Global Optimization

Hoang Tuy[1], Steffen Rebennack[2], and Panos M. Pardalos[3]

[1] Department of Optimization and Control, Institute of Mathematics, VAST, Vietnam, htuy@math.ac.vn
[2] Division of Economics & Business, Colorado School of Mines, USA, srebenna@mines.edu, WWW home page: http://www.rebennack.net
[3] Center for Applied Optimization, Department of Industrial & Systems Engineering, University of Florida, USA, pardalos@ufl.edu, WWW home page: http://www.ise.ufl.edu/pardalos/

1 Introduction

Consider an optimization problem of the general form

$$\min \left\{ f(x) \mid g_i(x) \leq 0,\ i = 1, \ldots, m,\ x \in X \right\} \tag{P}$$

where X is a closed convex set in $\mathbb{R}^n$, $f : \Omega \to \mathbb{R}$, and $g_i : \Omega \to \mathbb{R}$, $i = 1, \ldots, m$, are continuous functions defined on some open set Ω in $\mathbb{R}^n$ containing X. Setting

$$D = \left\{ x \in X \mid g_i(x) \leq 0,\ i = 1, \ldots, m \right\}$$

the problem can also be written as

$$\min \left\{ f(x) \mid x \in D \right\}.$$

Any point $\bar{x} \in D$ is called a *feasible solution* of the problem. A feasible solution $\bar{x}$ is called a *global optimal solution* if it is the best of all feasible solutions, *i.e.*, if it satisfies

$$f(\bar{x}) \leq f(x) \quad \forall x \in D. \tag{1}$$

A feasible solution $\bar{x}$ is called a *local optimal solution* if it is the best among all feasible solutions in some neighborhood of it, *i.e.*, if there exists a neighborhood W of $\bar{x}$ such that

$$f(\bar{x}) \leq f(x) \quad \forall x \in D \cap W. \tag{2}$$

Many important practical problems may have many local optimal solutions with different objective function values. The need may then arise to find the best among them, *i.e.*, a global optimal solution.

In convex (linear, resp.) programs where $X = \mathbb{R}^n$, $f(x)$, $g_i(x)$ are all convex (linear, resp.) any local optimal solution is global, and efficient algorithms are routinely used for solving these problems. Aside from these cases, finding a global optimal solution is a hard problem requiring quite different techniques. Due to its importance for many applications, global optimization is an active research field (Horst and Pardalos 1995; Floudas and Gounaris 2009; Pardalos and Coleman 2009).

Some typical examples of global optimization applications in operations research and management science (OR/MS) are presented to show the general mathematical structure of global optimization problems. Next, the general concept of branch and bound, the most popular method for solving global optimization problems, is described. Finally, different classes of global optimization problems encountered in applications and requiring specialized treatment are reviewed.

2 Examples from OR/MS

In most mathematical programming problems of form (P) encountered in OR/MS, each function $f(x),\ g_i(x)$ represents a cost, a performance (return, benefit, ...), or a negative utility (loss, pollution, ...) that depends on the decision variable $x \in \mathbb{R}^n$. The decision maker may want to determine $x \in \mathbb{R}^n$ so as to minimize the associated cost $f(x)$ subject to constraints $g_i(x) \leq 0$ expressing, for example, the requirement that the total cost of i-th resource involved should not exceed certain acceptable limit, or the expected i-th utility should not be less than a certain required level (in addition to constraints $x \in X$ reflecting other "technical" aspects of the problem). A common phenomenon is that economy of scale (or decreasing return) prevails in certain sectors or within certain scale limits, while diseconomy of scale (or increasing return) prevails in other sectors or beyond certain scale limits. Mathematically, economy of scale or decreasing return is modeled by concave or decreasing functions, diseconomy of scale or increasing return is modeled by convex or increasing functions. In more complex situations increasing and decreasing return phenomena may be copresent, and more general types of functions have to be used that are *dc functions* (differences of convex functions), or *dm functions* (differences of monotonic increasing functions).

Recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *convex* if $f(\alpha x+(1-\alpha)x') \leq \alpha f(x)+(1-\alpha)f(x')$ for any $x, x' \in \mathbb{R}^n$, and any real number α such that $0 \leq \alpha \leq 1$; it is said to be *increasing* if $f(x) \leq f(x')$ for any $x, x' \in \mathbb{R}^n$ such that $x_i \leq x'_i,\ i = 1, \dots, n$. A function $f(x)$ is said to be *concave* if $-f(x)$ is convex; *decreasing* if $-f(x)$ is increasing.

Global optimization has been applied to many fields such as biomedicine, economics, energy systems, computational chemistry and biology, and computer science (Floudas and Pardalos 2003; Rebennack et al. 2010a; Rebennack et al. 2010b). Three examples in OR/MS are presented here.

Example 1. (Production-transportation planning) Consider k factories producing a certain good to satisfy the demands $d_j, j = 1, \dots, m$, of m destination points. The production cost is $g(y_1, \dots, y_k)$ if the factory i produces y_i units, where $g(\cdot)$ is a concave function because of economies of scale. The transportation cost is a function $c_{ij}(x)$ for x units shipped from factory i to destination point j. In addition, there is a shortage penalty $h(z_1, \dots, z_m)$ to be paid if the destination point j receives $z_j \neq d_j$ units, where $h(z_1, \dots, z_m) = \sum_{j=1}^m h_j(z_j)$, with $h_j(z_j) \leq 0$ if $z_j \geq d_j$ and $h_j(\cdot)$ is a decreasing nonnegative function in the interval $[0, d_j)$. Usually, the penalty function $h(\cdot)$ is convex, so the total production-transportation cost is the function $f(x, y, z) = \sum_{i=1}^k \sum_{j=1}^m c_{ij}(x_{ij}) + g(y) + h(z)$. This function should be minimized subject to usual transportation constraints: $\sum_{j=1}^m x_{ij} = y_i, i = 1, \dots, k,\ \sum_{i=1}^k x_{ij} = z_j,\ j = 1, \dots, m,\ x_{ij} \geq 0, y_i, z_j \geq 0\ \forall i, j$.

Even if the transportation costs $c_{ij}(x)$ are linear, this problem cannot be handled successfully by conventional methods of nonlinear programming. Things become more complicated when the transportation costs $c_{ij}(x)$ along certain arcs (ij) are dc functions (for instance S-shaped functions), or some are concave, others are convex (Holmberg and Tuy 1993).

Example 2. (Location planning) A facility has to be constructed to serve n users located at points $a^j \in S$ of the plane (*i.e.*, $S \subset \mathbb{R}^2$). If the facility is located at $x \in S$, then the attraction of the facility to user j is $q_j\big(h_j(x)\big)$, where $h_j(x) = \|x - a^j\|$ is the distance from x to a^j and $q_j : \mathbb{R} \to \mathbb{R}$ is a convex decreasing function (the farther x is away from a^j the less attractive it looks to user j). To determine the location of the facility with maximal total attraction, one has to solve the problem

$$\text{maximize} \sum_{j=1}^n q_j\big(h_j(x)\big) \quad \text{s.t. } x \in S. \tag{3}$$

The function $\varphi(x) = \sum_{j=1}^n q_j\big(h_j(x)\big)$ is generally neither convex nor concave. However, since the functions $h_j : \mathbb{R}^2 \to \mathbb{R}_+$ and $q_j = \mathbb{R} \to \mathbb{R}$ are convex, it can be shown that their compositions $q_j\big(h_j(x)\big),\ j = 1, \dots, n$, are dc, and $\varphi(x)$ a sum of dc functions, hence itself a dc function.

In practice, some of the points a^j may actually be repulsion rather than attraction points. For example, there may exist a garbage dump, a sewage plant, or a nuclear plant in the area, and one may wish the facility to be located as far away from these points as possible. If J_1 is the set of attraction points, J_2 the set of repulsion points, then instead of (3) one should seek to maximize the function

$$\sum_{j\in J_1} q_j\big(h_j(x)\big) - \sum_{j\in J_2} q_j\big(h_j(x)\big). \tag{4}$$

An even more complex situation occurs when several facilities must be located. In this case, each user will be served by the nearest facility, so the problem is to determine the locations, say x, y and z, of the facilities, so as to maximize

$$\sum_{j=1}^{n} q_j\left(\tilde{h}_j(x, y, z)\right) \tag{5}$$

over $(x, y, z) \in S \times S \times S$, where

$$\tilde{h}_j(x, y, z) = \min\big\{h_j(x), h_j(y), h_j(z)\big\}. \tag{6}$$

Again it can be proven that $\tilde{h}_j(x, y, z)$ (pointwise minimum of finitely many convex functions) is a dc function, and $q_j\left(\tilde{h}_j(x, y, z)\right)$ (convex functions of dc functions) are also dc functions. Therefore, again this multifacility location problem appears to be a dc optimization problem (Chen et al. 1992). Note that this problem is sometimes formulated as a mixed integer program, much more difficult to solve.

Example 3. (Multilevel programming) Many decentralized decision-making systems in economics and other fields involve a "leader" (the higher level decision maker) who controls a variable $x \in \mathbb{R}^p$ and a "follower" (the lower level decision maker) who controls a variable $y \in \mathbb{R}^q$. For each decision x made by the leader, the follower chooses y in order to optimize his own objective function $\varphi(y)$, under a constraint set $\Omega(x)$ associated with the decision x, *i.e.*, the response y of the follower to the decision x of the leader is a vector such that

$$y \in \operatorname{argmin}\big\{\varphi(y') \mid y' \in \Omega(x)\big\}.$$

If the objective of the leader is to minimize a function $f(x, y)$, while his own constraint set is $D \subset \mathbb{R}^p \times \mathbb{R}^q$ then the problem he must solve is to choose x so as to

$$\text{minimize} \quad f(x, y) \tag{7}$$

$$\text{subject to} \quad (x, y) \in D,\ y \in \Omega(x); \tag{8}$$

$$\varphi(y) \le \varphi(y')\ \forall y' \in \Omega(x). \tag{9}$$

Even in the simplest cases when all data are linear: $f(x, y) = c_1 x + d_1 y, \varphi(y) = d_2 y, D = \big\{(x, y) \mid A_1 x + B_1 y \le g_1,\ x \in R^p_+\big\}$ while $\Omega(x) = \big\{y \mid A_2 x + B_2 y \le g_2,\ y \in R^q_+\big\}$, the feedback relation between upper and lower levels creates nonconvexities that cannot be easily handled by standards methods of nonlinear programming.

If $h(x)$ denotes the optimal value of the lower subproblem, *i.e.*, $h(x) = \min\big\{\varphi(y) \mid y \in \Omega(x)\big\}$, then the constraint (9) can also be written as

$$\varphi(y) \le h(x).$$

Often $\varphi(y)$ is a convex function and, as in the just mentioned linear case, for any x^1, x^2 and any $\alpha \in [0, 1]$, then $\alpha\Omega(x^1) + (1-\alpha)\Omega(x^2) \subset \Omega(\alpha x^1 + (1-\alpha)x^2)$. It can then easily be checked that $h(x)$ is a convex function, too, and so the constraint (9), *i.e.*, $\varphi(y) - h(x) \le 0$, is a dc constraint.

Bilevel and more generally, multilevel programming problems of the above kind have various applications in economics (Stackelberg duopoly model), agriculture (*e.g.*, fertilizer supply, water supply, agricultural policy), financial management, and network design (Wen and Hsu 1991). Thus a host of problems of practical interest in economics, OR/MS and engineering involve dc functions or dm functions in their description. Other problems reported from computer science (VLSI chip design, databases), wireless communications, system reliability, mechanics (structural optimization), physics (nuclear design, microcluster phenomena in thermodynamics), chemistry (phase and chemical reaction equilibrium), or ecology (design and cost allocation for waste treatment systems) can analogously be identified as dc or dm optimization problems (Floudas and Pardalos, 1999; Floudas 2000).

A problem (P) where all functions $f(x), g_1(x), \ldots, g_m(x)$ are dc (dm, resp.) is called a dc (dm, resp.) optimization problem. These are two basic classes of global optimization problems. Practically, every global optimization can be reformulated, possibly via a change of variables, as a dc or a dm optimization problem.

3 Branch and Bound Methods

A popular method for solving a global optimization problem (P) is by using a BB (branch and bound) procedure.

3.1 The Generic BB Procedure

For convenience assume that $X = [a, b] := \{x \in \mathbb{R}^n |\ a_j \leq x_j \leq b_j,\ j = 1, \ldots, n\}$, so that the feasible set in problem (P) is

$$D = \{x \in [a, b] \mid g_i(x) \leq 0,\ i = 1, \ldots, m\}.$$

The generic BB procedure for solving (P) involves two basic operations: *partitioning* and *bounding*.

• **Partitioning**: Starting from the initial box (hyperrectangle) $M_1 = [a, b]$, at each iteration a box is selected and subdivided into two subboxes according to a subdivision rule. Through this partitioning process, a tree is generated with the root at M_1 and the nodes represented by the subboxes that appear as successive descendants of the initial box.

Let $M = [p, q]$ be a box selected for subdivision in a given iteration. A common subdivision rule called the *standard bisection* consists in dividing M into two equal subboxes using a hyperplane perpendicular to a longest edge of M at the midpoint of this edge. An important property of this subdivision rule is its exhaustiveness, meaning that any infinite nested sequence of boxes M_k generated by it shrinks to a point (*i.e.*, diam $M_k \to 0$ as $k \to +\infty$).

• **Bounding**: At each iteration, two new boxes appear as a result of the subdivision operation. For each new box $M = [p, q]$ a lower bound is computed for $f(x)$ over the feasible points in M, *i.e.*, a number $\beta(M) \in \mathbb{R} \cup \{-\infty, +\infty\}$ satisfying

$$\beta(M) \leq \inf \{f(x) \mid x \in M \cap D\} \quad (10)$$

$$M_k \cap D = \emptyset \Rightarrow \beta(M) = +\infty. \quad (11)$$

The latter condition, which is essential, amounts to requiring that $\beta(M) < +\infty$ only if $M \cap D \neq \emptyset$.

The number $\beta(M)$ is usually computed by considering an *underestimator* $\varphi(x)$ of $f(x)$ over a set $\Omega \supset M$, *i.e.*, a function satisfying $\varphi(x) \leq f(x)\ \forall x \in \Omega$, and taking $\beta(M) = \inf \{\varphi(x) \mid x \in \Omega\}$, where $\varphi(x)$ and Ω are chosen so that the latter problem can be solved easily. Also to obtain tight bounds it is often necessary to replace the partition set M by a suitable smaller set $M' \subset M$: the procedure is then referred to as a *branch-and-reduce* algorithm.

While computing the lower bounds, it may happen that some feasible points are obtained: the feasible point with smallest objective function value is then recorded as the current best feasible solution (CBS) and the associated objective function value as the current best objective function value (CBV). Once every current box has been assigned a lower bound, all boxes M with $\beta(M) > CBV$ (in particular those with $\beta(M) = +\infty$) are pruned (deleted). If no box remains after that, the procedure is terminated and CBS gives a (global) optimal solution. Otherwise, a box with smallest lower bound among all remaining boxes is selected for further subdivision, and a new iteration is started.

Proposition. *If bounding is consistent with branching in the sense that*

$$\beta(M) - \min \{f(x) \mid x \in M \cap D\} \to 0 \quad \textit{as diam } M \to 0 \quad (12)$$

then as $k \to +\infty$, the box M_k with smallest lower bound at iteration k shrinks to a point which is an optimal solution, while $\beta(M_k)$ tends to the optimal value min(P) of the problem.

3.2 Case of a Nice Feasible Set

Suppose the feasible set is nice, *i.e.*, such that a feasible solution can be computed cheaply (as is the case, for example, when each $g_i(x)$ is convex, or each $g_i(x)$ is increasing). Then at each iteration k, a current best feasible solution CBS is available that provides an upper bound for min (P). In that case $\beta(M_k) \leq \min(P) \leq$ CBS, so if x^k is CBS at iteration k, then as $k \to +\infty$ the sequence $\{x^k\}$ tends to a limit $\bar{x}$ yielding an optimal solution of the problem. Furthermore, this convergence can be sped up by using instead of the standard bisection an *adaptive* bisection rule ensuring convergence without condition (12) (Tuy 1998, 2000). Given a tolerance $\eta > 0$, by stopping the procedure when $f(x^k) - \beta(M_k) \leq \eta$, an *$\eta$-optimal solution* of the problem, *i.e.*, a feasible solution x^* satisfying $f(x^*) \leq \min(P) - \eta$, is obtained. Thus, everything works well if the feasible set is nice.

3.3 Case of a Hard Feasible Set

By contrast, if the feasible set is such that a feasible solution cannot be computed at cheaply, then several difficulties may arise with the generic BB method. Namely, in this case it may not be easy to compute lower bounds satisfying conditions (11) and (12), while failing these conditions the algorithm may converge to an incorrect solution which is infeasible and quite far from the optimum. Another drawback is that since at every iteration no feasible solution is available, no partition set can be pruned aside from those M with $\beta(M) = +\infty$, causing an excessive growth of the size of the collection of partition sets to be stored. Also, the convergence accomplished with an exhaustive subdivision process is in general slow. As a result, in finitely many steps the BB procedure can at best give an (ε, η)-approximate optimal solution, *i.e.*, an $\bar{x}$ satisfying $g_i(\bar{x}) \leq \varepsilon, i = 1, \ldots, m$ and $f(\bar{x}) \leq \min(P) + \eta$. Unfortunately, such an (ε, η)-optimal solution is not guaranteed to be feasible and close to the true optimum, and, moreover, it may change drastically upon a small change of the tolerances ε and η, causing numerical instability problems in practical implementation.

The way out of these difficulties is to reduce any problem (P) with hard feasible set to a sequence of problems with nice feasible set. This is possible by using the following result (Tuy 2010):

By simple manipulations, any dc optimization problem (i.e., any problem (P) *where $f(x)$ and all $g_i(x)$ are dc) can be reformulated as an equivalent dc optimization problem with a* convex *objective function. Likewise, any dm optimization problem (i.e., any problem* (P) *where $f(x)$ and all $g_i(x)$ are dm) can be reformulated as an equivalent dm optimization problem with an* increasing *objective function.*

So for studying a dc optimization problem (P) one can without loss of generality assume that $f(x)$ is convex. Setting $g(x) = \min_{i=1,\ldots,m} g_i(x)$ the problem (P) can then be written as

$$\min\big\{f(x) \mid g(x) \leq 0,\ x \in [a,b]\big\}, \tag{P$'$}$$

where $g(x)$ is still a dc function by a known property of dc functions (Tuy 1998). Now, given any number $\gamma \geq \min\,(\mathrm{P}')$, consider the problem

$$\min\big\{g(x) \mid f(x) \leq \gamma,\ x \in [a,b]\big\}. \tag{Q$_\gamma$}$$

Since $f(x)$ is convex, this problem has a nice feasible set and can be solved by the above BB method. Clearly $\min\,(\mathrm{Q}_\gamma) \leq 0$ (because $\gamma \geq \min\,(\mathrm{P}')$) and if the problem (P') is such that

$$\min\,(\mathrm{P}') = \inf\big\{f(x) \mid g(x) < 0,\ x \in [a,b]\big\}$$

(a condition satisfied in most cases), it can easily be shown that $\min\,(\mathrm{Q}_\gamma) = 0$ only if $\min\,(\mathrm{P}') = \gamma$.

Based on this relationship between (P') and (Q_γ), the following method (Tuy 2010) can be used to find an η-optimal solution of (P'), *i.e.*, a feasible solution x^* of (P') such that $f(x^*) \leq \min\,(\mathrm{P}') - \eta$.

Suppose a feasible solution $\bar{x}$ of (P') is available. With $\gamma = f(\bar{x}) - \eta$, apply the above described BB procedure for solving (Q_γ). If at some iteration k of this BB procedure, the current best solution

x^k satisfies $g(x^k) \leq 0$, then x^k is a feasible solution of (P′) such that $f(x^k) \leq f(\bar{x}) - \eta$. Otherwise, $g(x^k) > 0\ \forall k$ and from (12) it follows that $\min(\mathrm{Q}_\gamma) = \lim g(x^k) \geq 0$, hence $\min(\mathrm{P}') = \gamma = f(\bar{x}) - \eta$, *i.e.*, $\bar{x}$ is an η-optimal solution of (P′).

Thus, given a feasible solution $\bar{x}$ of (P′), by solving (Q_γ) with $\gamma = f(\bar{x}) - \eta$ one can either identify $\bar{x}$ as an η-optimal solution of (P′) or find a feasible solution x of (P′) such that $f(x) \leq f(\bar{x}) - \eta$. Since $\eta > 0$, by repeating this procedure finitely many times, one will eventually obtain an η-optimal solution of (P′).

To obtain an initial feasible solution $\bar{x}$ of (P′), it suffices to take any number $\gamma > \min(\mathrm{P}')$ and apply the BB procedure for (Q_γ).

That is the basic idea of *successive incumbent transcending* for solving any dc optimization problem (P) with a hard feasible set. An analogous method can be used for solving a dm optimization problem with a hard feasible set: first rewrite it in the form (P′) with an increasing function $f(x)$ and a dm function $g(x)$; then find an η-optimal solution of (P′) by successive incumbent transcending via solving problem (Q_γ) with adaptively adjusted γ.

4 Specific Problem Classes

Aside from general purpose methods for global optimization, more efficient specific methods are available to solve specific problems by exploiting their underlying mathematical structure (Pardalos and Romeijn 2002). Other cases with specialized algorithms not discussed below include concave minimization and optimization with differences of monotonic functions or Lipschitz functions (Horst and Pardalos 1995). Furthermore, there is a rich body of literature on tailored decomposition algorithms for global optimization problems (Rebennack et al. 2009).

4.1 Quadratic optimization

A quadratic optimization problem is a problem (P) where $f(x), g_1(x), \ldots, g_m(x)$ are quadratic functions, *i.e.*, $f(x) = \frac{1}{2}\langle x, Q^0 x\rangle + c^0, x\rangle$, $\quad g_i(x) = \frac{1}{2}\langle x, Q^i x\rangle + \langle c^i, x\rangle + d_i$ with $Q^i, i = 0, 1, \ldots, m$, being symmetric $n \times n$ matrices and $c^i \in \mathbb{R}^n,\ d_i \in \mathbb{R}$.

To solve a quadratic optimization problem by BB method, the basic question is how to compute lower bounds. Two most used bounding methods are reformulation-linearization (Sherali and Adams, 1999), and Lagrange relaxation (Tuy 1998; Floudas 2000).

• **Reformulation-Linearization**: Setting $x_i x_j = y_{ij}$ for every (i, j) with $1 \leq i \leq j \leq n$ and $y = \{y_{ij}\}$, every quadratic function $f(x)$ of $x \in \mathbb{R}^n$ can be expressed as an affine function of x, y, denoted by $[f(x)]_\ell$. For example, $[2x_1x_3 + 3x_1^2 - 5x_2x_3 + 8x_3]_\ell = 2y_{13} + 3y_{11} - 5y_{23} + 8x_3$. Moreover, it can be shown that the constraint $p \leq x \leq q$ is equivalent to the system of quadratic constraints $g_{ij}(x) := (x_i - p_i)(x_j - q_j) \leq 0\ \forall i, j = 1, \ldots, n$. Then for $M = [p, q]$, the problem $\min\big\{f(x) \mid g_k(x) \leq 0,\ k = 1\ldots, m,\ p \leq x \leq q\big\}$ can be rewritten as

$$\min\big\{[f(x)]_\ell \mid [g_k(x)]_\ell \leq 0,\ k = 1\ldots, h,\ y_{ij} = x_i x_j,\ 1 \leq i \leq j \leq n\big\}, \tag{13}$$

where the constraints $g_k(x) \leq 0,\ k = m+1, \ldots, h$, include all the just mentioned constraints $g_{ij}(x) \leq 0$. Clearly (13) is a linear program, with the additional nonconvex constraints $y_{ij} = x_1 x_j,\ i \leq i \leq j \leq n$. Therefore, as a lower bound for $\min\big\{f(x) \mid g_k(x) \leq 0,\ k = 1, \ldots, m,\ x \in M\big\}$, one can take

$$\beta(M) = \inf\big\{[f(x)]_\ell \mid [g_k(x)]_\ell \leq 0,\ k = 1\ldots, h\big\}.$$

It can be shown that this bounding operation satisfies (11) and (12), so the generic BB algorithm using this bounding method is correct, although its convergence is generally slow, due to the large number of additional variables introduced. However, there are various ways to improve the method, for example, by adding implied constraints to (13) (Sherali and Adams 1999) and/or using the incumbent transcending approach discussed earlier.

• **Lagrangian Relaxation**: For a given problem (P), the function $L(x, u) = f(x) + \sum_{i=1}^m u_i g_i(x)$, $u \in \mathbb{R}^m_+$ is called the Lagrangian, and the problem

$$\sup_{u \in \mathbb{R}^m_+} \inf_{x \in X} L(x,u) \tag{LP}$$

the Lagrangian relaxation of (P). It is easily seen that $\min(\text{LP}) \le \min(\text{P})$, so $\min(\text{LP})$ is a lower bound of $\min(\text{P})$. The Lagrangian relaxation is called exact if $\min(\text{LP}) = \min(\text{P})$. This occurs effectively, for example, when $m = 1$ and $X = \mathbb{R}^n$, *i.e.*, when the problem is

$$\min\left\{f(x) \mid g(x) \le 0,\ x \in \mathbb{R}^n\right\}$$

where $f(x), g(x)$ are quadratic functions and there is an $x^* \in \mathbb{R}^n$ such that $g(x^*) < 0$. Indeed, it is known that in this special case, the Lagrangian relaxation is exact and equivalent to a SDP (semi-definite programming problem) that can be solved by efficient methods (Ben-Tal and Nemirovski, 2001). So, in particular, the minimization of a nonconvex quadratic function over an ellipsoid is equivalent to a SDP, *i.e.*, essentially a convex problem.

In the general case $m > 1, X = [p,q]$, by replacing the constraint $x \in [p,q]$ with an equivalent system of quadratic inequalities as mentioned above, it can always be assumed that $X = \mathbb{R}^n$. Although the corresponding Lagrange relaxation is also an SDP, it seems difficult to use exclusively this bounding method to produce a convergent BB algorithm. However, it can be incorporated into a convergent primal-relaxed dual decomposition approach (Floudas 2000).

Also note that a quadratic function of the form $\sum_{0 \le i \le j \le n} c_{ij} x_i x_j$ with $c_{ij} \ge 0$ is an increasing function on $\mathbb{R}^n_+$. Therefore any quadratic function on $\mathbb{R}^n_+$ is a dm function, and thus a quadratic optimization problem (P) where $X = \mathbb{R}^n_+$ can be viewed alternatively as a dm problem and as such can be solved by the method described earlier.

4.2 Multiobjective Programming

A multiobjective program is a generalization of problem (P) where $F(x)$ is a vector of k objective functions

$$\min\left\{F(x) = \left[f_1(x), f_2(x), \ldots, f_k(x)\right]^\top \mid g_i(x) \le 0,\ i = 1, \ldots, m,\ x \in X\right\}$$

where X represents the set of feasible decisions and $f_i(x) : \Omega \to \mathbb{R},\ i = 1 \ldots, k$ are the objective functions that the decision maker wants to minimize. A feasible decision x^0 is said to be efficient (Pareto-optimal) if for any $x \in X$, $f_i(x) \le f_i(x^0)\ \forall i$ implies $f_i(x) = f_i(x^0)\ \forall i$; it is said to be weakly efficient if there is no $x \in X$ such that $f_i(x) < f_i(x^0)\ \forall i$. An efficient or weakly efficient solution achieves a kind of equilibrium and in certain situations the decision maker may want to find an equilibrium minimizing some objective function. For instance, starting from a feasible solution, the decision maker may want to reach an equilibrium by the "cheapest" way (see Thach et al. 1996 for an example in bond portfolio optimization). If X_E denotes the set of efficient solutions, then the goal of the decision maker is to minimize a certain function $h(x)$ over X_E, *i.e.*,

$$\text{minimize } h(x) \text{ subject to } x \in X_E. \tag{14}$$

In general, the set X_E is nonconvex, so even if $h(x)$ is linear, this is a difficult global optimization problem (Marler and Arora 2004). A relaxed variant of problem (14) is

$$\text{minimize } h(x) \text{ subject to } x \in X_{WE}, \tag{15}$$

where X_{WE} denotes the set of weakly efficient solutions. It can be shown that when $f_i(x)$ are linear (15) is equivalent to the problem

$$\min\left\{h(x) \mid x \in X,\ \lambda \in \Lambda,\ g(\lambda) - \lambda F(x) \le 0\right\},$$

where Λ is a simplex in R^k and $g(\lambda) = \sup\left\{\lambda F(y) \mid y \in X\right\}$ is a convex function (so $g(\lambda) - \lambda F(x)$ is a dc function). This allows the use of the BB method discussed earlier.

4.3 Fractional Programming

Fractional programming deals with problems where a ratio of two objective functions has to be optimized. Several different forms of fractional programs can be distinguished (Frenk and Schaible 2004; Stancu-Minasian 1997).

• **Single-ratio fractional programs**: For extended real-valued continuous functions $f(x), h(x) : \Omega \to [-\infty, +\infty]$ with finite value on D, single-ratio fractional programs are given in the general form

$$\inf \left\{ \frac{f(x)}{h(x)} \;\middle|\; g_i(x) \le 0,\ i = 1, \ldots, m,\ x \in X \right\}. \tag{16}$$

For the special case of $f(x)$ and $g_i(x)$ are convex functions $\forall i$ and $h(x)$ is a positive concave function in D, (16) is called a single-ratio convex fractional program and is a nonconvex global optimization problem. Note that the ratio $\frac{f(x)}{h(x)}$ is not a convex function in general. Typical applications in OR/MS are the maximization of productivity, maximization of return on investments, maximization of return versus risk, minimization of cost versus time, and maximization of output versus input.

• **Generalized fractional program**: Extending (16) to multiple ratios leads to the generalized fractional programs of the form

$$\inf_{x \in D} \sup_{1 \le l \le k} \frac{f_l(x)}{h_l(x)}, \tag{17}$$

with $f_l(x), h_l(x) : \Omega \to [-\infty, +\infty]$ for all l and positive functions $h_l(x)$ for $x \in D$.

• **Sum-of-ratios fractional program**: Minimizing the sum of ratios leads to the following optimization problem

$$\inf \left\{ \sum_{l=1}^{p} \frac{f_l(x)}{h_l(x)} \;\middle|\; g_i(x) \le 0,\ i = 1, \ldots, m,\ x \in X \right\}, \tag{18}$$

with the same assumptions on the function $f_l(x)$ and $h_l(x)$ as in the generalized fractional programs. Bond portfolio optimization problems are examples for applications of sum-of-ratios fractional programming problems.

4.4 Multiplicative Programming

One standard approach to simultaneously optimizing several objectives without a common scale is to optimize the product of these objectives. This leads to consider multiplicative programming problems of the form

$$\inf \left\{ \prod_{l=1}^{p} f_l(x) \;\middle|\; g_i(x) \le 0,\ i = 1, \ldots, m,\ x \in \mathbb{R}^n \right\} \tag{19}$$

where $f_l : \mathbb{R}^n \to \mathbb{R}^+,\ g_i : \mathbb{R}^n \to \mathbb{R}$. Practical methods for solving these problems are available when each $f_l(x)$ is either quadratic or affine and each $g_i(x)$ is convex, while $p \le 5$ and $n, m \le 100$ (Konno et al. 1997). In particular, linear multiplicative programming problems, *i.e.*, problems (19) where $p = 2$ and all functions f_l, g_i are affine, can be solved very fast by a variant of parametric simplex algorithm. Applications of multiplicative programming include bond portfolio optimization and economic analysis.

References

1. A. Ben-Tal and A. Nemirovski: 2001, *Lectures on Modern Convex Optimization*, SIAM/MPS, Philadelphia.
2. P.-C. Chen, P. Hansen, B. Jaumard & H. Tuy: 1992, "Weber's Problem with Attraction and Repulsion," *Journal of Regional Science*, **32**:467-486.

3. C.A. Floudas: 2000, *Deterministic Global Optimization*, Kluwer Academic Publishers, Dordrecht/Boston/London.
4. C.A. Floudas and C.E. Gounaris: 2009, “A review of recent advances in global optimization,” *Journal of Global Optimization*, **45(1)**:3-38.
5. C.A. Floudas and P.M. Pardalos: 1999, *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Dordrecht/Boston/London.
6. J.B.G. Frenk and S. Schaible: 2004, *Fractional Programming*, Erasmus Research Institute of Management (ERIM), ERS-2004-074-LIS.
7. N.T. Hoai-Phuong and H. Tuy: 2003, “A unified monotonic approach to generalized linear-fractional programming,” *Journal of Global Optimization* **26**:229-259.
8. K. Holmberg and H. Tuy: 1993, “A production-transportation problem with stochastic demands and concave production cost,” *Mathematical Programming* **85**:157-179.
9. R. Horst and P.M. Pardalos: 1995, *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht/Boston/London.
10. R. Horst and H. Tuy: 1996, *Global Optimization* (Deterministic Approaches), 3rd edition, Springer 1996.
11. H. Konno, P.T. Thach and H. Tuy: 1997, *Optimization on Low Rank Nonconvex Structures*, Kluwer Academic Publishers, Dordrecht/Boston/London.
12. R.T. Marler and J.S. Arora: 2004, “Survey of multi-objective optimization methods for engineering,” *Structural and Multidisciplinary Optimization* **26**:369-395
13. P.M. Pardalos and T.F. Coleman (editors): 2009, Lectures on global optimization, vol. 55, Fields institute communications series. American Mathematical Society. 591
14. P.M. Pardalos and H.E. Romeijn (editors): 2002, *Handbook of Global Optimization*, Springer.
15. S. Rebennack, J. Kallrath, and P.M. Pardalos: 2009, “Column enumeration based decomposition techniques for a class of non-convex MINLP problems,” *Journal of Global Optimization*,**43(2-3)**:277-297.
16. S. Rebennack, P.M. Pardalos, M.V.F. Pereira, and N.A. Iliadis (editors): 2010a. *Handbook of Power Systems I*, Energy Systems series, Springer.
17. S. Rebennack, P.M. Pardalos, M.V.F. Pereira, and N.A. Iliadis (editors): 2010b. *Handbook of Power Systems II*, Energy Systems series, Springer.
18. H. D. Sherali and W. P. Adams: 1999, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer Academic Publishers, Dordrecht/Boston/London.
19. I.M. Stancu-Minasian: 1997, *Fractional Programming: Theory, Methods and Applications*, Kluwer Academic Publishers, Dordrecht/Boston/London.
20. P.T. Thach, H. Konno and D. Yokota: 1996, “Dual approach to minimization on the set of pareto-optimal solutions,” *Journal of Optimization Theory and Applications*, **88(3)**:689-707.
21. H. Tuy: 1998, *Convex Analysis and Global Optimization*, Kluwer (Springer) 1998.
22. H. Tuy: 2010, “$\mathcal{DC}$-optimization and robust global optimization,” *Journal of Global Optimization* **47**:485-501.
23. U.-P. Wen and S.-T Hsu: 1991, “Linear Bilevel Programming Problems – A Review,” *The Journal of the Operational Research Society*, **42**:125-133.