}_k - \frac{s_k(\overline{z}_0 - z_{IP})}{s_0},$$

where $\overline{z}_0$ denotes an objective value of the LP at the root node, z_{IP} denotes an estimate of an optimal integer solution, and s_k denotes the sum of integer infeasibilities at node k.

The projection is therefore the weighting between the objective function and the sum of infeasibilities. The weight $\frac{\overline{z}_0 - z_{IP}}{s_0}$ corresponds to the slope of the line between 0 and the node, at which an optimal integer solution is attained.

Let n_k be the number of integer infeasibilities at node k, then a more general projection formula might be provided. Let: $w_k = \mu n_k + (1 - \mu)s_k$, where $\mu \in [0, 1]$ and define:

$$p_k = \overline{z}_k - \frac{w_k(\overline{z}_0 - z_{IP})}{w_0}$$

2.2.4. Preprocessing and Reformulation

Prior to and during B&B there the Problem preprocessing might be applied and reformulation such as:

— Empty rows and columns removal. Implicit bounds and implicit slack variables detection.

— Removal of rows dominated by multiples of other rows.

— Bounds strengthening within rows, having compared individual variables and coefficients with the right-hand sides. Rounding might be used as an additional strengthening for integral variables.

— Determining of the upper and lower bounds for the left-hand side of a constraint, by rounding variables, and next comparison of them with the right-hand side.

— *Aggregation.*

Given an equality constraint for which on some variable there is a bound implied by the satisfaction of the bounds on the other variables, leads to the constraint removal, and the substitution of this variable.

Integral variables might be eliminated only if their integrality is obtained by the integrality of the remaining variables.

— *Coefficient reduction.*

Consider a constraint $\sum_{j \in K} a_j x_j \geq b$ in which all $a_j \geq 0$ and all $x_j \geq 0$.

If x_j is a binary variable and $a_j > b$, for some $j \in K$, replace a_j with b.

2.2.5. Continuous Reduced Cost Implications

At the lower bound one can notice that the reduced costs $\overline{c}_j$ are nonpositive for all nonbasis variables x_j, while at their upper bounds they become nonnegative, given an optimal solution to an LP relaxation with an objective value z_{LP}. When $\underline{z}_{ip}$ describes the objective value for an IP feasible solution, there are few facts:

— If in the LP relaxation x_j is at its lower bound, and additionally

$$z_{LP} - \underline{z}_{ip} \leq -\overline{c}_j,$$

then there exists an optimal IP solution with x_j at its lower bound.

— If in the LP relaxation x_j is at its upper bound, while

$$z_{LP} - \underline{z}_{ip} \leq \overline{c}_j,$$

then there exists an optimal IP solution, for which x_j is at its upper bound.

A reduction of the IP size, might occur as a result of the removal of the fixed variables from the problem when reduced-cost fixing is applied to the root node.

2.2.6. B&B algorithm with efficiency improvement

The question is, if we are able to choose which region should be divided next in order to obtain a near-optimal integer solution z' rapidly?

If so, then we can make immediate elimination of some potential subdivisions. In fact, if in any region the LP value $z \leq z'$, then none of the objective values of integer points in such an area can exceed z', which results in the lack of need of any further subdivisions in this region.

There is no universal procedure for decision-making, however, a lot of heuristic methods have been introduced, for instance: the largest optimal LP value selection.

The effectiveness of the B&B algorithm is hihgly dependent on the selection of the variables for branching. A brief description of a few algorithms used nowadays was given in the *Branching Variable Selection* section.

Derpich and Vera, inspired by Lenstra's algorithm, have used the "Flatness theorem", thanks to which it is possible to bound the width of the feasible region, with respect to the number of integral objective function values in different directions [5].

The B&B method slices the polyhedron in a horizontal manner, the new approach suggests to cut the feasible region, either horizontally, or vertically, but with a preference to the minimum lattice width direction. This requires solving a very difficult problem - the "shortest integral vector" problem.

The authors have used ellipses to approximate the shape of the feasible region. As the size of the ellipse's axes bears information about the thinness of the directions, it would possibly be tied with the minimum width directions of the feasible region.

They have constructed a pair of concentric ellipses:

$$E = \{x \in \mathbb{R}^n : \quad (x - x^0)^T Q(x - x^0) \leq 1\}$$

and:

$$E' = \{x \in \mathbb{R}^n : \quad (x - x^0)^T Q(x - x^0) \leq \gamma^2\}$$

such that:

$$E \subset P(d) \subset E',$$

where:

$$P(d) : \text{solve} \max \{C^T x : \quad Ax \leq b, \quad x \in \mathbb{Z}_+^n\}$$

They used ellipses based on a logarithmic barrier function for the polyhedron:

$$P = \{x : \alpha x \leq b\}$$

and:

$$\rho(x) = -\sum_{i=1}^{m} log(b - \alpha^T x),$$

where ρ was a barrier for polyhedron as $\rho(x) \to \infty$, when x approaches the boundary of P from the interior.

Then they defined an analytical center of P:

$$\overline{x} = \arg\min \rho(x).$$

As computation of this center requires a lot of extra work, they have proposed a different approach: a selection rule for the branching variable, based on the vectors corresponding to the principal axes of the Dikin ellipse associated to a central point. Having assumed that they have computed $\overline{x}$, they have defined:

$$Q = \nabla^2 \rho(\bar{x}) = A^T D(\bar{x})^{-2} A,$$

where:

$$D(\bar{x}) = diag(b - \alpha^T x).$$

Then, it can be shown that the ellipsoid constructed using Q satisfies the required properties and $\gamma = m + 1$. The ellipse E is called a Dikin ellipse for the polyhedron.

One of the problems that might occur is the fact that the shortest axis might be too small, thus searching in the orthogonal directions may results in finding more integral points. This brought them to use the following geometric fact:

Lemma 14. *Let $(\beta_1, \ldots, \beta_n)$ be the vector corresponding to the shortest semi-axis of the Dikin ellipse constructed with the point x^0 as a center. Let:*

$$\delta = min\{|\beta_j| : j = 1, \ldots, n\}.$$

Let $B(x^0, \delta)_\infty$ be the ball, in the L_∞ norm, of radius δ with center x^0.

Then if $\delta < 0.5$ and x^0 is the center of the unit hypercube, then there is no nonzero integral point in $intB(x^0, \delta)_\infty$.

Set-priority algorithm

1. As using an analytic center to define the ellipse is a very time consuming step, a different computation of the polyhedral center was proposed:

   $$\max\{t| \quad Ax + te \leq b, \quad t \geq 0\},$$

   where:

   $$e = [1, \ldots, 1]^T.$$

   Let $\bar{x}$ be the minimizer. This point is in the interior of the polyhedron, which in fact is the center of the largest sphere contained in the polyhedron.

2. Let:

   $$Q = \nabla^2 \rho(\bar{x}) = A^T D(\bar{x})^{-2} A,$$

   where:

   $$D(\bar{x}) = diag(b - \alpha^T x).$$

3. Let $(\beta_1, \ldots, \beta_n)$ be the shortest semi-axis of the following:

$$E = \{x \in \mathbb{R}^n : \quad (x - x^0)^T Q (x - x^0) \leq 1\}.$$

4. Let $\Theta = \{\beta_j : \quad \beta_j > \frac{1}{2}, \quad j = 1, \ldots, n\}$.

5. Let $P = 1, S = \Theta$.

6. Repeat until S=∅.

   As the β-coordinates correspond to the directions in which it is more probable to find integer coordinates for points contained in the polyhedron, the highest priorities should be given in the following way: let $k = \arg\max_j \{\beta_j : \quad \beta_j \in S\}$. Then $\alpha_k = p$ and $S = S - \{\beta_k\}, p = p + 1$.

   End repeat.

7. The priorities for the remaining components are set to zero.

The computational results presented by the authors has indicated 50% better performance of the B&B algorithm. However, the computation of the selection is not truly based on the analytic center of the feasible region, but on the much less time-consuming - polyhedra central point.

2.3. Dual Integer Programming

In general, any problem of form:

$$z_D = \max \{g(u) | u \in U\}$$

is called a dual problem with respect to the primal problem (2.1), when $z_D \leq z^*$, and strong dual if it becomes equal.

2.3.1. The Subadditive Dual

Wolsey introduced the following aproach to the dual problem:

$$z_D^g = \max_{g:\mathbb{R}^m \to \mathbb{R}} \{g(b) | g(Ax) \leq c^T x, x \in X'\} = \max_{g:\mathbb{R}^m \to \mathbb{R}} \{g(b) | g(d) \leq x^*(d), d \in \mathbb{R}^m\}$$

Then $z^*(d) = \min_{x \in X'(d)} c^T x$ is the value function, which expresses the optimal value of a MIP as a function of the right-hand side d and:

$$X'(d) = \{x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} | Ax = d, x \geq 0\}.$$

Functions, that are solutions to the above mentioned program, approximate the value function from below, hence the ones that agree with the value function at b are optimal [21].

Clearly only a few solutions bound the value function equally well. Naturally, the class of functions considered above might be restricted to belong to some class of functions. Johnson and Jeroslow proposed an idea of restricting the domain to the set of subaddiditve functions [21], such that it satisfies:

$$f(a+b) \leq f(a) + f(b) \text{ for all vectors } a, b \in D.$$

In that way proposed Wolsey an approach that might be rewrited in the pure integer case as the subadditive dual:

$$\begin{array}{ll} \max & f(b) \\ & f(A) \leq c^T \\ & f \in F \\ \text{where:} & F\text{- subadditive.} \end{array} \tag{2.3}$$

Blair indicated that the value function is able to be extended to a subadditive function defined on all of $\mathbb{R}^n$. Moreover Wolsey has showed that *the subadditive dual* extends many of the properties of the LP dual, such as the complementary slackness to MIPs, and what is more interesting *the subadditive dual* provides sensitivity analysis to right-hand side perturbations. However, such an approach becomes impractical with respect to large problems.

2.3.2. Branch & Bound Dual

Wolsey and Schrage proposed in [22] an alternative way of deriving a dual solution. Their approach of obtaining dual solutions from the B&B tree, does not provide any independently computed upper bound, but constitutes a useful sensitivity analysis tool.

The feasible set will be composed of the following functions:

$$f(d) = \max\{yd + y_0, \min\{f_1(d), f_2(d)\}\}, \tag{2.4}$$

where $y \geq 0$ and f_1, f_2 are either identically zero, or of the form (2.4). Strong duality might be shown as follows.

At each node t of the search tree, the linear relaxation for the (2.1) is solved:

$$\begin{array}{lll} \min & c^T x & \\ & & \\ Ax \geq b & & (u) \\ x \geq L^t & & (\alpha) \\ -x \geq -U^t & & (\beta) \\ & x \in \mathbb{Z}_+ & \\ & & \\ \text{where:} & c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \text{ and } b \in \mathbb{R}^m, & \end{array} \tag{2.5}$$

where u, α, β stands for the dual multipliers, and L^t, U^t are the lower and the upper bound defined by branching, respectively.

By weak LP duality the lower bound on the optimal value of (2.5) with perturbed right-hand side $d = b + \Delta b$ is

$$v_t(\bar{b}) = ub + \alpha L^t - \beta U^t.$$

When (2.5) is infeasible, the dual solution u, α, β of the phase I problem might still exist in which the objective function is the sum of negative constraint violations [16]. Then $v_t(\bar{b}) = -\infty$ if $u\bar{b} + \alpha - \beta > 0$, and $+\infty$ otherwise.

When t_1, t_2 are the child nodes of node t,

$$w_t(b) = \max\{v_t(\bar{b}), \min\{w_{t_1}(\bar{b}), w_{t_2}(\bar{b})\}\} \tag{2.6}$$

is a lower bound of the optimal value of (2.5) with right-hand side $\overline{b} = b + \Delta b$ and integral x.

Since $w_0(b)$ is the optimal value of (2.1), the recursively computed function w_0 tied with the root node, solves the dual problem (2.3).

2.3.3. Inference Duality

The inference dual is the problem of inferring from the constraints the best possible bound on the optimal value. A mixed integer problem has the form:

$$\begin{aligned} \min \quad & z = cx \\ & Ax \geq b \\ & 0 \leq x \leq h \\ & x_j \quad \text{integer}, \quad j = 1, \dots, k. \end{aligned} \tag{2.7}$$

The inference dual is defined:

$$\begin{aligned} \max \quad & z \\ & (Ax \geq b, 0 \leq x \leq h) \xrightarrow{D} cx \geq z \\ \text{where:} \quad & D = D_1 \times \ldots \times D_k \times \mathbb{R}^{n-k}. \end{aligned} \tag{2.8}$$

Therefore the largest z for which $f(x) \geq z$ can be inferred from the constraint set is being sought after during the dual.

When the optimal value of a minimization problem is allowed to be ∞ or $-\infty$, if the problem becomes infeasible or unbounded, and the same for a maximization problem, the optimal value of the primal (2.7) equals the optimal value of the dual (2.8), which is a consequence of the Strong Inference Duality Theorem proved by Hooker in [11]:

When an optimal value of the primal is z^*, then solving the dual (2.8) becomes simply the proof of $f(x) \geq z^*$ using the constraints as premises. The same proof exhibits that $f(x) \geq z^* - \Delta z$ for $\Delta z \geq 0$. For the MIP the proof is reconstructed from the B&B search tree, which solves the primal problem.

Logical Properties of Inequalities

Dawande & Hooker suggest to treat a mixed integer inequality of the form $bx \geq \beta$ as a proposition in the logic of discrete variables [4].

While the MIP problem is being solved there might be k integer variables: $x_1, \ldots, x_k$ and other variables $x_{k+1}, \ldots, x_n$, that are continuous, in sense that they take a fractional value.

Let us now suppose that:

$$x_j \in \{0, 1, \ldots, h_j\} \quad for \quad j = 1, \ldots, k \quad and \quad 0 \leq x_j \leq h_j \quad for \quad j = k{+}1, \ldots, n.$$

However, when the continuous variable x_j is unbounded, h_j ought to be set to infinity.

When $bx \geq \beta$ is satisfied by $(x_1, \ldots, x_n) = (v_1, \ldots, v_n)$ for some assignment to $x_{k+1}, \ldots, x_n$ such that $0 \leq v_j \leq h_j$ for $j = k + 1, \ldots, n$, then the following assignment $(x_1, \ldots, x_k) = (v_1, \ldots, v_k)$ does not violate the previously mentioned inequality $bx \geq \beta$, otherwise it does.

Hooker & Dawande's sensitivity analysis methodology is concerned with partial assignments that are imposed by the branch cuts in a search tree [4].

During the B&B procedure, one branches on a variable x_j that, in the solution of the LP relaxation, has a fractional value between two integers v and $v + 1$.

These values impose cuts of the following form:

$$x_j \leq v, \quad x_j \geq v + 1.$$

Such branch cuts from every given node make the partial assignment A:

$$x_j \in \{\underline{v}_j, \underline{v}_j + 1, \ldots, \overline{v}_j, \quad j = 1, \ldots, k\}$$

where $\underline{v}_j, \overline{v}_j$ takes integer values from $\{0, 1, \ldots, h_j\}$ for $j = 1, \ldots, k$.

With a partial assignment A, there are associated falsified clauses, the weakest one is:

$$C_A = \bigvee_{j=1}^{k} (x_j \notin \{\underline{v}_j, \ldots, \overline{v}_j\}).$$

Hooker & Dewande introduced the following necessary and sufficient condition, ensuring that an inequality implies a clause of the upper form [4]:

Lemma 15. *$bx \geq \beta$ implies C_A if and only if there exist $\bar{b}_1, \ldots, \bar{b}_n$ such that:*

$$\sum_{j=1}^{n} b_j \underline{v}_j + \bar{b}_j(\bar{v}_j - \underline{v}_j) < \beta,$$

$$\bar{b}_j \geq b_j \geq 0, j = 1, \ldots, n.$$

where $(\underline{v}_j, \bar{v}_j) = (0, h_j)$ for $j = k+1, \ldots, n$.

Solution Procedure

A solution of the dual (2.8) can be recovered in the following way. At each leaf node of the tree, let $\bar{z}$ be the best integral solution found so far for a solution that is feasible in (2.7), if a solution was not found, $\bar{z} = \infty$.

At each node, as the original constraint set is strengthened with branching cuts of the form $x_j \leq U^p$ or $x_j \geq L^p$, mentioned in the previous section, the following linear relaxation is being solved:

$$\begin{array}{lll} \min & z = cx & \\ & & \\ & Ax \geq b & (u) \\ & x \geq L^p & (\alpha) \\ & -x \geq -U^p & (\beta) \\ & 0 \leq x \leq h & \end{array} \tag{2.9}$$

Then one of the following cases is satisfied:

1. The linear relaxation (2.9) is infeasible. Then a non-negative vector (u, α, β) of dual multipliers proves infeasibility. That is:

$$u^p A + \alpha - \beta \leq 0 \quad \text{and} \quad u^p b + \alpha^p L^p - \beta^p U^p > 0.$$

Thus the following constraints are infeasible:

$$u^p Ax \geq u^p b, \quad L^p \leq x \leq U^p, \quad 0 \leq x \leq h.$$

In other words, the surrogate is inconsistent with bounds.

2. The solution of (2.9) is integral and equals to $\hat{z}_p$, where $\hat{z}_p < \overline{z}$. In that situation for any $\Delta z \geq 0$, the following set of constraints becomes infeasible:

$$-cx \geq -\hat{z} + \Delta z, \quad Ax \geq b, \quad x \geq L^p, \quad -x \geq -U^p, \quad 0 \leq x \leq h.$$

When (u^p, α^p, β^p) is a dual solution of (2.9), then the multipliers $(1, u^p, \alpha^p, \beta^p)$ prove infeasibility of the constraint set mentioned above, which means that bounds are inconsistent with the following surrogate:

$$(u^p A - c)x \geq U^p b - \hat{z} + \Delta z.$$

3. The solution of (2.9) is integral and equals $\hat{z}_p$, where $\hat{z}_p \geq \overline{z}$. In that situation for any $\Delta z \geq 0$, the following set of constraints becomes infeasible:

$$-cx \geq -\overline{z} + \Delta z, \quad Ax \geq b, \quad x \geq L^p, \quad -x \geq -U^p, \quad 0 \leq x \leq h.$$

As before, the multipliers $(1, u^p, \alpha^p, \beta^p)$ prove infeasibility, and bounds are inconsistent with the following surrogate:

$$(u^p A - c)x \geq U^p b - \overline{z} + \Delta z.$$

Therefore bounds are inconsistent with the surrogate at every leaf node, the main issue to sensitivity analysis is to keep contradictions at every leaf node, and the proof remains valid.

2.4. Inferenced-Based Sensitivity Analysis for MIP

Hooker with Dawande in [4] have introduced a new approach - search and inference, based on two parallel points of view. From the primal problem one can derive what values should be assigned to the variables, as from the dual, what may be inferred from the constraints.

The primal problem is solved by the usual B&B method, while the dual is solved with respect to inference methods for generating new constraints with the aim at inferring the best possible bound on the objective function value.

During sensitivity analysis the contribution of each constraint in the proof of optimality is examined. It might be revealed, that some of them do not have impact on the proof and might be dropped, while others are suspectible to alterations, which do not affect the proof.

The analysis presented by Dawande & Hooker allows any kind of perturbation of the problem data, i.e. within right-hand sides, constraint coefficients and the objective function values.

Hooker and Dawande have developed two ways of sensitivity analysis for a given minimization problem. The so-called dual analysis, which allows to determine how much the problem can be altered, while keeping the objective function value at some prespecified level. This method uses the inference dual.

They have proposed a primal analysis as well, achieved by solving LP problems at feasible leaf nodes of the search tree. The analysis gives an upper bound on the objective function, when the main problem is altered by a given amount.

Let z^* be the optimal value of (2.7), and suppose that (2.7) is perturbed as follows:

$$\begin{aligned} \min \quad & z = (c + \Delta c)x \\ & (A + \Delta A)x \geq b + \Delta b \\ & 0 \leq x \leq h \\ & x_j \quad \text{integer}, \quad j = 1, \ldots, k. \end{aligned} \tag{2.10}$$

The violated surrogate becomes in corresponding cases:

1. $u^p(A + \Delta A)x \geq u^p(b + \Delta b)$,

2. $(u^p(A + \Delta A) - (c + \Delta c))x \geq u^p(b + \Delta b) - \hat{z} + \Delta z$,

3. $(u^p(A + \Delta A) - (c + \Delta c))x \geq u^p(b + \Delta b) - \overline{z} + \Delta z$.

Let u^p be the vector u of dual multipliers at leaf node p. The surrogate at node p is:

$$(u^pA - u_0^pc)x \geq u^pb - z_p + \Delta z_p + \epsilon,$$

which can be written as:

$$q^px \geq u^pb - z_p + \Delta z_p + \epsilon.$$

where

$$\begin{aligned} q^p &= u^pA - u_0^pc, \\ \Delta q^p &= u^p\Delta A - u_0^p\Delta c. \end{aligned}$$

After the problem is perturbed this surrogate becomes:

$$(q^p + \Delta q^p)x \geq u^p(b + \Delta b) - z_p + \Delta z_p + \epsilon.$$

and:

$$(u_0^p, z_p - \Delta z_p) = \begin{cases} (0, \epsilon) & \text{case}(1) \\ (1, \hat{z}_p - \Delta z) & \text{case}(2) \\ (1, \bar{z}_p - \Delta z) & \text{case}(3). \end{cases}$$

The question rises: how much alteration is possible? To answer this question Hooker & Dawande introduced a helpful observation, based on the *Lemma 15*: the bounds are inconsistent with inequality $dx \leq \delta$ if and only if there exists a vector $\overline{d} > 0$, such that:

$$dL + \overline{d}(U - L) < \delta, \quad \overline{d} \geq d \quad \text{and} \quad \overline{d} \geq 0.$$

With respect to the n variables this becomes:

$$\sum_{j=1}^{n} d_jL_j + \overline{d}_j(U_j - L_j) < \delta, \quad \overline{d}_j \geq d_j \quad \text{and} \quad \overline{d}_j > 0, \quad j = 1, \ldots, n. \quad (2.11)$$

Having used this observation, the bound $z \geq z^* - \Delta z$ remains valid for (2.10) if the perturbations satisfy for some vector $\overline{q}^p > 0$ at every leaf node p, the following:

$$
\begin{aligned}
&\textstyle\sum_{j=1}^{n}(q_j^p + \Delta q_j^p)L_j^p + \bar{q}_j^p(U_j^p - L_j^p) \leq u^p(b + \Delta b) - z_p + \Delta z_p, \\
&\bar{q}_j^p \geq q_j^p + \Delta q_j^p, \\
&\bar{q}_j^p \geq 0,
\end{aligned} \tag{2.12}
$$

where:

$$(L_j^p, U_j^p) = (0, h_j) \quad \text{for} \quad j = k+1, \ldots, n.$$

Hooker & Dawande have introduced this as the following theorem:

Theorem 16. *[4] If (2.7) is perturbed as in (2.10), then the optimal value of (2.7) decreases at most Δz, when the perturbation satisfies the linear system consisting of the inequalities (2.11) for each leaf node p of the search tree.*

Thus assuming that there exist $s_1^p, \ldots, s_n^p$, we may take $\bar{q}_j^p = s_j^p + q_j^p$. The sensitivity analysis might be introduced for each constraint in the system $Ax \geq b$. If $s_1^p, \ldots, s_n^p$ satisfy the following set of inequalities, for each leaf node p with $u_i^p > 0$ the constraint $z \geq z^* - \Delta z$ remains valid:

$$
\begin{aligned}
&u_i^p \textstyle\sum_{j=1}^{n} A_{ij}L_j^p + \sum_{j=1}^{n} s_j^p(U_j^p - L_j^p) - u_i^p \Delta b_i \leq r_p, \\
&s_j^p \geq u_i^p \Delta A_{ij}, \;\; s_j^p \geq -q_j^p, \;\; j = 1, \ldots, n,
\end{aligned} \tag{2.13}
$$

where

$$r_p = -\sum_{j=1}^{n} q_j^p U_j^p + u^p b - z_p + \Delta z_p.$$

A perturbation Δc of the objective function can be similarly analyzed. The bound remains valid if, for each leaf node p with $u_0^p = 1$, there are $s_1^p, \ldots, s_n^p$ that satisfy the following:

$$
\begin{aligned}
&\textstyle\sum_{j=1}^{n} \Delta c_j L_j^p - s_j^p(U_j^p - L_j^p) \geq -r_p, \\
&s_j^p \geq -\Delta c_j, \;\; s_j^p \geq -q_j^p, \;\; j = 1, \ldots, n.
\end{aligned} \tag{2.14}
$$

To summarize:

If z^* is the optimal value of (2.7), then the bound $z \geq z^* - \Delta z$ remains valid when constraint $A_i x \geq b_i$ is perturbed to $(A_i + \Delta A_i)x \geq b_i + \Delta b_i$, provided that there are $s_1^p, \ldots, s_n^p$ that satisfy the linear system consisting of the inequalities (2.13) for each leaf node p with $u_i^p > 0$.

If z^* is the optimal value of (2.7), then the bound $z \geq z^* - \Delta z$ remains valid when the objective function is perturbed to $(c + \Delta c)x$, provided that there are $s_1^p, \ldots, s_n^p$ that satisfy the linear system consisting of the inequalities (2.14) for each leaf node p with $u_0^p = 1$.

In other words the basic idea of the proposed approach is to utilize the information obtained from the sensitivity analysis of the deterministic solution to determine the importance of different parameters and constraints and the range of parameters where the optimal solution remains unchanged.

More speciffically, there are two parts in the proposed analysis. In the first part, important information about the effect of different parameters is extracted following the sensitivity analysis step, whereas in the second part alternative solutions are determined and evaluated for different uncertainty ranges.

First, the deterministic problem is solved at the nominal values using a branch and bound solution approach, and the dual multipliers u_p are collected at each leaf node p.

Then the inference-based sensitivity analysis as described in previous section is performed. Note that, only the dual information of the nodes that correspond to nonzero dual variables is required.

Primal analysis

A simple primal analysis obtains an upper bound on the optimal value that results from a given perturbation of the problem data. Let F be the set of nodes of the search tree at which feasible solutions were found for the original problem. For each node $p \in F$ consider the linear programming problem:

$$\begin{aligned}
\min \quad & z = (c + \Delta c)x \\
& (A + \Delta A)x \geq b + \Delta b \\
& x \geq L^p \\
& -x \geq -U^p \\
& 0 \leq x \leq h \\
& x_j = \overline{x}_j, \ j = 1, \ldots, k,
\end{aligned} \tag{2.15}$$

where $\overline{x}_1, \ldots, \overline{x}_k$ are the integral solution values of $x_1, \ldots, x_k$ at node p. Then if z'_p is the optimal value of (2.15) at node p, $\min_{p \in F} z'_p$ is an upper bound on the optimal value of the perturbed problem (2.10).

Implementation

The procedure described is straightforward to implement within a B&B framework. Deriving a surrogating inequality which is violated by the partial assignment at a leaf node requires the following:

1. the bounds on the variable that are restricted by branch cuts at the leaf node,

2. the dual solution vector corresponding to the original problem constraints $Ax \geq b$.

If the inequalities, which are to be investigated during sensitivity analysis, are not known beforehand, then the part of the dual solution corresponding to the original constraints $Ax \geq b$ must be stored for every leaf node. However, if the constraints of interest are known in advance, only the leaf nodes at which the corresponding dual variables are nonzero are relevant.

Regardless of whether the constraints of interest are specified in advance, the tolerance Δz on the optimal value need not to be fixed until the linear system used to compute sensitivity ranges is already set up. The user can interactively try different values of Δz and obtain the corresponding ranges.

Jia and Ierapetritou Approach

Zhenya Jia and Marianthi G. Ierapetritou have considered the Short-Term Scheduling problem under Uncertainty Using MIP Sensitivity Analysis [13].

Their approach is based on the Hooker & Dawande approach. They basically use the Hooker & Dawande Duality Analysis to obtain possible problem perturbations, ranges of uncertain parameters for some alterations in the objective function. Next, they proceed to continue the B&B procedure, but only on the nodes with the objective value within some predicted limits.

The alternative schedules are assessed with respect to the robustness metric and the average and the nominal schedule in terms of the objective function

[13]. This, in consequence, makes the problem much larger, and much more difficult to be dealt with.

The New Approach

Having implemented the primal analysis, we can see that not only we are able to obtain an upper bound, but what is more - all possible different scenarios with the solution that is equal to an upper bound.

If now we would consider the original problem, and instead of the primal analysis, just solve our MIP without any perturbations at each feasible node, we will be provided with the set of all possible efficient solutions. That means that a decision maker now has the possibility to select which optimal solution he would like to choose. It is a big change of quality as usually software only provides the decision maker with the first efficient solution found. Thus it gives us an useful sensitivity analysis tool.

Chapter 3

Multicriteria Optimization

Multiple objective linear and integer programming problems play a central role in the field of multicriteria optimization. Besides their relevance for practical applications, they are theoretically interesting due to their connection with the classical theory of (single objective) linear and integer programming.

Wherever two or more conflicting objectives occur, an application of multi-objective optimization problems might be found. Decision makers may want to miximize the proft, while minimizing the cost at the same time. This situation stands as the easiest example of multicriteria optimization technique.

In this chapter some sensitivity analysis techniques are introduced, that have been introduced in previous years by researchers. However, we will not use this theory to conduct the sensitivity analysis within the call center framework.

3.1. Multiple Objective Linear Programming

MOLP constitutes a fast growing branch of mathematical programming, mainly it is resulted by offering plenty points of view to the solution of many complex problems. The vector maximization algorithm contributes to searching MOLP efficient solutions, from which a decision maker may select those, he prefers.

3.1.1. Classical MOLP problem

Let us define the classical form of the MOLP problem:

$$
\begin{aligned}
\text{Vmax} \quad & Cx \\
& Ax \leq b \\
& x \in \mathbb{R}^n_+ \\
\text{where:} \quad & b \in \mathbb{R}^m, C \in \mathbb{R}^{k \times n}, A \in \mathbb{R}^{m \times n}
\end{aligned}
\tag{3.1}
$$

Furthermore let us call $\mathbb{R}^n$ a decision space, while $\mathbb{R}^k$ is the criterion space.

3.1.2. Graphical method

There exist two equivalent graphical methods for computing the efficient solution sets: in a decision space, and in a criterion space. Let us first introduce the following theorem:

Theorem 17 (Characterization of the feasible region for MOLP)**.** *If the feasible region X is bounded, then it is a convex hull of all its own vertexes:*

$$X = \{x \in \mathbb{R}^n : x = \sum_{i=1}^{s}(\lambda_i A_i), \quad \sum_{i=1}^{s}(\lambda_i A_i) = 1, \quad \lambda \geq 0\},$$

where:

$\{A_1, \ldots, A_s\}$ - *the vertex set of X.*

Graphical method in a decision space

The method described below will be introduced in the situation when we deal with only two decision variables, but there is an arbitrary number of partial objective functions.

Theorem 18 (Characterization of the efficient solutions set X_s for MOLP)**.** *Let S be a polyhedra defined in the following way:*

$$S := \{x \in \mathbb{R}^n : Cx \geq 0\},$$

then:

— *if S is a point or a line, then $X = X_s$,*

— *if S a half-plane, then:*

$$X_s = \{x \in \mathbb{R}^n : X \cap int\{x + S\} = \emptyset\},$$

— *If S is a polyhedron different that the one descibed in the previous part, then:*

$$X_s = \{x \in \mathbb{R}^n : X \cap \{x + S\} = \{x\}\}.$$

Graphical method in a criterion space

The following method is based on a drawing of the set of all possible objective function vectors, thus it is applied only if there are only two objective functions, and only two decision variables.

Theorem 19 (Set of all possible objective function values)**.** *Let X be the feasible region:*

$$X = \{x \in \mathbb{R}^n : x = \sum_{i=1}^{s}(\lambda_i A_i), \quad \sum_{i=1}^{s}(\lambda_i A_i) = 1, \quad \lambda \geq 0\},$$

where:

$$\{A_1, \ldots, A_s\} \text{ - the vertexes set of } X,$$

then the set of all possible objective function values is defined by:

$$C(X) = \{y \in \mathbb{R}^k : x = \sum_{i=1}^{s}(\lambda_i C(A_i)), \quad \sum_{i=1}^{s}(\lambda_i A_i) = 1, \quad \lambda \geq 0\}.$$

Theorem 20 (Characterization of undominated price sets for MOLP)**.** *Let* $\mathbb{R}^k_+$ *be a polyhedron of non-negative vectors, that means:* $\mathbb{R}^k_+ = \{z \in \mathbb{R}^k : z \geq 0\}$ *then:*

$$y \in Y_N \Longleftrightarrow Y \cap \{y + \mathbb{R}^k_+\} = \{y\}.$$

3.1.3. Mulitiple objective simplex tableu

Let us introduce some basic notation:

$\mathcal{B} = \{j_1, \ldots, j_m\}$	–	the set of basis indexes
$\mathcal{N} = \{j_m, \ldots, j_{(m+n)}\}$	–	the set of non-basis indexes
$A_B = [a_{j_1}, \ldots, a_{j_m}]$	–	submatrix of A, associated with $\mathcal{B}$
A_N	–	submatrix of A without basis columns
x_B	–	vector of basis variables
x_N	–	vector of non-basis variables
C_B	–	basis columns of a cost matrix C
C_N	–	non-basis columns of a cost matrix C
$\overline{c}$	–	matrix of optimality indicators

Definition 8. *We call the vector* $[x_B, x_N]$*the basis feasible solution with respect to* $\mathcal{B}$ *if:*

$$x_B = A_B^{-1} b \geq 0,$$

$$x_N = 0.$$

And we call a such base the feasible base.

This leads us to a simplex tableau for fixed base $\mathcal{B}$:

		C	
		x	
x_B	C_B^T	$A_B^{-1}A$	$A_B^{-1}b$
		$\overline{C}$	$C_B A_B^{-1} b$

3.1.4. Pivoting rules

Neighbour vertex generating

If x^* is a BFS (basis feasible solution) with respect to $\mathcal{B}$, then for all $s \notin \mathcal{B}$ such that in the s-column of the $A_B^{-1}A$ matrix there exists at least one positive element, then there exists a neighbour BFS to x^*, such that x_s is a basis variable. Pivoting from x^* to x^{**} is described by:

$$\overline{x}_k^{**} = \begin{cases} x_k^* - \alpha(s) \cdot (A_B^{-1}A)_{is} & k \in \mathcal{B} \\ \alpha(s) & k = s \\ 0 & k \notin \mathcal{B} \cup \{s\}, \end{cases}$$

where:

$$\alpha(s) = \min \left\{ \frac{A_B^{-1} b_i}{(A_B^{-1}A)_{is}} : \quad (A_B^{-1})_{is} > 0, \quad i = 1, \ldots, m \right\}$$

Unbounded edge generating

If x^* is a BFS with respect to $\mathcal{B}$, and $s \notin \mathcal{B}$ is such that in the s-column of the $A_B^{-1}A$ matrix all elements are nonpositive, then there exist unbounded edge of the set X, with the beginning in x^*, and the direction described by:

$$\overline{x}_k = \begin{cases} -(A_B^{-1}A)_{is} & k \in \mathcal{B} \\ 1 & k = s \\ 0 & k \notin \mathcal{B} \cup \{s\}. \end{cases}$$

3.1.5. Basis feasible solution

It commonly occurs that for finding the first BFS, slack variables should be used, which requires solving the following subproblem:

$$\begin{aligned} \min \quad & \mathbf{1}^T v \\ & Ax + Iv = b \\ where: \quad & 0 \leq x \in \mathbb{R}^n, \quad 0 \leq v \in \mathbb{R}^m. \end{aligned} \tag{3.2}$$

Theorem 21. *If an optimal value of an objective function in the above mentioned slack subproblem is equal to 0, and the basis optimal solution is a vector* $[x^d, v^d]$*, then* x^d *is the BFS in the MOLP.*

However, if an optimal value of an objective function is larger than 0, then the set X of efficient solutions in MOLP is empty.

3.1.6. Finding Initial Efficient Extreme Point

Benson's method is known for generating the first basis efficient solution for a MOLP problem [2]. The method consists of two steps. In the first one, the following corresponding to MOLP problem - LP program is solved:

$$\begin{aligned} \min \quad & \{-z^T C x^d + w^T b - \alpha(\mathbf{1}^T b)\} \\ & z^T C - w^T A + \alpha(\mathbf{1}^T A) + Iv = -\mathbf{1}^T C, \\ where: \quad & 0 \leq v \in \mathbb{R}^n, \quad 0 \leq z \in \mathbb{R}^k, \quad 0 \leq w \in \mathbb{R}^m, \quad 0 \leq \alpha \in \mathbb{R}. \end{aligned} \tag{3.3}$$

Then if the program (3.3) does not have an optimal solution, there does not exist an efficient extreme point for a corresponding MOLP problem. However, if an optimal solution of the program (3.3) exists, it takes the form of the following vector $[z^*, u^*, v^*, \alpha]$.

When an optimal solution of the LP program (3.3) is equal to 0, and $x^d = 0$, then if and only if x^d is a basis solution, x^* is a basis efficient solution for the corresponding MOLP problem.

When an optimal solution of the LP program (3.3) is greater than zero, then we proceed to the second step of the algorithm.

The efficient vertex is derived by the solution of the following LP problem:

$$\begin{aligned} \max \quad & (z^* + \mathbf{1})^T C x \\ & Ax = b \\ where: \quad & x \geq 0. \end{aligned}$$

3.1.7. ADBASE algorithm

The ADBASE method consists of three phases:

1. Either compute the first basis feasible solution, or prove that considered problem is unsolvable.

2. Either compute the first basis efficient solution, or prove that considered problem is unsolvable.

3. Computation of all efficient solutions and efficient edges.

3.2. Sensitivity Analysis in MOLP

The several types of sensitivity analysis might be proposed:

— on objective function coefficients alterations

— on a removal of an objective function

— on an addition of an objective function

— on right-hand side changes

— or on alterations of the matrix A.

We will focus on the first branch of SA.

3.2.1. Sensitivity Analysis of efficiency

Preliminaries

Definition 9. *We call x^* an efficient solution of MOLP if there does not exist an x' such that:*

$$Cx^* \leq Cx' \wedge Cx^* \neq Cx'.$$

Definition 10. *If there exist $i \in \mathcal{B}$, such that $(x_B)_i = 0$, then we call the solution x_B degenerated, and $(x_B)_i$ a degenerated basis variable. Let us denote the number of degenerated variables by d, and analogously by $[A_B^{-1}A_N]_D$ the matrix associated with the degenerated basis variables.*

We will denote the set of all efficient solutions of our problem by X_S.

Theorem 22 (Testing efficiency of extreme points)**.** *[Steuer] Consider the following single objective linear program associated with an extreme solution x^*:*

$$\begin{aligned} \max \quad & \mathbf{1}^T v \\ & -\overline{C}_N y + Iv = 0 \\ & [A_B^{-1}A_N]_D y + Is = 0 \\ & y \geq 0, v \geq 0, s \geq 0. \end{aligned} \tag{3.4}$$

The solution x^* is efficient if and only if the above mentioned problem has an objective function value of zero.

We will consider the following problem:

$$V \max \{D_t^{ij} x : \quad x \in X\},$$

where D_t^{ij} is a matrix obtained from C, by changing only one parameter c_{ij} into parameter t:

$$[D_t^{ij}]_{kl} = \begin{cases} c_{kl} & (k,l) \neq (i,j) \\ t & \textit{otherwise.} \end{cases}$$

Whenever it does not cause misunderstanding, we omit the indices, and define $D := D_t^{ij}$, analogously. We will denote the reduced cost matrix of our new problem by $\overline{D}$.

Definition 11. *We denote the set of all values of parameter t for which a given solution is efficient of our problem in the following way:*

$$T_{x^*} = \{t \in \mathbb{R} : \quad x^* \in X_S\}.$$

Theorem 23. *The set T_{x^*} is convex.*

[23]. Let $t_0, t_1 \in T_{x^*}, \lambda \in (0,1)$. We denote $t_\lambda = \lambda t_1 + (1-\lambda)t_0$. It is easily seen that:

$$D_{t_\lambda} = \lambda D_{t_1} + (1-\lambda) D_{t_0}.$$

We will use the following well-known characterization of the efficient solutions:

$$t \in T_{x^*} \quad \Longleftrightarrow \quad \forall x \in X, \quad D_t x \geq D_t x^* \quad \Longrightarrow \quad D_t x = D_t x^*.$$

Let x' fulfill $D_{t_\lambda} x' \geq D_{t_\lambda} x^*$, then:

$$\lambda D_{t_1} x' + (1-\lambda) D_{t_0} x' \quad \geq \quad \lambda D_{t_1} x^* + (1-\lambda) D_{t_0} x^*.$$

Hence we get:

$$D_{t_1} x' \geq D_{t_1} x^* \quad \textit{or} \quad D_{t_0} x' \geq D_{t_0} x^*.$$

Let us assume the latter one. Since $t_0 \in T_{x^*}$ we have:

$$D_{t_0} x' = D_{t_0} x^*,$$

analogously the same holds for t_1, what provides:

$$D_{t_\lambda} x' = D_{t_\lambda} x^*.$$

We have shown that if an arbitrary $x' \in X$ fulfills $D_{t_\lambda} x' \geq D_{t_\lambda} x^*$, then $D_{t_\lambda} x' = D_{t_\lambda} x^*$. It means that $t_\lambda \in T_{x^*}$, which proves that T_{x^*} is convex. □

Graphical method for testing efficiency

Theorem 24 (Steuer)**.**

$$x^* \in X_S \quad \Longleftrightarrow \quad D_{x^*} \cap X = \{x^*\},$$

where:

$$D_{x^*} = \{x^*\} + \{y \in \mathbb{R}^n : \quad Cy \geq 0 \wedge Cy \neq 0\} \cup \{0 \in \mathbb{R}^n\}.$$

Numerical method for testing efficiency

We now build a numerical method in order to determine the set T_{x^*}, which is resulted by the Steuer theorem, introduced at the beginning of this chapter:

$$\begin{array}{ll} \max & \mathbf{1}^T v \\ & \\ & -\overline{C}_N y + Iv = 0 \\ & [A_B^{-1} A_N]_D y + Is = 0 \\ & y \geq 0, v \geq 0, s \geq 0. \end{array} \tag{3.5}$$

3.2.2. Sensitivity Analysis of weak-efficiency

The sensitivity analysis is presented here in the following way: is a given weak efficient solution still weak efficient after one objective function coefficient change? Moreover we limit our consideration to the extreme feasible solutions. Our aim is to compute the set of the parameters (corresponding to only one coefficient) for which a given feasible solution is weak efficient. This section shows that this set is a closed interval.

Preliminaries

Definition 12. *We call x^* a weak efficient solution of the MOLP if there does not exist an x' such that:*

$$Cx^* < Cx'.$$

We will denote the set of all efficient solutions of our problem by X_W.

Definition 13. *We denote the set of all values of parameter t for which a given solution is weak efficient of our problem in the following way:*

$$T^W_{x^*} = \{t \in \mathbb{R} : \quad x^* \in X_W\}.$$

Graphical method for testing weak efficiency

Theorem 25 (Steuer)**.**

$$x^* \in X_W \quad \Longleftrightarrow \quad D^W_{x^*} \cap X = \{x^*\},$$

where:

$$D^W_{x^*} = \{x^*\} + \{y \in \mathbb{R}^n : \quad Cy > 0\} \cup \{0 \in \mathbb{R}^n\}.$$

Numerical method for testing weak efficiency

Theorem 26 (Steuer)**.** *The extreme solution x^* is a weak solution if and only if the following problem:*

$$\begin{aligned} \max \quad & r \\ & -\overline{D}_N y + \mathbf{1} r \leq 0 \\ & [A_B^{-1} A_N]_d y + Is = 0 \\ & 0 \leq y \in \mathbb{R}^{n-m}, 0 \leq s \in \mathbb{R}^d, 0 \leq r \in \mathbb{R}, \end{aligned} \tag{3.6}$$

has a bounded objective function value of zero.

Properties of $T^W_{x^*}$ set.

Theorem 27. *The set $T^W_{x^*}$ is convex.*

[24]. Let $t_0, t_1 \in T^W_{x^*}, \lambda \in (0,1)$. We denote $t_\lambda = \lambda t_1 + (1-\lambda)t_0$. Moreover let us fix $x^* \in X$ and let D^i_t denote the i-th row of matrix D_t.

Suppose that for some $\lambda \in (0,1)$ we have: $t_\lambda \notin T^W_{x^*}$, then we may find such $x' \in X$ that:

$$D_{t_\lambda} x^* < D_{t_\lambda} x',$$

and for all $k \neq i$ one has:

$$D^k_{t_0} x^* = D^k_{t_0} x^* = D^k_{t_\lambda} x^* < D^k_{t_\lambda} x' = D^k_{t_1} x' = D^k_{t_0} x'.$$

This leads to two cases:

1. $D^i_{t_0}x' \leq D^i_{t_0}x^*$.

   Then it holds that:

   $$D^i_{t_1}x' > D^i_{t_1}x^*,$$

   as a result of comparison of two linear functions: $f_1 = D^i_t x^*$ and $f_2(t) = D^i_t x'$. This means that $t_1 \notin T^W_{x^*}$, thus we have obtained a contradiction with our basic assumptions.

2. $D^i_{t_1}x' \leq D^i_{t_1}x^*$.

   Then it holds that:

   $$D^i_{t_0}x' > D^i_{t_0}x^*,$$

   as a result of comparison of two linear functions: $f_1 = D^i_t x^*$ and $f_2(t) = D^i_t x'$. This means that $t_0 \notin T^W_{x^*}$, thus we have obtained a contradiction with our basic assumptions.

We have shown that for all $\lambda \in (0, 1)$ we have $t_\lambda \in T^W_{x^*}$, which means that the set $T^W_{x^*}$ is convex indeed.

□

It is well known that on the real line convex sets are intervals, hence we obtain the following corollary:

Corollary 1. *The set $T^W_{x^*}$ is a closed interval.*

The following corollary shows some connections between the sensitivity analysis of weak efficiency and the sensitivity analysis of efficiency.

Corollary 2. *It holds for all $x^* \in X$, that:*

$$clT_{x^*} \subset T^W_{x^*}.$$

Proof. From the definitions of $T_{x^*}, T^W_{x^*}$ sets we have:

$$T_{x^*} \subset T^W_{x^*}.$$

Hence:

$$clT_{x^*} \subset clT^W_{x^*}.$$

Moreover we know that $clT^W_{x^*} = T^W_{x^*}$, thus:

$$clT_{x^*} \subset clT^W_{x^*} = T^W_{x^*}.$$

□

The question is if there an equality might hold:

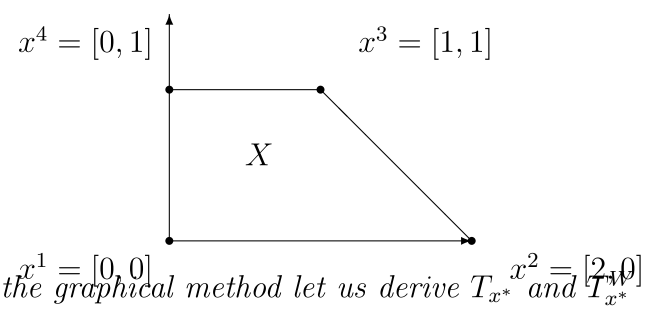
$$clT_{x^*} = T^W_{x^*} \quad ?$$

The following example shows this to be false.

Example 1. *Consider a problem:*

$$\begin{array}{ll} V\max & [x_1, x_2] \\ & x_1 + x_2 \leq 2, \\ & x_2 \leq 1. \end{array}$$

The feasible region $X = \{(x_1, x_2) : \quad x_1 + x_2 \leq 2, \quad x_2 \leq 1\}$ *is a polyhedron with the extreme points:*



Using the graphical method let us derive T_{x^*} *and* $T^W_{x^*}$ *sets for these points. We do that in the following way. First, we analyze the sensitivity for extreme points and* $c_{11} = t$, *then we obtain* T_{x^*} *and* $T^W_{x^*}$ *sets corresponding to each extreme point:*

$$T_{[0,0]} = \emptyset \qquad T^W_{[0,0]} = (-\infty, 0]$$

$$T_{[0,1]} = (-\infty, 0] \quad T^W_{[0,1]} = \mathbb{R}$$

$$T_{[1,1]} = [0, +\infty) \quad T^W_{[1,1]} = \mathbb{R}$$

$$T_{[2,0]} = (0, +\infty) \quad T^W_{[2,0]} = [0, +\infty).$$

The results above already show differences between the following sets: $clT^W_{x^*}$ and $T^W_{x^*}$. Thus the example contradicts that these sets are equal in general. However, it is still interesting to study which conditions guarantee such equality. As we see in our example only $clT^W_{[2,0]} = T_{[2,0]}$.

Summary

We studied the sensitivity analysis of weak efficiency in the MOLP for extreme feasible solutions. We proved that the set of the parameters for the considered problem is closed and convex. To compute the set of the parameters for which a given extreme solution is weak efficient we proposed the method based on the simplex algorithm. We showed in the illustrative examples the differences and similarities between the sensitivity analysis of efficiency and the sensitivity analysis of weak efficiency.

3.3. Multiple Objective Integer Programming

Having used integer variables in the MOLP problems results in a more realistic modelling approach. Some issues as the minimum capacity size of the new units, economies of scale in the investment cost are not possible to be modeled without integer or binary variables.

Now we are going to introduce the multicriteria version of Branch & Bound(MCBB) algorithm, developed by Mavrotas and Diakoulaki, which not only supplies the set of all efficient solutions, but at the same time, it indicates corresponding efficient combinations in MOIP and the same in MOMIP.

The entire chapter is based on the work of Mavrotas and Diakoulaki - the inventors and developers of the Multicriteria Branch & Bound method: [17],[18],[19].

3.3.1. Multicriteria Branch & Bound

In 1998, Mavrotas and Diakoulaki have introduced a *Branch-and-Bound* method for multicriteria optimization [18], where the characteristics of the multiple objective problem becomes vectors, instead of scalars, as it is in the single objective case. During the procedure, at each node the current best value is being calculated, and the vector of 'the best case' point is formed.

The set of efficient points is generated at final nodes, while it is stored in an *incumbent list* D_{ex}, updated each time a final node is visited. The final list of the points, stored in the list, becomes the set of the efficient solutions for the multiple objective problem.

Due to the vector comparisons at each node, the fathoming condition becomes much more difficult to be met. The generating of an efficient solutions' set at the final nodes remains the most time-consuming part: based on the MOLP Simplex Method, the efficient points there are derived using the Evans-Steuer criterion for the identification of the efficient movements in the Simplex algorithm.

There are several important differences between the usual B&B and MCBB algorithms:

1. First of all, an *ideal vector* with the individual optimal values for each objective function is computed at each node. However those values usually might not be reached by a feasible solution.

2. Moreover, all possible efficient points are computed at each leaf node. Mavrotas and Dioukalaki have introduced, the so-called *partially efficient points* (PEP), stored in an incumbent list D_{ex}, candidates for being efficient points of the Mixed 0-1 MOLP.

   At each leaf node, D_{ex}, is updated: at first, it should be checked if any point stored in D_{ex} dominates the new ones. Then we procede to examine whether any of the stored points are dominated by the new ones. The dominated ones are being removed out of D_{ex}.

3. If an ideal vector is dominated by a PEP, then a node is fathomed. The fathoming condition at each k node might be defined in the following way:

$$\exists i \in D_{ex} \quad \text{such that} \quad \max f_j(k) \leq effp_j(i) + \epsilon_j \quad \forall j = 1, \ldots, p$$

$f_j(k)$	–	j-th element of the ideal vector of node k
$effp_j(i)$	–	the value of the j-th objective of the i-th efficient point in D_{ex}
p	–	the number of objective functions
ϵ_j	–	small tolerance

with at least one strict inequality.

4. As there are more than one efficient solutions in the MOIP, thus the branching variable may take a lot of values, which results in the fact that the decision space is unable to be divided in two separate sub-regions for the multi-criteria case.

3.3.2. Transitions between nodes

In their latest paper, Mavrotas and Diakoulaki have proposed an implementation of the Revised Simplex with Bounded Variables (RSBV) method for MCBB algorithm, thanks to which it becomes easier to deal with binary variables with upper bound, determined by 1. The transitions are as follows:

— By *ind* denote the number of a node, while by *ilv* the number of a level of the B&B tree
— Let us suppose that the *ind* node is situated on the *ilv* level
— When the *idealvector* is derived at node *ind*, then the status of the simplex tableau ties with the optimal solution of L - the last objective function optimized.
— The indices of the non-basis columns which are allowed to take part in the simplex operations are contained by the list *cclist*.
— The LP optimizations necessary to derive the ideal vector, are performed consecutively with usage of a warm start rule.

3.3.3. Efficient points

The generation of each efficient point takes place at each leaf node of the MCBB tree. An ad hoc generation approach (EFFTREE) is used, to derive the set of efficient points, for each leaf node of MOLP sub-problems.

The EFFTREE method uses the RSBV method, with Evans-Steuer criterion [25], and the second Zeleny's criterion [31], for testing the non-basis variables

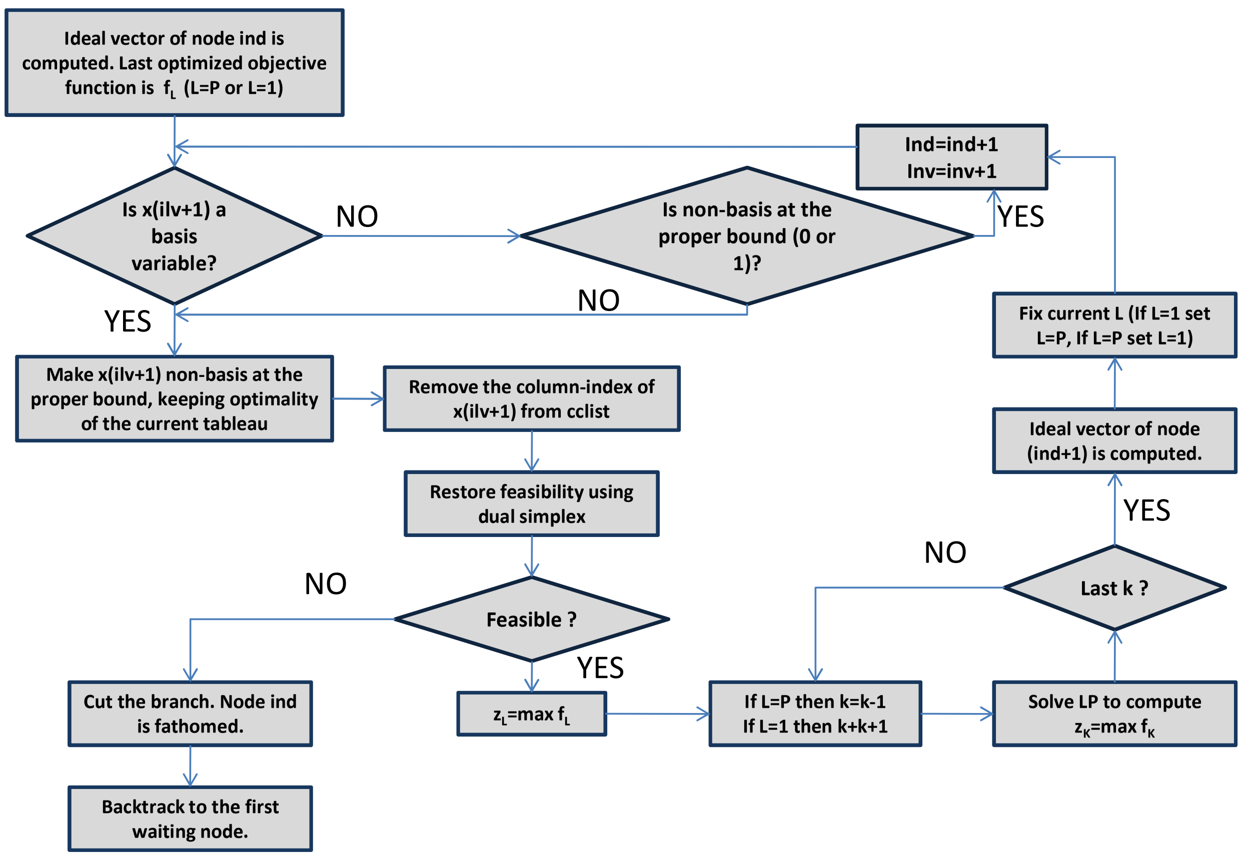


Fig 1. The figure shows the algorithmic steps for the transition from node *ind* to *ind + 1* in the next level *ilv + 1*.

efficiency.

The combination of both criteria let the number of the subroutine calls to the Evans Steuer subproblem be reduced. The EFFTREE algorithm is developed on the basis of the ADBASE method developed by Steuer and introduced as well in the MOLP section of this chapter. However, it is significantly altered to fit the MCBB algorithm in order to take account of the binary variables. Since the binary character of these variables are obliged to remain unchanged throughout the efficient extreme points' generation process and thus the corresponding columns are prevented from entering the basis using the information existing in *cclist*.

The MCBB algorithm provides the generation of both supported and unsupported efficient points, giving the complete efficient set of the Mixed 0-1 MOLP problem.

3.3.4. PEP detection

After the termination of MCBB in D_{ex} only the points are left, that are not dominated by any other one in D_{ex}.

Still there might occur situations when partially efficient points might be dominated by a linear combination of other partially efficient extreme points in D_{ex}, while it is not dominated by any other PEP of efficient combinations. The linear combination corresponds to a non-extreme efficient point of the efficient combination.

The points, that occur to be dominated by some non-extreme efficient points, are called *pseudo-efficient points.*

The test developed by Mavrotas and Diakoulaki for detecting the pseudo-efficient points in D_{ex} is based on the fact that, in the final nodes the decision space of the corresponding MOLP subproblems has no discontinuities, since all the binary variables have fixed values [19].

In consequence, at a leaf node the non-extreme efficient points might be obtained from the appropriate linear combinations of the efficient extreme points of this particular node.

The following procedure is performed. For all the efficient combinations in D_{ex} it is checked if the point $P \in D_{ex}$ is pseudo efficient, in other words, if there exists any linear combination $L(EFFC(x))$, which dominates point P. If so, the point P is eliminated from D_{ex} and the procedure is repeated for every other point from D_{ex}.

Examinination whether the point $y = (y_1, \ldots, y_p) \in D_{ex}$ is dominated by a linear combination of the efficient extreme points from an efficient combination X, let us assume that the particular efficient combination X has n efficient extreme points $x_1, \ldots, x_n$, where $x_n = (x_{1n}, x_{2n}, \ldots, x_{pn})$ is the vector formed by the values of the objective functions.

Mavrotas and Diakoulaki has introduced the following LP problem with λ_k and ϵ_k as decision variables:

$$\max \quad \epsilon_1 + \epsilon_2 + \cdots + \epsilon_p$$

$$\begin{aligned} &\lambda_1 x_{11} + \lambda_2 x_{12} + \cdots + \lambda_n x_{1n} - \epsilon_1 \geq y_1 \\ &\cdots \\ &\lambda_1 x_{p1} + \lambda_2 x_{p2} + \cdots + \lambda_n x_{pn} - \epsilon_p \geq y_p \\ &\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1 \end{aligned}$$

When there does not exist any feasible solution, then there is no linear combination $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ of the points $x_1, x_2, \ldots, x_n$. In other words, there does not exist any non-extreme efficient point in X that could dominate y.

3.3.5. Implemention of an algorithm to a pure MOIP

All MOIP problems might be converted to an 0-1 MOIP, through the transformation, if only y is an integer with an upper bound UB, then y can be expressed as:

$$y = \delta_0 + 2\delta_1 + 4\delta_2 + 8\delta_3 + \cdots + 2^k\delta_k$$

where δ_i are 0-1 variables and $2^k \leq UB \leq 2^{k+1}$. Certainly, this approach is a strong burden by an increase in the number of variables in the model.

3.3.6. Determination of alternative optima in the single objective case

The MCBB algorithm provides the decision maker with all alternative solutions with respect to the single objective case for the MIP.

Since the vast majority of the solvers only provides the first found integer solution, the alternative solutions might be not derived.

Using the MCBB algorithm, the ideal vector calculation might be reduced to a single LP case, and furthermore at the leaf nodes instead of the generation of the all efficient points, is reduced to the derivation of the LP optimal value.

The list D_{ex} of incumbent solutions, anlogously to the MOLP case will store all "current" optimal solutions, thus afterwards it contains all alternative ones.

The same methodology might be applied to the following sensitivity analysis. Let us assume that after obtaining an optimal solution z, we want to find all solutions with some tolerance Δz from z. A node is then fathomed, when:

$$z_p \leq \overline{z} + \Delta z,$$

where z_p is the optimal solution of the LP subproblem at node p, and $\overline{z}$ stands for the current incumbent solution.

If Δz is a small negative value, nodes for which it holds that:

$$z_p \leq \overline{z}$$

will not be fathomed.

Updating D_{ex} follows analogously. All solutions $z_p \leq \overline{z}$ are stored for the negative Δz. Within that framework after the MCBB termination a decision maker is provided with the list of all possible alternative solutions within some small tolerance in MIP problems.

Chapter 4

Call center: single-skill environment model

The workload of the large and rapidly growing branch of industry, the call center or even a contact center is incredibly hard to predict.

Workforce management

The workforce management deals with few phases of the labour allocation process:

1. workload prediction - due to diversity of events that need to be taken into account, a call volume estimation on the basis of historical data is extremely difficult, but nevertheless results in the forecast of the arrival rate for each interval of a time horizon.

2. staffing - minimum number of agents calculations needed to reach the service level for each interval, using the Erlang formula.

3. shift schedulling - as to meet the service levels derived earlier, there might be a huge variety of different shifts, depending on starting times, lengths and the moments of the breaks.

4. rostering - shifts assignment to agents.

Due to uncertainty there always arise some fluctuations in anticipated call volume, thus schedules have to be updates in a continuous manner, traffic loads, service levels (SL), to be able to adjust to alterations.

The standard WFM approach, however has a lot of drawbacks. When it is aimed at obtaining the minimum number of agents in each interval, it may result in excessing the minimum of an overall service level. It is reasonable to consider a compensation of the low SL intervals by those ones with a high SL.

Let λ_i be the arrival rate in interval $i \in \{1, \ldots, T\}$, $SL_i(s_i)$ the SL as a function of the number of agents, then the expected daily SL is given by:

$$SL = \sum_{i=1}^{T} \frac{\lambda_i}{\sum_{j=1}^{T} \lambda_j} SL_i.$$

In other words, it is reasonable to overstaff during busy periods and to understaff during quiet periods. This allows for more flexibility when scheduling, leading to cost reductions. Such an approach would integrate the second and third step of the decision process, namely determining the minimum levels s_t and determining the shifts.

Shift scheduling

Shift scheduling constitutes a mathematical model for the optimal employment of personnel. Let us now consider a model for a single-skill environment, proposed by Bhulai, Koole, and Pot [3]. The purpose of this is to implement the sensitivity analysis tools presented in the previous chapter.

It is assumed that time is split into mutually independent T time periods. In each period the system remains in equilibrium. For every time interval t a number s_t is given representing the minimum number of employees needed in interval t.

Arrivals

Call arrivals are usually modelled as an inhomogenous Poisson process, of which parameters depend on time. Let us use as the rate function piecewise constant rates for 60 minutes intervals: λ_t in period $t \in T$. Therefore, there is no stationary situation, but as such a stochastic process converges quickly to its equilibrium, it might be taken as an approximation.

Service times

Since exponential service times stand a good approximation within call centers, let us assume that the service time has an expected distribution with parameter μ_t.

Service level

The service level is defined as the percentage of arrivals that waits less than the acceptable waiting time (AWT) in the queue. The minimal requirement is denoted by α. In practice, most of the time call centers choose $\alpha = 80\%$ and $AWT = 20s$. All calculations are made due to an Erlang C, which plays a central role in call center management, all necessary formulas for calculations are introduced in *Appendix A*.

The number of different shifts is denoted by K. When the shift is being used, then the cost is produced. The aim is to choose a set of shifts, for which there are minimal costs.

Integrated Method

To obtain an optimal solution, computation of both staffing levels and shift scheduling might be integrated, by solving an IP model. An appropriate model for multi-skill environment was proposed by Thompson. Bhulai, Koole and Pot have proposed some modifications for the single-case model.

A new parameter $\gamma_{s,t}$ was introduced, which denotes the expected number of customers, that wait less than the AWT during interval t, when we schedule s agents. We pose a restriction on the overall service level, which is defined as the weighted average of the interval service level.

An additional binary decision $n_{s,t}$ denotes 1 if there are exactly s agents scheduled during interval t, and 0 otherwise. To ensure that in each interval t exactly one of these variables is equal to 1, an additional constraint is proposed, which is obtained by requiring $n_{st} \in \{0,1\}$ and $\sum_{s=0}^{S} n_{st} = 1$ for every t, with S the maximum number of employees that can be scheduled.

The model takes the following form:

$$\begin{array}{ll} \min & \sum_{k\in K} c_k x_k \\ & \\ & \sum_{k\in K} a_{k,t} x_k = \sum_{s\in S} n_{s,t} s \quad t\in T \\ & \sum_{s\in S, t\in T} \lambda_t n_{s,t} \gamma_{s,t} \geq \alpha \sum_{t\in T} \lambda_t \\ & \sum_{s\in S} n_{s,t} = 1 \quad t \in T \\ & x_k \geq 0 \quad \text{and} \quad \text{integer} \quad k\in K \\ & n_{s,t} \in \{0,1\} \quad t\in T, \end{array} \tag{4.1}$$

where it is assumed that $a_{k,t}$ is set to 1 if shift k overlaps time interval t and 0 otherwise, futhermore x_k denotes the number of agents working in shift k, and c_k is the cost associated with an agent working in shift k.

4.1. Implementation of Sensistivity Analysis

Let us introduce the model used for an integrated method to obtain an optimal shift-scheduling (4.1). First of all, the time horizon in our problem is divided into 17 distinct intervals, which represent each hour between 7-24.

The Matrix A represents all possible sorts of shifts, we would like to use in our scheduling. Due to 17-hour day, we are obliged to get an optimal schedule, we have decided to propose not only usual 4-, 8-, 6- 12- hour shifts but we have introduced the more unusual shifts as 2-,3-,5-,7-, and 9-hour as well. The reason is the unique beginning of work in the call centre, namely at 7 am, which made us to fill the gap. Thus we have introduced possible 2-,3-,5-,7-, and 9-hour shifts in the morning.

Furthermore, the matrix A consists of seven 4- hour shifts, the first one starts at 8 am, and each next one begins 2 hours later. This scheme was proposed also for 6-, 8- and 12-hour shifts, and gave us an adequate number of different kinds of shifts.

Let us now introduce the matrix A.

$$\mathbf{A} = \begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

The cost vector is simply proportional to the number of hours a given shift has, thus the cost of a 4-hour shift is set to 4, and analogously for any other kinds of shift:

$$\mathbf{c} = \begin{array}{llllllllll}
(2 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 5 \\
6 & 6 & 6 & 6 & 6 & 6 & 7 & 8 & 8 & 8 \\
8 & 8 & 9 & 12 & 12 & 12) & & & &
\end{array}$$

Mosel Model

All calculations for our model were performed with *Xpress-IVE 7.0* optimizer with usage of an Xpress' modelling language - *Mosel.*

The computational results were obtained on an Intel Pentium Dual CPU 1.73 GHz with 2 GB RAM.

Let us now introduce the *Mosel* code for the main problem (4.1):

```
Objective          sum(k in K)    X(k)*C(k)

forall(t in T)     sum(k in K)    A(k,t)*X(k)= sum(s in S) (N(s,t)*s)
forall(t in T)     sum(s in S)    N(s,t)=1}
   sum(s in S)     sum(t in T)    L(t)*N(s,t)*G(s,t)>= ALFA* sum(t in T) L(t)
forall(k in K)                    X(k) is\_integer
forall(t in T)  forall(s in S)    N(s,t) is\_binary

Minimize(Objective)
```

The Matrix A and the vector C are defined as in the previous subsection. The Matrix G and the vector L were prepared in *MS Excel* using *Erlang C* fomulas that are introduced in *Appendix A.* All the matrices and vectors were stored in separate file, from which they were initialized during the process of the Mosel algorithm.

In our model the dimensions of all matrices are as follows:

$$A \in \{0,1\}^{26\times 17}, c \in \mathbb{R}^{26}, X \in \mathbb{Z}^{26}, G \in \mathbb{R}^{19\times 17}, L \in \mathbb{R}^{17}, N \in \{0,1\}^{19\times 17}.$$

Our program had 35 different constraints, in other words, the main matrix of the program had 35 rows. Furthermore, there were 349 variables, which means that the main matrix had 349 columns.

4.1.1. Optimal Schedules Obtained by Solver

The Xpress-IVE solver only derives the first optimal solution of a B&B algorithm, which is done relatively fast with in less than 1 second time. During the calculation of the mentioned optima, we had stopped at each leaf node, and after the LP relaxation had been solved, the basis was written in the memory.

When an optimal solution had been received, the solver loaded each basis written in the memory, and it solved the entire problem for each basis separately.

This simple procedure means that after an optimal solutions had been found, the entire problem was solved at each node once again, and thanks to that, we have obtained several alternative schedules with the same optimal value, in other words we have received the set of all efficient solutions, described in Fig 1.

A similar technique is used during Primal Analysis, thus the new approach generating all possible efficient solutions stems from an idea of Primal Analysis.

4.1.2. Primal Analysis

The Primal Analysis was described in Chapter 3. It is only different with previous reasoning in two points.

Namely, when the main problem was being solved, at each leaf nodes all variables which had had an integer value of the LP relaxation were written into memory. Those variables are fixed to the found integer variables for each leaf node separately.

The second issue is the fact, that after an optimal solution of the main problem had been derived by a solver, the coefficients of the cost vector c were changed, possibly they could have been altered as well all other coefficients. However, for simplicity purposes we have changed only the mentioned cost vector.

After alteration of this vector, our MIP algorithm was solved at each leaf node. The minimum objective value indicates an upper bound of the new problem (with altered vector c.)

The perturbation of vector c is denoted by Δc and takes form of:

$$\mathbf{\Delta c} = \begin{array}{llllllllll} (5 & 4.5 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 3.5 \\ 3 & 3 & 3 & 3 & 3 & 3 & 2.5 & 2 & 2 & 2 \\ 2 & 2 & 1.5 & 0 & 0 & 0). & & & & \end{array}$$

Our new cost vector becomes the result of $c + \Delta c$:

$$\mathbf{c} + \mathbf{\Delta c} = \begin{array}{llllllllll} (7 & 7.5 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8.5 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9.5 & 10 & 10 & 10 \\ 10 & 10 & 10.5 & 12 & 12 & 12). & & & & \end{array}$$

To compare the obtained results with a “real” optimal value of the new perturbed problem, we have solved the problem with the procedure described in the previous subsection.

All results are depicted in Fig 2.

Time	Cost	X(k)	The Set of All Efficient Solutions																																												
07-09	2	X(1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
07-10	3	X(2)	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08-12	4	X(3)	3	0	4	2	8	2	1	8	3	0	8	0	2	8	1	8	3	5	0	6	7	2	1	1	2	6	2	2	0	1	4	3	3	3	3	7	3	1	8	8	1	1	2	3	3
10-14	4	X(4)	2	1	3	2	2	4	2	1	0	1	2	1	3	2	2	0	3	1	0	0	2	2	1	1	2	1	2	0	1	0	3	0	2	2	0	0	2	1	2	2	0	1	3	1	0
12-16	4	X(5)	2	0	3	2	7	2	0	8	2	0	0	0	1	2	0	0	1	5	0	5	6	0	0	1	1	7	0	0	1	0	2	3	3	2	2	5	3	1	0	1	3	0	1	3	3
14-18	4	X(6)	8	9	8	9	4	9	11	3	7	9	4	11	8	3	8	1	8	3	9	3	4	9	10	8	10	3	8	6	10	8	8	7	8	9	6	0	8	9	4	4	6	10	9	8	6
16-20	4	X(7)	2	0	2	2	1	1	0	2	1	0	0	0	0	2	0	2	0	2	2	0	1	0	0	0	0	2	0	1	0	2	1	2	2	1	2	2	2	2	0	0	2	0	2	2	2
18-22	4	X(8)	5	5	5	5	0	5	1	0	4	5	1	1	5	5	5	5	5	0	5	0	0	5	0	5	0	0	5	2	0	5	5	4	5	5	3	0	5	5	1	0	5	0	5	5	5
20-24	4	X(9)	1	0	1	1	1	0	5	1	0	0	5	5	1	1	1	1	0	1	1	0	0	0	5	1	5	1	0	0	5	1	0	1	1	1	1	1	1	1	5	5	1	5	1	1	1
07-12	5	X(10)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
08-14	6	X(11)	5	7	3	5	0	0	7	0	5	6	0	6	5	0	4	0	4	0	8	2	0	5	7	5	6	2	5	5	8	7	3	5	2	5	5	0	2	7	0	0	4	7	5	5	2
10-16	6	X(12)	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12-18	6	X(13)	0	0	0	0	0	0	0	0	0	0	7	0	0	5	0	8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	7	6	0	0	0	0	0
14-20	6	X(14)	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	1	0	0	1	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16-22	6	X(15)	0	0	0	0	5	0	0	5	0	0	0	0	0	0	0	0	0	2	0	4	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	4	0	0	0	1	0	0	0	0	0
18-24	6	X(16)	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07-14	7	X(17)	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0	0	0	2
08-16	8	X(18)	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10-18	8	X(19)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	2	0	0	0	2
12-20	8	X(20)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14-22	8	X(21)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16-24	8	X(22)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07-16	9	X(23)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08-20	12	X(24)	0	1	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10-22	12	X(25)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0	1	0	0	1	0	0	0	0	3	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	0	0
12-24	12	X(26)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Optimal value:			124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124

Fig 1. The tableau represents the set of all possible alternative schedules generated at each leaf node of a search tree of the main problem, that was solved beforehand.
The first column of the set represents an optimal solution obtained for the original problem beforehand.
Each next column represents an optimal solution found having started from the leaf nodes.

Time	Cost+dCost	X(k)	The Primal Analysis																																					
07-09	7	X(1)	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	0	2	0	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1
07-10	7.5	X(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	4	4	3	4	0
08-12	8	X(3)	0	0	0	0	0	2	0	0	1	0	0	1	1	2	7	7	0	0	1	1	0	0	3	0	3	2	1	3	3	0	0	0	4	0	1	1	0	1	0
10-14	8	X(4)	2	2	0	1	1	0	1	1	0	0	2	0	2	0	1	1	2	2	2	2	2	2	2	0	0	1	0	0	0	3	1	3	0	2	8	8	5	0	1
12-16	8	X(5)	0	0	3	0	0	2	0	2	2	2	0	0	0	7	7	7	0	0	1	1	0	0	2	0	2	2	2	2	2	0	0	0	3	0	0	1	0	1	0
14-18	8	X(6)	0	0	8	0	1	9	9	9	9	9	11	0	11	0	4	4	1	13	9	9	9	9	8	9	8	9	9	8	0	8	1	1	8	0	10	0	0	0	11
16-20	8	X(7)	0	0	2	2	0	2	2	2	2	2	0	0	0	2	0	2	1	0	1	1	1	1	2	1	2	1	1	2	0	2	0	2	2	3	0	1	0	1	0
18-22	8	X(8)	0	0	5	5	5	5	5	5	5	5	1	1	1	0	0	0	6	10	6	6	6	6	5	6	5	6	6	5	0	5	6	5	5	4	0	0	0	0	1
20-24	8	X(9)	0	1	1	1	1	1	1	1	1	1	5	5	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	5
07-12	8.5	X(10)	0	1	1	1	1	0	1	1	1	1	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
08-14	9	X(11)	0	0	5	0	0	3	5	2	2	4	7	7	7	0	1	1	0	6	4	2	2	1	5	3	2	5	4	4	0	0	0	0	4	0	0	0	0	0	6
10-16	9	X(12)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12-18	9	X(13)	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
14-20	9	X(14)	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16-22	9	X(15)	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18-24	9	X(16)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
07-14	9.5	X(17)	0	0	1	0	0	2	0	3	3	3	1	0	0	0	0	0	0	0	2	3	3	4	0	2	3	1	3	0	0	2	0	0	0	0	0	0	0	0	2
08-16	10	X(18)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	2	0	0	0	0	0	1	0	3	0	1	0	0	1	0	1	0	0
10-18	10	X(19)	1	0	0	1	2	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	2	1	1	0	0	0	0	1	0
12-20	10	X(20)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14-22	10	X(21)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	4	0	0
16-24	10	X(22)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07-16	10.5	X(23)	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0
08-20	12	X(24)	8	6	0	7	6	1	0	1	1	0	0	0	0	2	0	0	8	0	0	0	0	0	0	1	0	0	0	0	8	0	6	6	0	5	0	2	3	2	0
10-22	12	X(25)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	0
12-24	12	X(26)	1	3	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	4	1	1	0
Objective value:			150	150.5	225.5	183.5	173.5	233	218.5	227	227	225	224.5	209	229	209	225	237	191	270.5	237	225	225.5	219	236	215	224.5	230.5	231.5	231	155.5	219	181.5	195.5	237	188.5	246	210	179.5	160	224

X(k)		
X(1)	0	0
X(2)	0	0
X(3)	0	0
X(4)	0	0
X(5)	0	0
X(6)	1	0
X(7)	0	0
X(8)	0	0
X(9)	0	0
X(10)	0	1
X(11)	0	0
X(12)	0	0
X(13)	0	0
X(14)	0	0
X(15)	0	0
X(16)	0	0
X(17)	0	0
X(18)	0	0
X(19)	1	2
X(20)	0	0
X(21)	0	0
X(22)	0	0
X(23)	1	0
X(24)	8	8
X(25)	0	0
X(26)	1	1
Objective value:	136.5	136.5

Fig 2. The tableau represents the set of all solutions of the main algorithm, derived at each leaf node having used fixed integer variables, that were obtained beforehand, during the solution of an original problem.
The first column of the set represents the solution with the lowest objective value. Each next column represents the corresponding solution of the algorithm at each node.
The separate two columns indicate two alternative optimal schedules for the problem with altered vector cost.

Appendix A

Erlang C

Let us introduce some basic Erlang C formulas derived in [15]:

$$SL = \mathbb{P}(W_Q \leq t) = 1 - \mathbb{P}(W_Q > t) = 1 - C(s,a)e^{-(s\mu-\lambda)t},$$

where $C(s,a)$ is the delay probability, provided by s servers and the offered load $a = \frac{\lambda}{\mu}$. The $C(s,a)$ might be easily calculated using Erlang B formulations:

$$C(s,a) = \frac{sB(s,a)}{(s - a(1 - B(s,a)))},$$

where $B(s,a)$ stands for the blocking probability, which means that $aB(s,a)$ represents the rejected load, and furthermore $a(1 - B(s,a))$ gives the formula for the load that enters the system, in other words, it is the expected number of the occupied servers.

Calculating $B(s,a)$ is very easy thanks to the following, let $N \sim \text{Poisson}(a)$:

$$B(s,a) = \frac{\mathbb{P}(N = s)}{\mathbb{P}(N \leq s)},$$

and the Poisson distribution N with parameter λ is defined as follows:

$$\mathbb{P}(N = n) = \frac{\lambda^n}{n!}e^{-\lambda}.$$

Bibliography

[1] E. Balas, *Integer programming*, C.A. Floudas, P.M. Pardalos(Eds.): Encyclopedia of Optimization, p. 492-499, Kluwer Academic Publishers, 2001.

[2] H. P. Benson, *Finding Initial Efficient Extreme Point for a Linear Multiple Objective Program*, Journal of the Operational Research Society 32, 1981.

[3] S. Bhulai, G. M. Koole, S. A. Pot, *Simple Methods for Shift-Scheduling in Muli-Skill Call Centers*, MSDM, p. 421-428, 2008.

[4] M.W. Dawande, J.N. Hooker, *Inference-based sensitivity analysis for mixed integer/linear programming*, Principle and Practice of Constraint Programming-CP96, Lecture Notes in Computer Science 1118, Springer 1996.

[5] I. Derpich, J.R. Vera, *Improving the efficiency of the Branch and Bound algorithm for integer programming based on "flatness" information*, European Journal of Operational Research 174, 2006.

[6] W. Dinkelbach, *Sensitivitataanalysen und Parametrische Programmierung*, Springer, Berlin, New York, 1969.

[7] T. Gal, *Postoptimal Analyses, Parametric Programming and Related Topics*, Walter de Gruyter, Berlin, 1995.

[8] T. Gal, H. J. Greenberg, *Advances in Sensitivity Analysis and Parametric Programming*, Kluwer Academic Publishers, Boston, 1997.

[9] F. Glover, G. Kochenberg, *State-of-the-Art Handbook in Metaheuristics*, Kluwer Academic Publishers, Boston, 2003.

[10] P. Hansen, M. Labbe, R. E. Wendell, *Sensitivity Analysis in Multiple Objective Linear Programming: The Tolerance Approach*, European Journal of Operational Research 38, 1989.

[11] J.N. Hooker, *Inference duality as a basis for sensitivity analysis*, Principle and Practice of Constraint Programming-CP96, Lecture Notes in Computere Science 1118, Springer 1996.

[12] J.N. Hooker, *Integer programming duality*, C.A. Floudas, P.M. Pardalos(Eds.): Encyclopedia of Optimization, p. 533-543, Kluwer Academic Publishers, 2001.

[13] Z. Jia, M. G. Ierapetritou, *Short-term scheduling under uncertainty using MILP sensitivity analysis*, Ind Eng Chem Res., 43, 2004.

[14] P. Q. Khanh, N. D. Tuan, *Optimality Conditions for Nonsmooth Multiobjective Optimization Using Hadamard Directional Derivatives*, Journal of Optimization Theory and Applications 133, 2007.

[15] G. Koole, *Optimization of Business Processes: An Introduction to Applied Stochastic Modeling*, Lecture Notes in Optimization of Business Processes, VU University Amsterdam 2010.

[16] E. K. Lee, J. E. Mitchell, *Integer programming: branch and bound methods*, C.A. Floudas, P.M. Pardalos(Eds.): Encyclopedia of Optimization, p. 509-519, Kluwer Academic Publishers, 2001.

[17] G. Mavrotas, D. Diakoulaki, *A branch and bound algorithm for mixed zero-one multiple objective linear programming*, European Journal of Operational Research 107, 1998.

[18] G. Mavrotas, D. Diakoulaki, *A mixed integer multiple objective linear programming model for capacity expansions in an autonomous power generation system*

[19] G. Mavrotas, D. Diakoulaki, *Multi-criteria branch and bound: A vector maximization algorithm for Mixed 0-1 Multiple Objective Linear Programming, Applied Mathematics and Computation 171, 2005*

[20] G. L. Nemhauser, L. A. Wolsey, *Integer and combinatorial optimization*, Wiley, 1988.

[21] T. K. Ralphs, M. Guzelsoy, *Duality and Warm Starting in Integer Programming*, Proceedings of the 2006 NSF Design, Service, and Manufacturing Grantees and Research Conference, 2006.

[22] L. Schrage, L. A. Wolsey, *Sensitivity analysis for branch and bound integer programming*, Operation Research 33, p. 1008-1023, 1985.

[23] S. Sitarz, *Postoptimal analysis in multicriteria linear programming*, European Journal of Operational Research 191, 2008.

[24] S. Sitarz, *Sensitivity Analysis of Weak Efficiency in Multiple Objective Linear Programming*, Asia-Pacific Journal of Operational Resesarch, 2010.

[25] R. Steuer, *Multiple Criteria Optimization Theory: Computation and Application*, John Willey, New York, 1986.

[26] R. J. Vanderbei, *Linear programming: foundations and extensions*, Springer 2008.

[27] R. Vetschera, *Volume-Based Sensitivity Analysis for Multi-Criteria Decision Models*, A. Göpfert A., J. Seeländer, Tammer Chr. (Eds.): Methods of Multicriteria Decision Theory, Hänsel-Hohenhausen, Egelsbach, 1997.

[28] Q. Wei, H. Yan, G. Hao, *Characteristics and structures of weak efficient surfaces of production possibility sets*, Journal of Mathematical Analysis and Applications, 327, 2007.

[29] Winston, W. L. , *Operations Research, Applications and Algorithms*, Duxbury Press, International Thomson Publishing, 3. ed., 1993, ISBN 0-534-20971-8 and 0-534-52020-0.

[30] Y. N. Wu, T. C. E. Cheng, *Convergence Results for Weak Efficiency in Vector Optimization Problems with Equilibrium*, Journal of Optimization Theory and Applications, 2005.

[31] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, 1982.