LARGE-SCALE OPTIMIZATION

Alexander Martin
Darmstadt University of Technology, Germany

Keywords: large-scale optimization; mixed integer program; LP relaxation; Lagrangian Relaxation; Dantzig-Wolfe decomposition; Benders' decomposition; aggregation; projection; modeling issues.

Contents

Summary

In this chapter we present classical methods for the solution of large-scale integer, optimization problems. Among them are LP relaxations and cutting planes, Lagrangian relaxation, Dantzig-Wolfe and Benders' decomposition as well as ideas from lifting and projection. We also discuss some modeling issues and their influence on the solvability of some large-scale problems.

1. Introduction

Optimization deals with the problem of minimizing/maximizing a certain objective subject to some set of side constraints. Such problems appear in everyone's daily life, for instance, when one tries to go from *A* to *B* by plane, train, or car as fast as possible or when one tries to buy some special goods available at different stores as cheap as possible. Optimization problems are mathematically modeled by introducing variables reflecting the options/quantities to be determined and by expressing the objective and the side constraints by functions defined on the domains of the variables. Depending on the characteristics of these functions, one speaks of linear or non-linear optimization problems. If some or all variables are required to be integer the prefix 'integer' is added. In this chapter we restrict our discussion to linear and integer linear optimization problems, i. e., to optimization problems where the objective function and the set of side constraints are linear functions and where some or all variables must be integers. Such problems are expressed in the following form

$$\begin{array}{ll} \min & c^T x \\ s.t. & Ax \le b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{array} \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $p \in \{0, ..., n\}$. If p equals zero, (1) is called a linear program, or LP for short, if $p = n$ we speak of a (pure) *integer linear program* (*IP*), and in all other cases of a *mixed integer program* (*MIP*). MIPs are a powerful tool to model many real-world, optimization problems. For instance, with 0/1 variables, i. e., integer variables that are restricted to take values zero or one, decisions can be modeled. For example, do we produce product *i* (yes or no), do we go from A to *B* by train (yes or no), or do we open a facility at location *i* (yes or no)? For these kinds of questions, we introduce a variable which we set to one, if we say ’yes’ and which we set to zero, if we say ’no’. The main source of 0/1 linear programs comes from combinatorial optimization problems (see *Combinatorial Optimization and Integer Programming*).

The application areas in which MIPs occur are huge, ranging from telecommunication, VLSI-design, production and energy planning to problems in traffic and transport or scheduling. And the number of applications is still increasing.

In this chapter, we concentrate on large-scale MIPs. The term ’large-scale’ is relative. Its meaning changes with time. What has been considered ‘large-scale’ a couple of years ago is now no longer large-scale. Consider, for instance, the traveling salesman problem (TSP), the problem of determining a minimal tour through a given number of cities. In the fifties, a TSP through 49 cities in the US (which corresponds to 1176 variables in the standard IP formulation) has been considered large-scale, today the world record of solving a TSP is 13 509 cities (or 91 239 786 variables). An interesting note is that the method that has been used to solve the problem to optimality at that time is basically the same as the one used today, although various new insights and theoretical results improved this method substantially.

In addition, ’large-scale’ does not only depend on the number of variables or constraints. Very often, problems are considered ‘large-scale’ even if these numbers are moderate, but contain certain structures that are considered difficult for current methods. Therefore we focus in this chapter on methods that have been used and are still used to solve problems considered large rather than on the presentation of some specific large-scale models and their solution techniques.

The basic idea of all methods is to get rid of the part of the problem that makes it difficult. The methods differ in which parts to delete and in the way to reintroduce the deleted parts. In Section 2, we consider linear programming relaxations. Here the integrality constraints on the variables are deleted and the resulting linear program is strengthened by cutting planes. In Section 3, part of the constraint matrix is deleted and put into the objective function attached with some penalties. In Section 4, we discuss decomposition methods, in particular Dantzig-Wolfe and Benders’ decomposition. These methods also delete part of the constraint matrix, reformulate this part and reintroduce the reformulated part into the constraint matrix. So far, we assumed that the problem formulation of (1) is given. However, the one and the same problem can often be modeled in different ways and the methods discussed in Sections 2 through 4 solve sometimes one formulation better than others. In Section 5, we discuss some reformulation techniques, among others aggregation and projection, and show their influence on the solution quality for some examples.

2. LP Relaxations

If we relax the integrality constraints in (1) we obtain the so-called linear programming relaxation of (1):

$$\begin{array}{ll} \min & c^T x \\ s.t. & Ax \le b \\ & x \in \mathbb{R}^n, \end{array} \tag{2}$$

For the solution of linear programs polynomial and efficient methods are known (see *Linear Programming*). In case the optimal LP solution x^* is integral, we solved (1). Otherwise there must be some inequality (called cutting plane) that separates x^* from P_I = conv($\{x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \mid Ax \le b\}$). The problem of finding such inequalities is called separation problem. If we find such an inequality we strengthen the LP relaxation by adding this inequality to the LP and continue. Either we find an optimal solution this way or, if we do not find further inequalities and the optimal LP solution is still fractional, we embed the whole procedure in an enumeration scheme. Details of this so-called cutting plane or branch-and-cut method can be found in *Combinatorial Optimization and Integer Programming*.

The key for the success of this method is to find/know good cutting planes for the polyhedron under consideration. In the chapter *Combinatorial Optimization and Integer Programming* such inequalities are presented, mainly for 0/1 polytopes resulting from applications in combinatorial optimization. We supplement this approach by representing some of the inequalities that are helpful for mixed integer problems, i. e., inequalities that combine integer with continuous variables. We first discuss ways of generating, cutting planes independent of any problem structure. We then look at MIPs with some local structure.

2.3 General Cutting Planes

In the chapter *Combinatorial Optimization and Integer Programming* one particular class of inequalities which can be applied independent of any problem structure has already been discussed for pure integer programs, namely Gomory cuts. Consider again the situation discussed there, where we are given an integer program $\max\{c^T x : Ax = b, x \in \mathbb{Z}_+^n\}$ and an optimal LP solution $x_N^* = 0$ and $x_B^* = A_B^{-1} b - A_B^{-1} A_N x_N^*$ where $B \subseteq \{1, ..., n\}$, $|B| = m$, and $N = \{1, ..., n\} \setminus B$. Consider an index $i \in B$ with $x_i^* \notin \mathbb{Z}$. We use the following abbreviations $\bar{a}_j = A_{i\bullet}^{-1} A_{\bullet j}$, $\bar{b} = A_{i\bullet}^{-1} b$, $f_j = f(\bar{a}_j)$, $f_0 = f(\bar{b})$, where $f(\alpha) = \alpha - \lfloor \alpha \rfloor$ and $\lfloor \alpha \rfloor$ is the largest integer less than or equal to $\alpha \in \mathbb{R}$. Here $A_{i\cdot}$ denotes the i–th row of matrix A and $A_{\cdot j}$ the j–th column. From the fact

$$A_B^{-1}b - A_B^{-1}A_N x_N \in \mathbb{Z} \tag{3}$$

we derive the Gomory cut, (see *Combinatorial Optimization and Integer Programming*)

$$\sum_{j\in N} f_j x_j \geq f_0 \tag{4}$$

It is valid for P_I = conv$\{x \in \mathbb{Z}_+^n : Ax = b\}$ and cuts off x^*. This inequality is no longer valid if continuous variables are involved, because adding integer multiples to continuous variables is no longer possible. For instance, $\frac{1}{3} + \frac{1}{3}x_1 - 2x_2 \in \mathbb{Z}$ with $x_1 \in \mathbb{Z}_+$, $x_2 \in \mathbb{R}_+$ has a larger solution set than $1_3 + {}^1{}_3 x_1 \in \mathbb{Z}$. Nevertheless, it is possible to derive valid inequalities using the following *disjunctive argument*.

Observation 1

Let $(a^k)^T x \leq \alpha^k$ *be a valid inequality for a polyhedron* P^k *for* $k = 1, 2$. *Then,*

$$\sum_{i=1}^{n} \min(a_i^1, a_i^2) x_i \leq \max(\alpha^1, \alpha^2) \tag{5}$$

is valid for both $P^1 \cup P^2$ *and* conv$(P^1 \cup P^2)$.

This observation, applied in different ways, yields valid inequalities for the mixed integer case. We present three methods that are all more or less based on Observation 1.

Gomory's Mixed Integer Cuts

Consider again the situation in (3). Expression (3) is equivalent to $\sum_{j\in N} \bar{a}_j x_j = f_0 + k$ for some $k \in \mathbb{Z}$. We distinguish two cases, $\sum_{j\in N} \bar{a}_j x_j \geq 0$ and $\sum_{j\in N} \bar{a}_j x_j \leq 0$. In the first case,

$$\sum_{j\in N} \bar{a}_j x_j \geq f_0 \tag{6}$$

must hold, where $N^+ = \{j \in N : \bar{a}_j \geq 0\}$ and $N^- = N \setminus N^+$. In the second case, we have $\sum_{j\in N^-} \bar{a}_j x_j \leq f_0 - 1$ which is equivalent to

$$-\frac{f_0}{1-f_0} \sum_{j\in N^-} \bar{a}_j x_j \geq f_0. \tag{7}$$

Now we apply Observation 1 to the disjunction $P^1 = P_I \cap \{x : \sum_{j\in N} \bar{a}_j x_j \geq 0\}$ and $P^2 =$

$P_I \cap \{x : \sum_{j\in N}\bar{a}_j x_j \le 0\}$ with $P_I = \text{conv}\{x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p} : Ax = b\}$. We obtain the valid inequality

$$\sum_{j\in N^+}\bar{a}_j x_j - \frac{f_0}{1-f_0}\sum_{j\in N^-}\bar{a}_j x_j \ge f_0 \tag{8}$$

This inequality may be strengthened in the following way. Observe that the derivation of (8) remains unaffected when adding integer multiples to integer variables. By doing this, we may put each integer variable either in the set N^+ or N^-. If a variable is in N^+, the final coefficient in (8) is $\bar{a}_j$ and thus the best possible coefficient after adding integer multiples is $f_j = f(\bar{a}_j)$. In N^- the final coefficient in (8) is $-\frac{f_0}{1-f_0}\bar{a}_j$ a_j and thus $\frac{f_0(1-f_j)}{1-f_0}$ is the best choice. Overall, we obtain the best possible coefficient by using min(f_j $\frac{f_0(1-f_j)}{1-f_0}$). This yields Gomory's mixed integer cut

$$\begin{aligned}&\sum_{\substack{j: f_j \le f_0 \\ j \text{ integer}}} f_j x_j + \sum_{\substack{j: f_j > f_0 \\ j \text{ integer}}} \frac{f_0(1-f_j)}{1-f_0} x_j \\ &+ \sum_{\substack{j\in N^+ \\ j \text{ non-integer}}} \bar{a}_j x_j - \sum_{\substack{j\in N^- \\ j \text{ non-integer}}} \frac{f_0}{1-f_0}\bar{a}_j x_j \ge f_0.\end{aligned} \tag{9}$$

It can be shown that an algorithm based on iteratively adding these inequalities solves $\min\{c^T x : x \in P_I\}$ in a finite number of steps provided $c^T x \in \mathbb{Z}$ for all $x \in X$. Note also that (9) is at least as strong as (4) in the pure integer case.

Mixed-Integer-Rounding Cuts.

Consider the following basic mixed integer set $X = \{(x, y) \in \mathbb{Z} \times \mathbb{R}_+ : x - y \le b\}$ with $b \in \mathbb{R}$ and the inequality

$$x - \frac{1}{1-f(b)}y \le \lfloor b \rfloor. \tag{10}$$

Inequality (10) is valid for $P_I = \text{conv}(X)$. The validity of this inequality is illustrated in Figure 1. A formal proof can be obtained by applying Observation 1 to the disjunction $P^1 = P_I \cap \{(x, y) : x \le \lfloor b \rfloor\}$ and $P^2 = P_I \cap \{(x, y) : x \ge \lfloor b \rfloor + 1\}$.

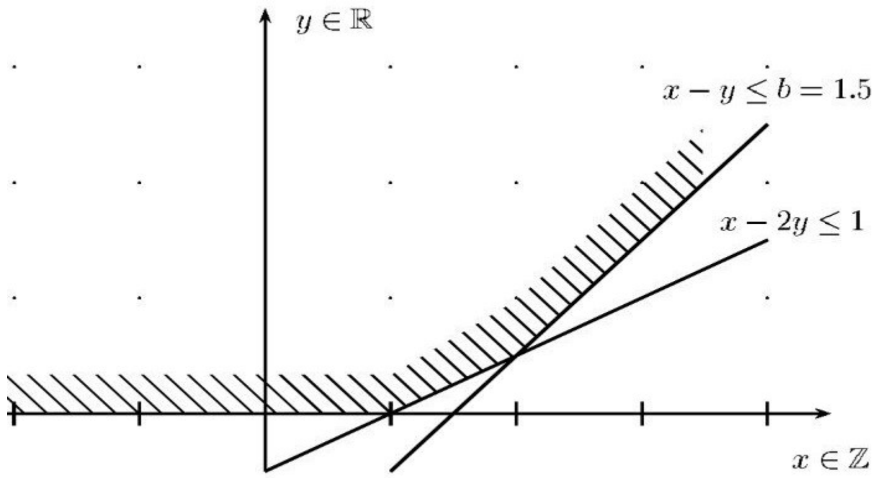

Figure 1: Illustration of basic MIR inequality (10)

The two-dimensional case can be generalized to higher dimensions. Consider the following mixed integer set

$$X = \{(x,\ y)\ \in \mathbb{Z}_+^n \times \mathbb{R}_+ : a^T x\ -\ y \leq b\} \tag{11}$$

with $a \in \mathbb{R}^n$, $b \in \mathbb{R}$. We take $f_i = f(a_i)$ and $f_0 = f(b)$ in the sequel. The inequality

$$\sum_{i=1}^{n}\left(\lfloor a_i \rfloor + \frac{(f_i - f_0)^+}{1 - f_0}\right)x_i - \frac{1}{1 - f_0}\,y \leq \lfloor b \rfloor \tag{12}$$

is called a *mixed integer rounding* (*MIR*) *inequality*, where $v^+ = \max(0, v)$ for $v \in \mathbb{R}$. It is valid for $P_I = \mathrm{conv}(X)$. To see this apply the two-dimensional inequality (10) to the relaxation $w - z \leq b$ of $a^T x - y \leq b$, where $w = \sum_{\{i \in N: f_i \leq f_0\}} \lfloor a_i \rfloor x_i + \sum_{\{i \in N: f_i > f_0\}} \lceil a_i \rceil x_i \in \mathbb{Z}$ and $z = y + \sum_{\{i \in N: f_i > f_0\}}(1 - f_i)x_i \geq 0$.

MIR inequalities imply Gomory's mixed integer cuts (9) when applied to the mixed integer set $X = \{(x, y^-, y^+) \in \mathbb{Z}_+^n \times \mathbb{R}_+^2 : a^T x + y^+ - y^- = b\}$. To see this consider the relaxation $a^T x - y^- \leq b$ of X. Applying (12) yields

$$\sum_{i=1}^{n}\left(\lfloor a_i \rfloor + \frac{(f_i - f_0)^+}{1 - f_0}\right)x_i - \frac{1}{1 - f_0}\,y^- \leq \lfloor b \rfloor. \tag{13}$$

Subtracting the original inequality $a^T x + y^+ - y^- = b$ gives Gomory's mixed integer cut (9). MIR inequalities provide a complete description for any mixed 0/1 polyhedron.

Lift-and-Project Cuts.

The idea of 'lift and project' is to consider the integer programming problem, not in the original space, but in some space of higher dimension (lifting). Then inequalities found in this higher dimensional space are projected back to the original space resulting in tighter integer programming formulations. Versions of this approach differ in how the lifting and the projection are performed. All approaches only apply to 0/1 mixed integer programming problems. The validity of the procedure is based on an easy observation.

Observation 2

If $c_0 + c^T x \geq 0$ *and* $d_0 + d^T x \geq 0$ *are valid inequalities for X, then* $(c_0 + c^T x)(d_0 + d^T x) \geq 0$ *is valid for X.*

Consider a 0/1 integer program $\min\{c^T x : x \in X\}$ with $X = \{x \in \{0, 1\}^p \times \mathbb{R}^{n-p} : Ax \leq b\}$, in which the system $Ax \leq b$ already contains the trivial inequalities $0 \leq x_i \leq 1$ for $i = 1, ..., p$. Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ and $P_I = \text{conv}(X)$. Consider the following procedure.

Algorithm 3 (*Lift-and-Project*)

1. *Select an index* $j \in \{1, ..., p\}$.
2. *Multiply* $Ax \leq b$ *by* x_j *and* $1 - x_j$ *giving*

$$\begin{aligned} (Ax)x_j &\leq bx_j \\ (Ax)(1-x_j) &\leq b(1-x_j) \end{aligned} \tag{14}$$

and substitute $y_i := x_i x_j$ *for* $i = 1, ..., n$, $i \neq j$ *and* $x_j := x_j^2$ *(lifting).*
Call the resulting polyhedron $L_j(P)$.

3. *Project* $L_j(P)$ *back to the original space by eliminating variables* y_i. *Call the resulting polyhedron* P_j.

It can be shown that the j–th component of each vertex of P_j is either zero or one. Now apply Algorithm 3 to P_j by selecting some other index. After repeating this procedure n times, P_I is obtained.

The problem that remains in order to implement Algorithm 3 is to carry out Step 3. Let $L_j(P) = \{(x, y) : Dx + By \leq d\}$. Then the projection of $L_j(P)$ onto the x-space can be described by

$$P_j = \{x : (u^T D)x \leq u^T d \text{ for all } u \in C\} \tag{15}$$

where $C = \{u : u^T B = 0, u \geq 0\}$. Thus, the problem of finding a valid inequality in Step 3 of Algorithm 3 that cuts off a current (fractional) solution x^* can be solved by the linear program

$$\begin{aligned} \max \quad & u^T(Dx^* - d) \\ & u \in C. \end{aligned} \tag{16}$$

C is a polyhedral cone and thus the linear program (16) is unbounded, if there is a violated inequality. For algorithmic convenience, C is often truncated by some "normalizing set". If an integer variable x_j that attains a fractional value in a basic feasible solution is used to determine the index j in Algorithm 3, then an optimal solution to (16) indeed cuts off x^*.

Observation 2 can be applied to a more general setting by multiplying $Ax \le b$ not only with x_j and $1 - x_j$, but with products of higher order of the form $\left(\prod_{j\in J_1} x_j\right)\left(\prod_{j\in J_2}(1-x_j)\right)$ such that $J_1, J_2 \subseteq \{1, \ldots, n\}$ are disjoint and $|J_1 \cup J_2| = d$ for some fixed value $d \ge 1$.

We want to emphasize here that in contrast to the pure integer case none of the cutting plane procedures presented yields a finite algorithm for general mixed integer programs. It is still an open question whether such a procedure exists.

-
-
-

Bibliography

Applegate D., Bixby R. E., Chvátal V., and Cook W. (1998). The solution of traveling salesman problems. *Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung, International Congress of Mathematicians*, pp. 645-656. [Gives a nice impression on the skills that are necessary to solve large-scale TSPs, see Section 6.]

Balas E., Ceria S., and Cornuéjols G. (1993). A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, **58**(295), pp. 324. [Contains all the details about lift-and-project cuts discussed in Section 2.1.]

Dell'Amico M., Maffioli F., and Martello S. (1997). *Annotated Bibliographies in Combinatorial Optimization*. Chichester: Wiley. [A good reference book including literature on decomposition and column generation.]

Goemans M. X. and Myung Y. (1993). A catalog of Steiner tree formulations. *Networks*, **23**(19), pp. 28. [Compares various formulations for the Steiner tree problem discussed in Section 5.]

Williams H. P. (1990). *Model building in mathematical programming*. Chichester: Wiley. [The book discusses modeling issues in detail (See Section 5) as well as pros and cons of various formulations.]

Martin R. (1999). *Large-scale linear and integer programming: A unified approach*. Kluwer. [This is the book form of this chapter. The methods described here and many more are extensively discussed.]

Nemhauser G. L. and Wolsey L. A. (1988). *Integer and Combinatorial Optimization*. Chichester: Wiley. [Complements Schrijver's book on integer programming. It also focuses on practical issues with many illustrating examples.]

Schrijver A. (1986). *Theory of Linear and Integer Programming*. Chichester: Wiley. [A comprehensive book with excellent historical notes on the theory of linear and integer programming.]

Biographical Sketch

Alexander Martin studied mathematics and economics at the University of Augsburg, where he got his diploma in 1988. After doctoral studies in Augsburg and at the Konrad-Zuse-Zentrum fur Informationstechnik Berlin (ZIB) he received his ¨ Ph.D. in Mathematics from the Berlin University of Technology in 1992, where he also did his "Habilitation" in 1998. From 1998-2000, he was Deputy-Head of the Optimization Group at ZIB. Since January 2000, he has been a full Professor at the Darmstadt University of Technology. He was Visiting Professor at Rice University in Houston and at the University of Trier, Germany. His main research interests are the solution of hard combinatorial optimization and integer programming problems, such as graph partitioning, knapsack, network design, or frequency assignment problems. Dr. Martin has experiences with many industrial projects, especially in the areas of telecommunication and energy management.