A Unified Branch-and-Bound and Cutting Plane Algorithm for a Class of Nonconvex Optimization Problems. Application to Bilinear Programming.

L.D. Muu and W. Oettli

Nr. 88 (1988)

Fakultät für Mathematik und Informatik, Universität Mannheim, D-6800 Mannheim 1, Germany

A Unified Branch-and-Bound and Cutting Plane Algorithm for a Class of Nonconvex Optimization Problems. Application to Bilinear Programming.

L. D. MUU[1,2] and W. OETTLI[1]

University of Mannheim, Germany

Abstract. A unified approach to branch-and-bound and cutting plane methods for solving a certain class of nonconvex optimization problems is proposed. Based on this approach an implementable algorithm is obtained for programming problems with a bilinear objective function and jointly convex constraints.

1. Introduction and Problem Statement

In what follows we describe a combined branch-and-bound and cutting plane algorithm for solving a certain class of nonconvex optimization problems. The algorithm is first described as a conceptual method, without particular reference to implementation. We then specialize it to problems with a bilinear objective function and with jointly convex constraints. In this case all subproblems occurring are convex, and the algorithm appears to be implementable efficiently.

We are going to consider the following problem, denoted by (P):

$$\text{(P)} \qquad \min \{f(x,y) | x \in X, y \in Y, g_j(x,y) \le 0 \quad \forall j \in J\},$$

where we assume that: J is a finite set, X is a (sequentially) compact set, $Y \subset \mathbb{R}^m$ is a compact convex set, the functions f, g_j $(j \in J) : X \times \mathbb{R}^m \to \mathbb{R}$ are continuous on $X \times \mathbb{R}^m$ and convex on $\mathbb{R}^m$ for any fixed $x \in X$.

Of particular interest will be the special case, denoted by (BL), of a bilinear objective function and convex constraints:

$$\text{(BL)} \quad \min \{f(x,y) := \langle p, x\rangle + \langle x, My\rangle + \langle q, y\rangle | x \in X, y \in Y, g_j(x,y) \le 0 \quad \forall j \in J\},$$

where $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^m$ are compact convex sets, g_j $(j \in J) : X \times \mathbb{R}^m \to \mathbb{R}$ are continuous convex functions, M is a given matrix, and p, q are given vectors. This problem can be solved using only convex subprograms. Hence we obtain in this case an implementable algorithm.

Bilinear programs without the joint constraints $g_j(x,y) \le 0$ and with X, Y polyhedra have been studied by several authors, see [2,3,6,7,8,9,10]. All of these methods are based on the fact that a solution is obtained within the vertex-set of X and Y. This property does no longer hold for problems with joint constraints. These methods therefore cannot be adapted to the present more general setting.

Bilinear programs with joint convex constraints have been studied in [1]. The present algorithm, if specialized to the case of bilinear programs, is very much different from the algorithm described in [1]. There the bounding operation is based on using lower convex envelopes of the function $\langle x, My\rangle$, whereas in our case the bounding operation is based on relaxing the constraints.

---

[1] Universität Mannheim, Fakultät für Mathematik und Informatik, D-6800 Mannheim 1, Germany

[2] On leave from Institute of Mathematics, Hanoi, as a fellow of Alexander von Humboldt Foundation

Let us note that the case of an objective function $f(x,y) := \langle x, My\rangle + f_0(x,y)$, where f_0 is convex, can be reduced to the form (BL) by minimizing instead the objective function $\langle x, My\rangle + t$ subject to an additional constraint $f_0(x,y) - t \le 0$.

2.Description of the algorithm

We are going to describe an algorithm for solving problem (P). In addition to the already stated assumptions about (P) we require the following: Y is given as

$$Y := \{y \in \mathbb{R}^m | \varphi(y) \le 0\},$$

where $\varphi : \mathbb{R}^m \to \mathbb{R}$ is convex, and subgradients of φ are supposed to be available. We choose once and for all a partition $J = J_1 \cup J_2$ $(J_1 \cap J_2 = \emptyset)$ of the family of constraints. We select a sequence $\{\varepsilon_k\}$ of positive numbers such that $\varepsilon_k \searrow 0$. We let

$$Y^k := \{y \in \mathbb{R}^m | \varphi(y) \le \varepsilon_k\}.$$

Finally we assume that we dispose of a compact polyhedron $B^0 \subset \mathbb{R}^m$ such that every optimal solution of (P) is contained in $X \times B^0$. Denote by G the feasible region of (P), and let f^* denote the optimal value of problem (P) (we always adopt the convention that an optimal value equals $+\infty$ if no feasible points exist).

Given a closed subset $B \subset B^0$ we define the problem $\mathrm{P}(B)$ as

$$\mathrm{P}(B) \qquad \min \{f(x,y) | x \in X, y \in B, g_j(x,y) \le 0 \quad \forall j \in J\},$$

and we define the relaxed problem $\mathrm{R}(B)$ of $\mathrm{P}(B)$ as

$$\mathrm{R}(B) \qquad \begin{aligned} \min \{f(x,y_1) | x \in X, y_1 \in B, g_j(x,y_1) \le 0 \ \forall j \in J_1, \\ y_2 \in B, g_j(x,y_2) \le 0 \ \forall j \in J_2\}. \end{aligned}$$

By $\beta(B)$ we denote the optimal value of $\mathrm{R}(B)$. Due to our compactness assumptions, whenever $\beta(B) < +\infty$, then $\mathrm{R}(B)$ has an optimal solution (x^B, y_1^B, y_2^B). By α_k we shall denote the least upper bound for f^* known so far in iteration k.

Roughly speaking the algorithm runs as follows: At a typical step k, say, we are given a collection Γ_k of subsets $B^q \subset B^0$, all B^q described by affine inequalities, such that any optimal solution of (P) is contained in $X \times \cup_q B^q$. Since $\beta(B)$ cannot exceed the optimal value of $\mathrm{P}(B)$ this implies that $f^* \ge \min_q \ \beta(B^q)$. For all $B^q \in \Gamma_k$ the relaxed problem $\mathrm{R}(B^q)$ should be solved. If $\beta(B^q) > \alpha_k$, or if $Y \cap B^q = \emptyset$, then B^q is deleted from Γ_k. Out of the remaining sets one, $B^k \in \Gamma_k$ say, is selected such that $\beta(B^k)$ is minimal. It is then clear that $\beta(B^k) \le f^* \le \alpha_k$. Now either we cut off part of B^k, thus obtaining a subset B^k_- of B^k such that $Y \cap B^k_- = Y \cap B^k$, or we bisect B^k, thus obtaining two complementary subsets B^k_+ and B^k_- of B^k. We replace B^k by B^k_- in the first case and by $\{B^k_-, B^k_+\}$ in the second case to obtain Γ_{k+1}. The algorithm may terminate finitely; it will do so in particular if $f^* = +\infty$. The algorithm can now be described in detail as follows.

Algorithm

Initialization. With the given set B^0 solve problem $\mathrm{R}(B^0)$. If $\beta(B^0) < +\infty$, let (x^0, y_1^0, y_2^0) be an optimal solution of $\mathrm{R}(B^0)$. Let $\alpha_{-1} := +\infty$ and let $\Gamma_0 := \{B^0\}$.

Iteration k $(k = 0, 1, \cdots)$. At the beginning of iteration k we have a collection Γ_k of subsets $B^q \subset B^0$ such that every solution of (P) is contained in $X \times \cup \{B^q | B^q \in \Gamma_k\}$. For each

$B^q \in \Gamma_k$ we know $\beta(B^q)$ and, if $\beta(B^q) < +\infty$, we know an optimal solution (x^q, y_1^q, y_2^q) of $\mathrm{R}(B^q)$. Furthermore $\alpha_{k-1} \geq f^*$ is at hand. Set $\alpha_k := \min\{\alpha_{k-1}, \min\{\tau_q | B^q \in \Gamma_k, \beta(B^q) < +\infty\}\}$, where $\tau_q := \min\{f(x^q, \eta) | \eta \in [y_1^q, y_2^q], (x^q, \eta) \in G\}$. Let $\Delta_k := \{B^q \in \Gamma_k | \beta(B^q) \leq \alpha_k \text{ and } Y \cap B^q \neq \emptyset\}$ (from $\alpha_k \geq f^* \geq \min\{\beta(B^q) | B^q \in \Gamma_k, Y \cap B^q \neq \emptyset\}$ follows $\Delta_k \neq \emptyset$). Select $B^k \in \Delta_k$ such that $\beta(B^k) = \min\{\beta(B^q) | B^q \in \Delta_k\}$. Let $\beta_k := \beta(B^k) = f(x^k, y_1^k)$.
If $\beta_k = \alpha_k$, then terminate: $\alpha_k = f^*$ (*Stop 1*).
If $\beta_k < \alpha_k$ then form the two convex sets

$$B_1^k := \{y \in B^k \cap Y^k | g_j(x^k, y) \leq 0 \quad \forall j \in J_1, f(x^k, y) - \beta_k \leq 0\},$$
$$B_2^k := \{y \in B^k \cap Y^k | g_j(x^k, y) \leq 0 \quad \forall j \in J_2\}.$$

We have to distinguish three possible cases:

<u>Case 1</u>: $B_1^k \cap B_2^k \neq \emptyset$.
In this case select $u^k \in B_1^k \cap B_2^k$.
If $u^k \in Y$, then terminate: $\beta_k = f^*$ (*Stop 2*).
If $u^k \notin Y$, then select $t^k \in \partial\varphi(u^k)$ – a subgradient of φ at u^k – and let

$$B_-^k := \{y \in B^k | \varphi(u^k) + \langle t^k, y - u^k \rangle \leq 0\}.$$

Solve $\mathrm{R}(B_-^k)$. Let $\Gamma_{k+1} := \Delta_k \setminus \{B^k\} \cup \{B_-^k\}$. Go to iteration $k+1$.

<u>Case 2</u>: $B_1^k \cap B_2^k = \emptyset$ and $y_i^k \in Y^k$ for all $i \in \{1, 2\}$.
In this case $y_i^k \in B_i^k$ $(i = 1, 2)$. Select an affine function l_k with $\|\nabla l_k\| \leq 1$ and a point $u^k \in B_2^k$ such that

$$(*) \qquad t \cdot \psi_k(u^k) \leq \begin{cases} -l_k(y_1^k) \\ l_k(y_2^k), \end{cases}$$

where $t > 0$ (independent of k) and $\psi_k(y) := \max\{f(x^k, y) - \beta_k, \max_{j \in J_1} g_j(x^k, y)\}$ (from $B_1^k \cap B_2^k = \emptyset$ follows $\psi_k(u^k) > 0$). Set

$$B_-^k := \{y \in B^k | l_k(y) \leq 0\}, \ B_+^k := \{y \in B^k | l_k(y) \geq 0\}.$$

Solve $\mathrm{R}(B_-^k)$, $\mathrm{R}(B_+^k)$. Let $\Gamma_{k+1} := \Delta_k \setminus \{B^k\} \cup \{B_-^k, B_+^k\}$. Go to iteration $k+1$.

<u>Case 3</u>: $B_1^k \cap B_2^k = \emptyset$ and $y_i^k \notin Y^k$ for some $i \in \{1, 2\}$.
In this case select $w^k \in \{y_1^k, y_2^k\}$ such that $\varphi(w^k) = \max_{i=1,2} \varphi(y_i^k)$. Select $t^k \in \partial\varphi(w^k)$ and let

$$B_-^k := \{y \in B^k | \varphi(w^k) + \langle t^k, y - w^k \rangle \leq 0\}.$$

Solve $\mathrm{R}(B_-^k)$. Let $\Gamma_k := \Delta_k \setminus \{B^k\} \cup \{B_-^k\}$. Repeat iteration k (k unchanged).

This completes the description of iteration k.

Remarks. 1. Assume that $J_2 = \emptyset$. Then $B_1^k \cap B_2^k = B_1^k$, hence $B_1^k \cap B_2^k = \emptyset$ implies $B_1^k = \emptyset$ and therefore $y_1^k \notin Y^k$. This means that case 2 (bisection) never occurs, and the algorithm becomes then a pure cutting plane method.

2. Assume that $B^0 = Y$ (ε_k then being irrelevant). Then $B_i^k \subset Y$ $(i = 1, 2)$. Hence $y_i^k \in Y$ $(i = 1, 2)$, and if $u^k \in B_1^k \cap B_2^k$, then $u^k \in Y$. This means that only case 2 or

termination can occur. The algorithm becomes in this situation a pure branch-and-bound method.

3.Examples

We give here two examples for the determination of l_k and u^k as requested under case 2 of the above algorithm.

1. Let $u^k \in B_2^k$ be a solution of min $\{\|u - y_1^k\|^2 | u \in B_2^k\}$, where $\|\cdot\|$ denotes euclidean norm. From $B_1^k \cap B_2^k = \emptyset$ follows $u^k \neq y_1^k$. Now let $r^k := (u^k - y_1^k)/\|u^k - y_1^k\|$, and define

$$l_k(y) := \langle r^k, y\rangle - \langle r^k, (y_1^k + u^k)/2\rangle.$$

Then $\|\nabla l_k\| = 1$, moreover $l_k(u^k) - l_k(y_1^k) = \|u^k - y_1^k\|$ and $-l_k(y_1^k) = l_k(u^k)$. An elementary geometric argument shows that $l_k(u^k) \leq l_k(y)\ \forall y \in B_2^k$, hence in particular $l_k(u^k) \leq l_k(y_2^k)$. Thus we have

$$\|u^k - y_1^k\| = l_k(u^k) - l_k(y_1^k) = 2l_k(u^k) = -2l_k(y_1^k) \leq 2l_k(y_2^k).$$

Assume now that ψ_k is Lipschitz continuous with Lipschitz constant $L > 0$ independent of k. Then, since $\psi_k(y_1^k) \leq 0$, it follows

$$\psi_k(u^k) \leq \psi_k(u^k) - \psi_k(y_1^k) \leq L \cdot \|u^k - y_1^k\|.$$

Hence condition $(*)$ is satisfied with $0 < t \leq (2L)^{-1}$. We remark that the above construction can be used even if Lipschitz continuity is not fulfilled, since we may replace the left-hand side of inequality $(*)$ by $t \cdot \|u^k - y_1^k\|$ without affecting the validity of the convergence theorem given in the next section.

2. The second example is designed towards the case of a bilinear objective function. Assume that $f(x, \cdot)$ is affine, and choose $J_1 = \emptyset$. Then $\psi_k(y) = f(x^k, y) - \beta_k$. Choose $u^k \in B_2^k$ such that $\psi_k(u^k) \leq \psi_k(y_2^k)$. Assume that $\|\nabla\psi_k\| \leq L$ with constant $L > 0$ independent of k (this assumption is satisfied, if $f(x, y)$ is bilinear and the x^k vary in the compact set X). Define

$$l_k(y) := L^{-1} \cdot (\psi_k(y) - \psi_k(u^k)/2).$$

Then $\|\nabla l_k\| \leq 1$, moreover from $\psi_k(y_1^k) \leq 0$ and $\psi_k(y_2^k) \geq \psi_k(u^k)$ follows $l_k(y_1^k) \leq -L^{-1}\psi_k(u^k)/2$ and $l_k(y_2^k) \geq L^{-1}\psi_k(u^k)/2$. Hence condition $(*)$ is satisfied with $0 < t \leq (2L)^{-1}$.

4.Convergence of the algorithm

We turn now to the convergence result for the above algorithm. It is easily seen that for fixed k case 3 can occur only finitely often. Indeed, the cardinality of Γ_k does not increase under case 3, and therefore after finitely many occurences of case 3 one has $\varphi(y_i^k) \leq \varepsilon_k$ $(i = 1, 2)$ – see [5] – so that then either case 1 or case 2 must occur, leading to an increase of k. Hence we may assume without loss of generality, if the algorithm does not terminate finitely, that for each k a couple $(x^k, u^k) \in X \times B^0$ is produced such that (x^k, y_1^k, y_2^k) is a solution of $\mathrm{R}(B^k)$ and u^k obeys the rules of case 1 or case 2. Recall that $\beta_k := f(x^k, y_1^k) = \beta(B^k)$. It is clear that

$$\beta_k \leq \beta_{k+1} \leq f^* \leq \alpha_{k+1} \leq \alpha_k.$$

Theorem.

i) If the algorithm terminates at iteration k with *Stop 1*, then $\alpha_k = f^*$.

ii) If the algorithm terminates at iteration k with *Stop 2*, then $\beta_k = f^*$, and (x^k, u^k) is a solution of (P).

iii) If the algorithm is not finite, then $\beta_k \nearrow f^*$, and every cluster point of $\{(x^k, u^k)\}$ is a solution of (P).

Proof: i) If the algorithm terminates with *Stop 1*, then $\beta_k = \alpha_k$. From $\beta_k \le f^* \le \alpha_k$ follows $f^* = \alpha_k$.

ii) If the algorithm terminates with *Stop 2*, then $u^k \in B_1^k \cap B_2^k$ and $u^k \in Y$, hence $(x^k, u^k) \in G$ and $f(x^k, u^k) - \beta_k \le 0$. So $f^* \le f(x^k, u^k) \le \beta_k$. It remains $f^* = f(x^k, u^k) = \beta_k$, and (x^k, u^k) is optimal for (P).

iii) Assume now the algorithm never terminates. Then from monotonicity $\beta_k \nearrow \overline{\beta} \le f^*$. Let $(\overline{x}, \overline{u}) \in X \times B^0$ be a cluster point of the sequence $\{(x^k, u^k)\}$, and let $x^{k(i)} \to \overline{x}$, $u^{k(i)} \to \overline{u}$, $y_1^{k(i)} \to \overline{y}_1$, $y_2^{k(i)} \to \overline{y}_2$ for some suitable subsequence. Assume that case 1 happens infinitely often for this subsequence. Since $u^{k(i)} \in B_1^{k(i)} \cap B_2^{k(i)}$ under case 1, it follows by a simple continuity argument that $(\overline{x}, \overline{u}) \in G$ and $f(\overline{x}, \overline{u}) - \overline{\beta} \le 0$. But from $(\overline{x}, \overline{u}) \in G$ and $f(\overline{x}, \overline{u}) \le \overline{\beta} \le f^*$ follows $f(\overline{x}, \overline{u}) = \overline{\beta} = f^*$. Hence $(\overline{x}, \overline{u})$ solves (P) and $\beta_k \nearrow f^*$.

Assume now that case 2 happens infinitely often for this subsequence. Then, choosing again a subsequence if necessary, we may assume that either $B^{k(i+1)} \subset B_+^{k(i)}$ for all i (case A) or $B^{k(i+1)} \subset B_-^{k(i)}$ for all i (case B). If case A holds, then $y_1^{k(i+1)} \in B_+^{k(i)}$, hence $l_{k(i)}(y_1^{k(i+1)}) \ge 0$, and therefore from the rules of case 2

$$\begin{aligned} t \cdot \psi_{k(i)}(u^{k(i)}) &\le -l_{k(i)}(y_1^{k(i)}) \le l_{k(i)}(y_1^{k(i+1)}) - l_{k(i)}(y_1^{k(i)}) \\ &\le \|y_1^{k(i+1)} - y_1^{k(i)}\| \to 0. \end{aligned}$$

Likewise, if case B holds, then from $y_2^{k(i+1)} \in B_-^{k(i)}$ we obtain

$$t \cdot \psi_{k(i)}(u^{k(i)}) \le \|y_2^{k(i)} - y_2^{k(i+1)}\| \to 0.$$

Hence it follows that $\psi_{k(i)}(u^{k(i)}) \to 0$. This leads to $g_j(\overline{x}, \overline{u}) \le 0 \quad \forall j \in J_1$ and $f(\overline{x}, \overline{u}) - \overline{\beta} \le 0$, whereas from $u^{k(i)} \in B_2^{k(i)}$ follows $\overline{u} \in Y$ and $g_j(\overline{x}, \overline{u}) \le 0 \quad \forall j \in J_2$. Again we have obtained $(\overline{x}, \overline{u}) \in G$ and $f(\overline{x}, \overline{u}) \le \overline{\beta} \le f^*$, which implies $f(\overline{x}, \overline{u}) = \overline{\beta} = f^*$ and the optimality of $(\overline{x}, \overline{u})$. q.e.d.

We remark that the constraints $g_j(x, u) \le 0$ with $j \in J_2$ are satisfied for all iterates (x^k, u^k), whereas the constraints with $j \in J_1$ are possibly satisfied only in the limit. Thus the partition of J into J_2 and J_1 – besides allowing for mathematical needs – could reflect a distinction between "hard" technological constraints which should be rigorously satisfied even for approximate solutions, and "soft" managerial constraints, which admit a certain tolerance.

5. Bilinear programming

Under the convexity assumptions made for problem (P) all subproblems encountered in the above algorithm are convex, with the possible exception of R(B^q). The solution of R(B^q) is therefore crucial for implementing the algorithm. Here we shall briefly discuss the situation for problem (BL). In this case $R(B^q)$ can be solved by using only convex subprograms and a vertex finding procedure for B^q.

Assume, slightly more general than in (BL), that $f(x,y)$ is convex with regard to x and affine with regard to y, and that the $g_j(x,y)$ are convex jointly in both variables. Let $J_1 := \emptyset$, $J_2 := J$. Problem $R(B^q)$ reads then

$$R(B^q) \qquad \min \{f(x,y_1) | x \in X, y_1 \in B^q, y_2 \in B^q, g_j(x,y_2) \leq 0 \quad \forall j \in J\}.$$

Let

$$G^q := \{(x,y) \in X \times B^q | g_j(x,y) \leq 0 \quad \forall j \in J\},$$

a convex set. Denote by v^i $(i = 1, \cdots, m_q)$ the vertices of B^q. Since $f(x,\cdot)$ is affine one has $\min_{y \in B^q} f(x,y) = \min_i f(x,v^i)$. Hence one obtains

$$\begin{aligned} \beta(B^q) &= \min \{f(x,y_1) | y_1 \in B^q, (x,y_2) \in G^q\} = \min \{\min_{y_1 \in B^q} f(x,y_1) | (x,y_2) \in G^q\} \\ &= \min \{\min_i f(x,v^i) | (x,y_2) \in G^q\} = \min_i (\min \{f(x,v^i) | (x,y_2) \in G^q\}). \end{aligned}$$

Thus if the vertices v^i are known, $R(B^q)$ can be solved by solving the convex programs

$$C(v^i) \qquad \min \{f(x,v^i) | (x,y) \in G^q\}.$$

Since the polyhedron B^q is generated from some predecessor $B^{q'}$ by adding one affine inequality, the vertices of B^q can be calculated from those of $B^{q'}$ by some available methods, among which a method described in [4] seems to be efficient. If $f(x,y)$ is bilinear, the $g_j(x,y)$ are linear and X is a polyhedron, then the programs $C(v^i)$ become linear.

References

[1] Al-Khayyal F.A. and Falk J.E.(1983). Jointly Constrained Biconvex Programming. *Math. Oper. Res.* 8, 273-286.

[2] Altman M.(1968). Bilinear Programming. *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys.* 16, 741-746.

[3] Gallo G. and Ülkücü A.(1977). Bilinear Programming: An Exact Algorithm. *Math. Programming* 12, 173-194.

[4] Horst R., de Vries J., Thoai N.V.(1988). On Finding New Vertices and Redundant Constraints in Cutting Plane Algorithms for Global Optimization. *Oper. Res. Lett.* 7, 85-90.

[5] Kelley J.E.(1960). The Cutting-Plane Method for Solving Convex Programs. *J. Soc. Indust. Appl. Math.* 8, 703-712.

[6] Konno H.(1976). A Cutting Plane Algorithm for Solving Bilinear Programs. *Math. Programming* 11, 14-27.

[7] Sherali H.D. and Shetty C.M.(1980). A Finitely Convergent Algorithm for Bilinear Programming Problems Using Polar Cuts and Disjunctive Face Cuts. *Math. Programming* 19, 14-31.

[8] Thieu T.V.(1988). A Note on the Solution of Bilinear Programming Problems by Reduction to Concave Minimization. *Math. Programming* 41, 249-260.

[9] Vaish H. and Shetty C.M.(1976). The Bilinear Programming Problem. *Naval Res. Logist. Quart.* 23, 303-309.

[10] —— (1977). A Cutting Plane Algorithm for the Bilinear Programming Problem. *Naval Res. Logist. Quart.* 24, 83-94.