Journal of Global Optimization **14:** 267–281, 1999.

An Outer Approximation Algorithm Guaranteeing Feasibility of Solutions and Approximate Accuracy of Optimality

SYUUJI YAMADA[1] and TAMAKI TANAKA[2]
[1]*Department of Electronics and Information Systems, Graduate School of Engineering, Osaka University, Osaka 565-0871, Japan;* [2]*Department of Mathematical System Science, Faculty of Science and Technology, Hirosaki University, Hirosaki 036-8561, Japan*
(e-mail: tanaka@si.hirosaki-u.ac.jp)

(Received 2 April 1997; accepted in revised form 3 May 1998)

Abstract. We treat a concave programming problem with a compact convex feasible set. Assuming the differentiability of the convex functions which define the feasible set, we propose two solution methods. Those methods utilize the convexity of the feasible set and the property of the normal cone to the feasible set at each point over the boundary. Based on the proposed two methods, we propose a solution algorithm. This algorithm takes advantages over classical methods: (1) the obtained approximate solution is always feasible, (2) the error of such approximate value can be evaluated properly for the optimal value of such problem, (3) the algorithm does not have any redundant iterations.

Key words: Concave programming problem, Cutting plane method, Global optimization, Outer approximation method, Supporting hyperplane method

1. Introduction

In optimization theory, while theoretical researchers are interested in the existence of optimal solutions of the problems, and engineering researchers are interested in the solution method. In general, it is difficult to obtain optimal solutions directly. Hence, both of them attempt to design an iterative method to solve the problem, but, unfortunately, the procedure does not always terminate within finite iterations. Therefore, they may compromise their aim to get one of approximate solutions to the problems. Outer approximation methods were contrived to obtain such an approximate solution for global optimization problems. The first approach was the cutting plane method and it was developed in order to solve convex programming problems (see Cheney and Goldstein [2]; Kelley [8]). Another approach called the supporting hyperplane method was proposed by Veinott [13].

The purpose of the paper is to get over three incomplete points which we encounter on implementation of a classical outer approximation algorithm for a concave programming problem. First, an approximate solution obtained by the algorithm is not always feasible. Secondly, we cannot evaluate the error of the obtained solution by the algorithm from the optimal value of the problem. Lastly, the algorithm may have redundant iterations. We denote these incomplete points I, II and III, respectively.

The organization of the paper is as follows: In Section 2, in order to find an optimal solution to minimize a quasi-concave function over a compact convex set defined by differentiable convex functions under the slater condition, we formulate outer approximation algorithms based on two procedures: one is the cutting plane method, the other is the supporting hyperplane method. In Section 3, we explain three incomplete points of the classical algorithms. In Section 4, we get over the incomplete points I–III by improving each algorithm. For the incomplete points I and II, we improve the algorithms by using convexity of functions defining the feasible set. For the incomplete point III, we improve the algorithm based on the supporting hyperplane method by using a property of the normal cone to the feasible set at each points over the boundary. Finally, we combine such improvement to propose one algorithm which gets over the incomplete points I, II and III, simultaneously.

Throughout the paper, we use the followings: int D, bd D and co D denote the interior set of $D \subset R^n$, the boundary set of D and the convex hull of D, respectively. Let for $a, b \in R^n$, $]a, b[= \{x \in R^n : x = a + \delta(b - a),\ 0 < \delta < 1\}$. Given a convex polyhedral set (or polytope) $D \subset R^n$, $V(D)$ denotes the set of all vertices of D. Given a closed set $D \subset R^n$, $E(D)$ denotes the set of all extreme points. Given a nonempty set $D \subset R^n$, D^* denotes the nonpositive polar cone of D. Given a nonempty closed set $D \subset R^n$, $T_D(y)$ and $N_D(y)$ denote the tangent cone to D at $y \in D$ and the normal cone to D at $y \in D$, respectively. Given a convex function $f : R^n \to R$, $\partial f(x)$ denotes the subdifferential of f at x. Given a differentiable function $f : R^n \to R$, $\nabla f(x)$ denotes the gradient of f at $x \in R^n$. For a function $f : R^n \to R$ and a nonempty set $D \subset R^n$, $\min f(D)$ and $\arg\min f(D)$ denote the minimum value of f over D and the set of all solutions minimizing f over D, respectively. The terminology and above definitions in convex analysis can be referred to in [1] and [11].

2. Outer approximation algorithm for the quasi-concave programming problem

Let us consider a quasi-concave programming problem defined as follows:

$$(P)\begin{cases}\text{Minimize} & f(x)\\ \text{subject to} & g_i(x) \le 0, \quad i = 1, \dots, m,\end{cases}$$

where $f : R^n \to R$ is a continuous quasi concave function and $g_i : R^n \to R$, $i = 1, \dots, m$, are differentiable convex functions. We assume that the feasible set

$$D := \{x \in R^n : g_i(x) \le 0,\ i = 1, \dots, m\}$$

is compact, and that

$$\text{int } D = \{x \in R^n : \max_{i=1,\dots,m} g_i(x) < 0\} \neq \emptyset.$$

Let $g(x) := \max_{i=1,\dots,m} g_i(x)$, then $D = \{x \in R^n : g(x) \leq 0\}$ and $g : R^n \to R$ is a convex function. The minimum of problem (P) is always attained in at least one extreme point of D (Thieu et al. [12]). Therefore, $\min f(D) = \min f(E(D))$ and $(\arg\min f(D)) \cap E(D) \neq \emptyset$. If the feasible set D is a polytope, $(\arg\min f(D)) \cap V(D) \neq \emptyset$.

To find a minimum solution of problem (P) whose feasible set D is not a polytope, we now formulate the outer approximation algorithm:

ALGORITHM OAM

Step 0. Generate a polytope S_1 such that $S_1 \supset D$. Set $k \leftarrow 1$.

Step k. (i) Choose $v^k \in V(S_k)$ such that v^k solves the following relaxed problem (Q_k):

$$(Q_k) \begin{cases} \text{Minimize} & f(x) \\ \text{subject to} & x \in S_k. \end{cases}$$

(ii) If $v^k \in D$, then stop; v^k solves problem (P). **(SC1)**

(iii) Otherwise, construct an affine function $h_k : R^n \to R$ satisfying that

$$h_k(v^k) > 0 \tag{1}$$

and that

$$h_k(x) \leq 0 \quad \text{for all } x \in D, \tag{2}$$

and set

$$S_{k+1} := S_k \cap \{x \in R^n : h_k(x) \leq 0\}. \tag{3}$$

(iv) Set $k \leftarrow k + 1$ and go to step k.

Assume that the algorithm OAM generates a sequence $\{v^k\}$ such that $v^k \in V(S_k)$ is a solution for problem (Q_k). Then for any $v^p, v^q \in \{v^k\}$, $p < q$, we have

$$f(v^p) \leq f(v^q). \tag{4}$$

If the algorithm OAM terminates with k_0 iterations $(k_0 > 0)$, v^{k_0} solves problem (P).

THEOREM 2.1. *(See [12].) If the algorithm OAM generates an infinite sequence $\{v^k\}_{k=1}^{\infty}$, then every accumulation point of $\{v^k\}_{k=1}^{\infty}$ belongs to D and it solves problem (P).*

COROLLARY 2.1. *If we replace stopping criterion* **(SC1)** *by the following one: If $g(v^k) \leq \varepsilon$ for a given $\varepsilon > 0$, then stop* **(SC2)***; v^k is an approximate solution of (P), then the algorithm OAM terminates within finite iterations for problem (P).*

REMARK 2.1. Let $\{v^k\}_{k=1}^{\infty}$ be generated by the algorithm OAM and let $\bar{v}$ be an accumulation point of $\{v^k\}_{k=1}^{\infty}$, then by condition (4), we have

$$f(v^1) \leq f(v^2) \leq \cdots \leq f(v^k) \leq \cdots \leq f(\bar{v}) \text{ and } f(v^k) \to f(\bar{v}) \text{ as } k \to \infty.$$

As for ways of constructing an affine function, we have the following two kinds of procedures. One is based on the cutting plane method, the other is based on the supporting hyperplane method. At first, we consider the former procedure. In the case that $v^k \notin D$ at step k in the algorithm OAM, we can construct an affine function $h_k : R^n \to R$ by

$$h_k(x) := \langle \nabla g_{i_k}(v^k), x - v^k \rangle + g(v^k) \tag{5}$$

where $i_k \in L(v^k) := \{i : g_i(v^k) = g(v^k),\ i = 1, \ldots, m\}$. Since $g(v^k) > 0$, $\langle \nabla g_i(v^k), v^k - v^k \rangle + g(v^k) = g(v^k) > 0$ for any $i \in L(y^k)$. Moreover, since that $\nabla g_i(v^k) \in \partial g(v^k)$ for any $i \in L(y^k)$, and that $g(x) \leq 0$ for any $x \in D$, $\langle \nabla g_i(v^k), x - v^k \rangle + g(v^k) = g(x) \leq 0$ for any $x \in D$. Therefore, the way of choosing from elements in $L(y^k)$ is not restrained.

Next, we consider the other procedure, i.e., the supporting hyperplane method. In order to implement the procedure for problem (P), we add the process to choose $\hat{x} \in \operatorname{int} D$ on step 0 of the procedure. Since $\operatorname{int} D = \{x \in R^n : g(x) < 0\} \neq \emptyset$, we can choose $\hat{x} \in \operatorname{int} D$. At step k of the algorithm, if $v^k \notin D$, then we can construct an affine function $h_k : R^n \to R$ by

$$h_k(x) := \langle \nabla g_{i_k}(y^k), x - y^k \rangle \tag{6}$$

where $i_k \in L(y^k)$ and $y^k \in \operatorname{bd} D$ such that $y^k \in]v^k, \hat{x}[$. Indeed, since $\{x \in R^n : h_k(x) = 0\}$ is a supporting hyperplane of D at $y^k \in \operatorname{bd} D$, the affine function $h_k : R^n \to R$ defined by condition (6) satisfies that $h_k(v^k) > 0$ and that $h_k(x) \leq 0$ for any $x \in D$.

As previously indicated, S_{k+1} is defined by constraints from condition (3). A constraint $h_{j_0}(x) \leq 0$ ($j_0 \in \{1, \ldots, k\}$) is said to be redundant for S_{k+1} if the removal of it does not change set S_{k+1}, i.e.,

$$\begin{aligned} S_{k+1} &= S_1 \cap \{x \in R^n : h_j(x) \leq 0,\ j = 1, \ldots, k\} \\ &= S_1 \cap \{x \in R^n : h_j(x) \leq 0,\ j = 1, \ldots, k \text{ and } j \neq j_0\}. \end{aligned}$$

A non redundant constraint for S_{k+1} is called essential for S_{k+1}.

3. Confronting incomplete points

When we use the algorithm OAM replacing stopping criterion **(SC1)** by **(SC2)** for problem (P), we are confronted by the following incomplete points:

I. An approximate solution given by the algorithm is not always contained in the feasible set D of problem (P).

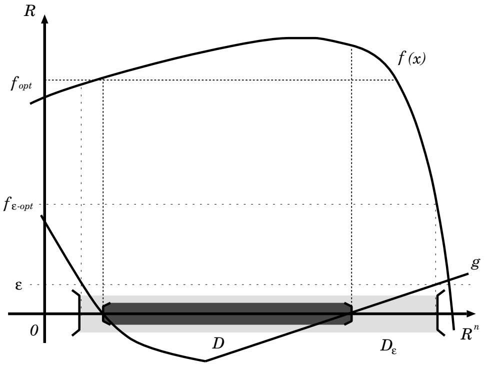

Figure 1. Stopping criterion **(SC2)**; $D_\varepsilon := \{x \in R^n : g(x) < \varepsilon\}$ $(\varepsilon > 0)$.

II. We cannot evaluate the error of the solution obtained by the algorithm for the optimal value of problem (P).

III. At step k_0 of the algorithm, a constructed affine function h_{k_0} may be redundant to the feasible set S_k of the relaxed problem (Q_k) for all $k > k_0 + 1$ (the constraint $h_{k_0}(x) \leq 0$ is essential for S_{k_0+1}, because $S_{k_0+1} = S_{k_0} \cap \{x \in R^n : h_{k_0}(x) \leq 0\}$ and $v^{k_0} \in S_{k_0} \backslash S_{k_0+1}$ where v^{k_0} solves the relaxed problem (Q_{k_0})).

Let us see the the inconvenience caused by each incomplete point.

First, we consider the incomplete point I. When we solve problem (P) by an outer approximation method, our purpose is to get an optimal solution of the problem. However, it often occurs that we do not get an optimal solution of problem (P) within finite iterations. In order to terminate the procedures within finite iterations, we may compromise our aim by getting an approximate solution for problem (P). Since such an approximate solution is possibly infeasible, there are some cases where we cannot be satisfied with the solution.

The reason why the incomplete point II occurs is as follows. Assume that v^k is an approximate solution obtained by the algorithm OAM with stopping criterion **(SC2)** and that f_{opt} denotes the optimal value. Since v^k is an approximate solution minimizes f over $S_k \supset D$, it can be assumed that $f_{\text{opt}} - f(v^k) > 0$. From the approximation viewpoint, we should determine a specific $\delta > 0$ such that $f_{\text{opt}} - f(v^k) < \delta$, but we cannot find such a scalar δ by the algorithm. Because the perturbation set D_ε in Figure 1 depends on a restriction function g as well as a given tolerance $\varepsilon > 0$, and hence the minimum value of f over D_ε, say $f_{\varepsilon-\text{opt}}$ in Figure 1, is not so close to the optimal value f_{opt} even if ε is sufficiently small.

Finally, we see the incomplete point III.

At step k_0 in the algorithm OAM based on the supporting hyperplane method, we assume that v^{k_0} is an optimal solution for the relaxed problem (Q_{k_0}). We consider two cases in construction of an affine function $h_{k_0} : R^n \to R$ separating the point v^{k_0} from the feasible set D. The cases A and B are depicted at the left hand in Figures 2 and 3, respectively. The affine function h_{k_0} in case A may come to be

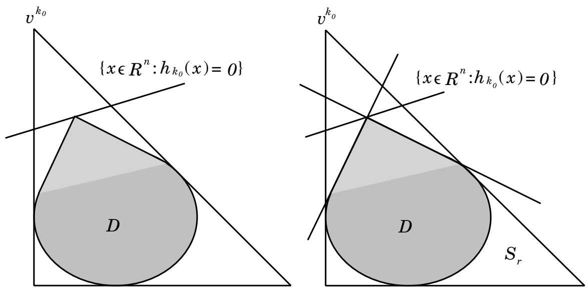

Figure 2. At step r $(r > k_0 + 1)$, $h_{k_0}(x) \leq 0$ is a redundant condition for S_r.

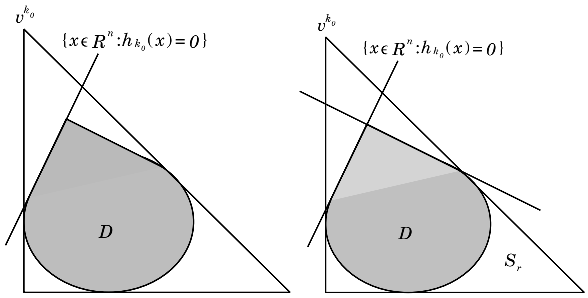

Figure 3. At any step r $(r > k_0 + 1)$, $h_{k_0}(x) \leq 0$ is an essential condition for S_r.

redundant in the relaxed problem at some steps after step k_0 as shown at the right hand in Figure 2. Furthermore, if $h_{k_0}(x) \leq 0$ is redundant for S_r $(r > k_0 + 1)$, then it is not so easy to find $V(S_r)$. To overcome this difficulty, the method of eliminating redundant constraints has been suggested by Horst and Tuy [6], and Thieu et al. [12]. On the other hand, the affine function h_{k_0} in case B is always essential for composing feasible sets of each relaxed problem after step k_0 as shown at the right hand in Figure 3. At each step k in the algorithm, we should construct an essential affine function $h_k : R^n \to R$ for composing feasible sets of each relaxed problem after step k.

4. Improvement of the algorithm

4.1. IMPROVEMENT OF THE ALGORITHM OAM FOR GETTING OVER THE INCOMPLETE POINTS I AND II

In order to getting over the incomplete points I and II, we improve the algorithm OAM by replacing the stopping criterion **(SC1)** by another stopping criterion. We distinguish the following two cases:

Case 1: An infinite sequence is generated by the algorithm for problem (P).

Case 2: A finite sequence is generated by the algorithm for problem (P).

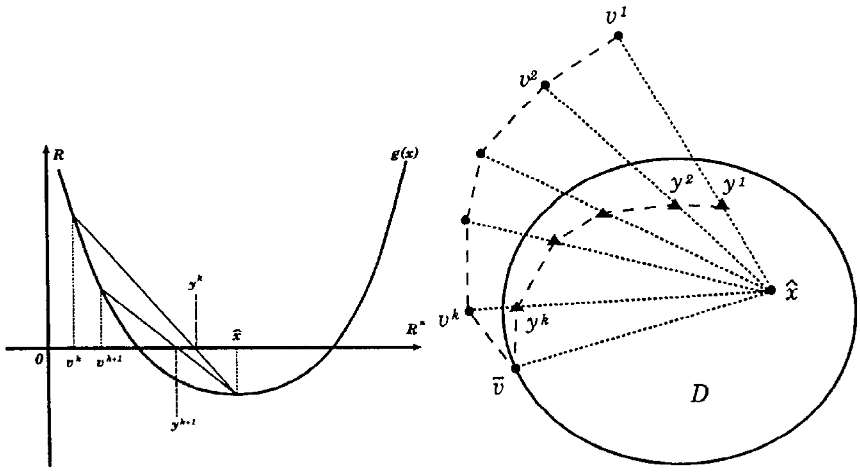

Figure 4. $\{y^k\}_{k=1}^{\infty}$ determined by (7) for $\{v^k\}_{k=1}^{\infty}$.

We do not need to think over such an improvement of the algorithm in Case 2. Because an improvement for Case 1 implies that for Case 2 at the same time.

4.1.1. *On the cutting plane method*

We improve the algorithm OAM based on the cutting plane method in Case 1. Denote by $\{v^k\}_{k=1}^{\infty}$ an infinite sequence generated by the algorithm for problem (P). Assume that $\bar{v}$ is an accumulation point of $\{v^k\}_{k=1}^{\infty}$. Then, from Theorem 2.1, we remember that $\bar{v}$ is contained in the feasible set D of problem (P) and that $\bar{v}$ is an optimal solution for problem (P), i.e., $f(\bar{v})$ is the optimal value for problem (P). Since int $D \neq \emptyset$, we can choose $\hat{x} \in$ int D. Clearly, $g(\hat{x}) < 0$. Let for all $k \in \{1, 2, \ldots\}$,

$$y^k := (1 - \lambda_k)v^k + \lambda_k \hat{x}, \tag{7}$$

where $\lambda_k := g(v^k)/g(v^k) - g(\hat{x})$, then it follows from the following theorem that $\{y^k\}_{k=1}^{\infty}$ belongs to D and that $\{y^k\}_{k=1}^{\infty}$ has a subsequence $\{y^{k_q}\}_{q=1}^{\infty}$ such that $y^{k_q} \to \bar{v}$ as $q \to \infty$.

THEOREM 4.1. *Assume that an infinite sequence $\{v^k\}_{k=1}^{\infty}$ is generated by the algorithm OAM based on the cutting plane method for problem (P) and that $\bar{v}$ is an accumulation point of $\{v^k\}_{k=1}^{\infty}$, then the infinite sequence $\{y^k\}_{k=1}^{\infty}$ defined by condition* (7) *for $\{v^k\}_{k=1}^{\infty}$ satisfies the following conditions:*

(i) $\{y^k\}_{k=1}^{\infty} \subset D$ and

(ii) there is a subsequence $\{y^{k_q}\}_{q=1}^{\infty} \subset \{y^k\}_{k=1}^{\infty}$ such that $y^{k_q} \to \bar{v}$ as $q \to \infty$.

Proof. At first, we prove the statement (i). Let $\lambda_k = g(v^k)/g(v^k) - g(\hat{x})$ for each $k \in \{1, 2, \ldots\}$. Since $g(v^k) > 0$ for all $k \in \{1, 2, \ldots\}$ and $g(\hat{x}) < 0$, we have $g(v^k) - g(\hat{x}) > 0$ for all $k \in \{1, 2, \ldots\}$. Therefore, $0 < \lambda_k < 1$ for all $k \in$

$\{1, 2, \dots\}$. Moreover, since g is a convex function, $g(y^k) = g((1-\lambda_k)v^k + \lambda_k \hat{x}) \leq (1-\lambda_k)g(v^k) + \lambda_k g(\hat{x}) = 0$ for all $k \in \{1, 2, \dots\}$. Consequently, $\{y^k\}_{k=1}^{\infty} \subset D$.

Next, we prove the statement (ii). Since $\bar{v}$ is an accumulation point of $\{v^k\}_{k=1}^{\infty}$, there is a subsequence $\{v^{k_q}\}_{q=1}^{\infty}$ such that $v^{k_q} \to \bar{v}$ as $q \to \infty$. Moreover, since g is continuous, $g(v^{k_q}) \to g(\bar{v}) = 0$ as $q \to \infty$. Therefore, $\lambda_{k_q} \to 0$ as $q \to \infty$. Consequently, $y^{k_q} = (1-\lambda_{k_q})v^{k_q} + \lambda_{k_q}\hat{x} \to \bar{v}$ as $q \to \infty$. □

For the sequence $\{y^k\}_{k=1}^{\infty}$ in Theorem 4.1, we set $M_k := \min\{f(y^i) : i = 1, 2, \dots, k\}$ for all $k \in \{1, 2, \dots\}$. By Theorem 4.1 and the continuity of f, we can verify that

$$M_1 \geq M_2 \geq \cdots \geq M_k \geq \cdots \geq f(\bar{v}) \quad \text{and} \quad M_k \to f(\bar{v}) \text{ as } k \to \infty. \tag{8}$$

According to condition (8) and Remark 2.1,

$$\forall \varepsilon > 0, \quad \exists k_0 \in \{1, 2, \dots\} \quad \text{such that} \quad M_{k_0} - f(v^{k_0}) < \varepsilon. \tag{9}$$

Then, we consider the following stopping criterion:

If $M_k - f(v^k) < \varepsilon$ for a given $\varepsilon > 0$, then stop: $\hat{y} \in \arg\min\{f(y^i) : i = 1, 2, \dots, k\}$ is an approximate solution of problem (P) **(NSC)**.

By condition (9), the algorithm with **(NSC)** terminates after finite iterations. Assume that the algorithm terminates at step s, that is, $M_s - f(v^s) < \varepsilon$ for a given $\varepsilon > 0$. Then, we have an approximate solution $\hat{y}$ for problem (P) by the algorithm with **(NSC)**, it satisfies the following:

$$f(\hat{y}) - f(\bar{v}) \leq f(\hat{y}) - f(v^s) < \varepsilon \quad \text{and} \quad \hat{y} \in D.$$

Consequently, we get over the incomplete points I and II by improving the algorithm OAM based on the cutting plane method by replacing the stopping criterion **(SC1)** by **(NSC)**.

4.1.2. *On the supporting hyperplane method*

In this section, we improve the algorithm OAM based on the supporting hyperplane method in Case 1. Denote by $\{v^k\}_{k=1}^{\infty}$ an infinite sequence generated by the algorithm for problem (P). Then, we remember that the infinite sequence $\{y^k\}_{k=1}^{\infty} \subset D$ satisfying $y^k \in]v^k, \hat{x}[\cap \text{bd}\, D$ $(k = 1, 2, \dots)$ are generated by the algorithm for $\{v^k\}_{k=1}^{\infty}$ (see condition (6)). Assume that $\bar{v}$ is an accumulation point of $\{v^k\}_{k=1}^{\infty}$. Then, it follows from the following theorem that $\{y^k\}_{k=1}^{\infty}$ has a subsequence $\{y^{k_q}\}_{q=1}^{\infty}$ such that $y^{k_q} \to \bar{v}$ as $q \to \infty$.

THEOREM 4.2. *Assume that $\hat{x}$ is an interior point of the feasible set D of problem (P) and that infinite sequences $\{v^k\}_{k=1}^{\infty}$ and $\{y^k\}_{k=1}^{\infty}$ are generated based on the supporting hyperplane method for the problem where v^k is an optimal solution of the relaxed problem (Q_k) for all $k \in \{1, 2, \dots\}$ and $y^k \in]v^k, \hat{x}[$ for all $k \in \{1, 2, \dots\}$. Let $\bar{v}$ be an accumulation point of $\{v^k\}_{k=1}^{\infty}$. Then $\{y^k\}_{k=1}^{\infty}$ has a subsequence $\{y^{k_q}\}_{q=1}^{\infty} \subset \{y^k\}_{k=1}^{\infty}$ such that $y^{k_q} \to \bar{v}$ as $q \to \infty$.*

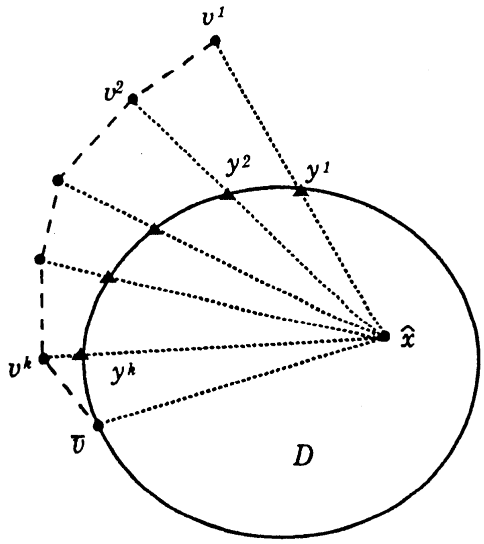

Figure 5. $\{y^k\}_{k=1}^{\infty}$ generated by the algorithm OAM based on the supporting hyperplane method for problem (P).

Proof. Since $\bar{v}$ is an accumulation point of $\{v^k\}_{k=1}^{\infty}$, $\{v^k\}_{k=1}^{\infty}$ has a subsequence $\{v^{k_q}\}_{q=1}^{\infty}$ such that $v^{k_q} \to \bar{v}$ as $q \to \infty$. Let $\mu_{k_q} := g(v^{k_q})/g(v^{k_q}) - g(\hat{x})$ for all $q \in \{1, 2, \dots\}$. Since $g(v^{k_q}) > 0$ for all $q \in \{1, 2, \dots\}$ and $g(\hat{x}) < 0$, we have $0 < \mu_{k_q} < 1$ for all $q \in \{1, 2, \dots\}$. Moreover, since $g(v^{k_q}) \to g(\bar{v}) = 0$ as $q \to \infty$, we get that $\mu_{k_q} \to 0$ as $q \to \infty$.

For $\{v^{k_q}\}_{q=1}^{\infty}$, we remember that a subsequence $\{y^{k_q}\}_{q=1}^{\infty} \subset \{y^k\}_{k=1}^{\infty}$ satisfying $y^{k_q} \in]v^{k_q}, \hat{x}[\cap \mathrm{bd}\ D$ for all $q \in \{1, 2, \dots\}$ is generated by the algorithm (see condition (6)). Therefore, there are $\lambda_{k_q} \in]0, 1[$, for $q = 1, 2, \dots$, such that $y^{k_q} = (1 - \lambda_{k_q})v^{k_q} + \lambda_{k_q}\hat{x}$. Since $g(y^{k_q}) = 0$ for all $q \in \{1, 2, \dots\}$ and g is convex, we have

$$\begin{aligned} 0 &= g(y^{k_q}) = g((1-\lambda_{k_q})v^{k_q} + \lambda_{k_q}\hat{x}) \\ &\le (1-\lambda_{k_q})g(v^{k_q}) + \lambda_{k_q}g(\hat{x}) \\ &= (1-\mu_{k_q})g(v^{k_q}) + \mu_{k_q}g(\hat{x}) + (\mu_{k_q} - \lambda_{k_q})(g(v^{k_q}) - g(\hat{x})) \end{aligned}$$

for all $q \in \{1, 2, \dots\}$. Moreover, since $(1 - \mu_{k_q})g(v^{k_q}) + \mu_{k_q}g(\hat{x}) = 0$ for all $q \in \{1, 2, \dots\}$, we get $0 \le (\mu_{k_q} - \lambda_{k_q})(g(v^{k_q}) - g(\hat{x}))$ for all $q \in \{1, 2, \dots\}$. Since $g(v^{k_q}) - g(\hat{x}) > 0$ for all $q \in \{1, 2, \dots\}$, we get that $\lambda_{k_q} \le \mu_{k_q}$ for all $q \in \{1, 2, \dots\}$.

Consequently, we get that $\lambda_{k_q} \to 0$ as $q \to \infty$. Thus, by (6), $y^{k_q} \to \bar{v}$ as $q \to \infty$. This completes the proof. □

From the result of Theorem 4.2, we get over the incomplete points I, II by improving the algorithm by using the stopping criterion **(NSC)**.

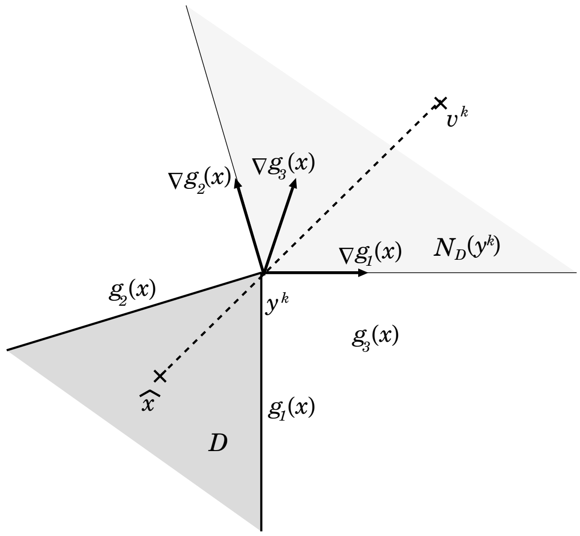

Figure 6. $N_D(y^k)$ is the normal cone to D at y^k.

4.2. IMPROVEMENT OF THE ALGORITHM OAM FOR GETTING OVER THE INCOMPLETE POINT III

The algorithm OAM based on the supporting hyperplane method for problem (P) does not have the incomplete point III, if the restriction of the feasible set is a strict convex function (Horst and Tuy [6]). In this section, for problem (P) whose feasible set D is defined by a not-strict restriction, we improve the algorithm OAM for getting over the incomplete point III. We suggest another way of constructing an affine function $h_k : R^n \to R$ in the case of $v^k \notin D$ at step k, of the algorithm OAM for problem (P). At first, we consider an affine function

$$h_k(x) := \langle a^k, x - y^k \rangle \tag{10}$$

where $y^k \in]v^k, \hat{x}[\cap \mathrm{bd}\ D$ for a given $\hat{x} \in \mathrm{int}\ D$ and $a^k \in \partial g(y^k)$ is an extreme direction of $N_D(y^k)$.

REMARK 4.1. Let $D \subset R^n$ be a compact, convex set and $D := \{x \in R^n : g_i(x) \leq 0,\ i = 1, \ldots, m\}$ where $g_i : R^n \to R, i = 1, 2, \ldots, m$, are differentiable convex functions. Then, for all $y \in D$,

$$\partial g(y) = \left\{ x \in R^n : x = \sum_{i \in L(y)} \lambda_i \nabla g_i(y),\ \sum_{i \in L(y)} \lambda_i = 1,\ \lambda_i \geq 0 \text{ for all } i \in L(y) \right\}.$$

LEMMA 4.1. *Let $D \subset R^n$ be a compact, convex set and $D := \{x \in R^n : g_i(x) \leq 0,\ i = 1, \ldots, m\}$ where $g_i : R^n \to R$, $i = 1, \ldots, m$, are differentiable convex functions and let $N_D(y)$ be the normal cone to D at $y \in D$. If* int $D = \{x \in R^n :$

$g_i(x) < 0,\ i = 1, \ldots, m\} \neq \emptyset$, *then for all* $y \in \text{bd } D$,

$$N_D(y) = \left\{ x \in R^n : x = \sum_{i \in L(y)} \lambda_i \nabla g_i(y), \lambda_i \geq 0 \textit{ for all } i \in L(y) \right\}.$$

Proof. Let $T_D(y)$ be the tangent cone to D at $y \in D$. Since int $D = \{x \in R^n : g_i(x) < 0, i = 1, \ldots, m\} \neq \emptyset$, we have for all $y \in \text{bd } D$,

$$T_D(y) = \{x \in R^n : \langle \nabla g_i(y), x - y \rangle \leq 0, i \in L(y)\}.$$

Therefore, for all $y \in \text{bd } D$,

$$N_D(y) = T_D(y)^* = \left\{ x \in R^n : x = \sum_{i \in L(y)} \lambda_i \nabla g_i(y), \lambda_i \geq 0 \text{ for all } i \in L(y) \right\}.$$

Then, by the following theorem that we can get over the incomplete point III.

THEOREM 4.3. *Assume that* $v^{k_0} \notin D$ *is an optimal solution of the relaxed problem* (Q_{k_0}) *at step* k_0 *in the algorithm OAM based on the way of constructing an affine function defined by condition (10) for problem* (P)*. Then, a constraint* $h_{k_0}(x) \leq 0$ *is essential for* S_k $(k > k_0)$*, i.e.,*

$$\begin{aligned} S_k &= S_1 \cap \{x \in R^n : h_j(x) \leq 0,\ j = 1, \ldots, k-1\} \\ &\neq S_1 \cap \{x \in R^n : h_j(x) \leq 0,\ j = 1, \ldots, k-1 \text{ and } j \neq k_0\}. \end{aligned}$$

Proof. We consider a closed half space $X := \{x \in R^n : h_{k_0}(x) \leq 0\}$. For each $k > k_0$, let

$$H_k^{'} := \{x \in R^n : h_j(x) \leq 0,\ j = 1, \ldots, k-1 \text{ and } j \neq k_0\}.$$

We shall show that for each $k > k_0$, $H_k^{'} \not\subset X$. Suppose to the contrary that $h_{k_0}(x) \leq 0$ for all $x \in H_k^{'}$. For $y^{k_0} \in]v^{k_0}, \hat{x}[\cap \text{bd } D$ for some $\hat{x} \in \text{int } D$, we have $y^{k_0} \in \text{bd } H_k^{'}$ since $D \subset H_k^{'} \subset X$ and $y^{k_0} \in \text{bd } D \cap \text{bd } X$. Thus, there is $p \in \{1, \ldots, k-1\} \backslash \{k_0\}$ such that $h_p(y^{k_0}) = 0$. Let $P_k^{'}$ be the set of all indices satisfying that $h_p(y^{k_0}) = 0$. Since H_{k_0} is a closed convex cone and the set $Y := \{x \in R^n : h_p(x) \leq 0, p \in P_k^{'}\}$ is a convex cone, we have $Y \subset X$. Moreover, for all $p \in P_k^{'}$, and $x \in R^n$,

$$\begin{aligned} h_p(x) &= \langle a^p, x - y^p \rangle = \langle a^p, x - y^{k_0} \rangle + \langle a^p, y^{k_0} - y^p \rangle \\ &= \langle a^p, x - y^{k_0} \rangle + h_p(y^{k_0}) = \langle a^p, x - y^{k_0} \rangle, \end{aligned} \tag{11}$$

and so $Y = \{x \in R^n : \langle a^p, x - y^{k_0} \rangle \leq 0, p \in P_k^{'}\}$ and by condition (10), $X = \{x \in R^n : \langle a^{k_0}, x - y^{k_0} \rangle \leq 0\}$. We note that X and Y are convex polyhedral cones, and then

$$\begin{aligned} (X - y^{k_0})^* &= \{z \in R^n : z = \lambda a^{k_0},\ \lambda \geq 0\}, \\ (Y - y^{k_0})^* &= \{z \in R^n : z = \textstyle\sum_{p \in P_k^{'}} \lambda_p a^p,\ \lambda_p \geq 0,\ p \in P_k^{'}\}. \end{aligned}$$

Since $Y - y^{k_0} \subset X - y^{k_0}$, we have $(Y - \{y^{k_0}\})^* \supset (X - \{y^{k_0}\})^*$. Consequently, we get that there are $\lambda_p \geq 0$, $p \in P_k^{'}$, satisfying that

$$\sum_{p \in P_k^{'}} \lambda_p a^p = a^{k_0}. \tag{12}$$

Since $a^p \in \partial g(y^p)$ for all $p \in P_k^{'}$ and $y^p \in \text{bd } D$,

$$g(x) \geq \langle a^p, x - y^p \rangle + g(y^p) = \langle a^p, x - y^p \rangle, \quad \text{for all } x \in R^n. \tag{13}$$

From conditions (11) and (13), it follows that for all $x \in R^n$, $g(x) \geq \langle a^p, x - y^{k_0} \rangle$, $p \in P_k^{'}$. Consequently, we get that

$$a^p \in \partial g(y^{k_0}) \subset U(y^{k_0}), \quad \text{for all } p \in P_k^{'}. \tag{14}$$

The optimality of v^{k_0} for the relaxed problem (Q_{k_0}) implies that $y^{k_0} \notin \text{bd} S_j$ for all $j \leq k_0$, i.e., $P_k^{'} \cap \{1, \ldots, k_0 - 1\} = \emptyset$. Since $h_{k_0}(\hat{x}) < 0$ and $h_{k_0}(v^p) \leq 0$ for all $p \in P_k^{'}$, we have

$$h_{k_0}(y^p) = \langle a^{k_0}, y^p - y^{k_0} \rangle = (1 - \mu_p) \langle a^{k_0}, v^p - y^{k_0} \rangle + \mu_p \langle a^{k_0}, \hat{x} - y^{k_0} \rangle < 0, \tag{15}$$

where $\mu_p \in]0, 1[$, $p \in P_k^{'}$, satisfying $y^p = (1 - \mu_p) v^p + \mu_p \hat{x}$. Moreover, $h_p(y^{k_0}) = \langle a^p, y^{k_0} - y^p \rangle = 0$ for all $p \in P_k^{'}$. Consequently, we get that

$$a^{k_0} \neq \lambda_p a^p, \quad \text{for all } p \in P_k^{'}, \quad \lambda_p \geq 0. \tag{16}$$

By conditions (12), (14) and (16), there are $x^1, x^2 \in U(y^{k_0})$, $\alpha_1, \alpha_2 > 0$ such that $x^1 \neq a^{k_0}$, $x^2 \neq a^{k_0}$ and $a^{k_0} = \alpha_1 x^1 + \alpha_2 x^2$. By condition (10), a^{k_0} is an extreme direction of $N_D(y^{k_0})$. This is a contradiction, and hence there is $z \in H_k^{'}$ such that $h_{k_0}(z) > 0$, i.e., $H_k^{'} \not\subset X$. Since $y^{k_0} \notin \text{bd } S_1$, we get that $S_k \neq S_1 \cap H_k^{'}$. This completes the proof. $\square$

From the result of Theorem 4.3, we can get over the incomplete point III by improving the algorithm by using condition (10). However, in order to implement the algorithm OAM, the following question must be examined:

(A) *How do we find an extreme direction of $N_D(y^k)$ in each step ?*

We consider the case that $v^k \notin D$ at step k in the algorithm OAM. If $|L(y^k)| \leq 2$, then for any $i \in L(y^k)$, $\nabla g_i(y^k)$ is an extreme direction of $N_D(y^k)$. Otherwise, by denoting that $L(y^k) = \{i_1, \ldots, i_{l_k}\}$ $(l_k \geq 3)$, we consider the following problem:

$$(E_k) \begin{cases} \text{Minimize} & \langle v^k - y^k, c^j \rangle \\ \text{subject to} & c^j = \frac{1}{\|\nabla g_{i_j}(y^k)\|} \nabla g_{i_j}(y^k), \quad j = 1, \ldots, l_k. \end{cases}$$

Note that at each step k in the algorithm, $l_k \leq m$ (we note that m is the number of the differentiable convex functions which define the feasible set D of problem (P)). Thus, at each step k in the algorithm, the number of the feasible solutions of problem (E_k) is finite.

THEOREM 4.4. *If $\bar{c}$ is an optimal solution of (E_k), then $\bar{c}$ is an extreme direction of $N_D(y^k)$.*

Proof. We shall show that $\bar{c} \neq \lambda_1 c^{j_1} + \lambda_2 c^{j_2}$ for any $c^{j_1}, c^{j_2} \in \{c^j : j = 1, \ldots, l_k\}$ and $\lambda_1, \lambda_2 > 0$. Indeed, suppose to the contrary that there exist $j_1, j_2 \in \{1, \ldots, l_k\}$, $\lambda_1, \lambda_2 > 0$ such that $\bar{c} = \lambda_1 c^{j_1} + \lambda_2 c^{j_2}$, $\bar{c} \neq c^{j_1}$, $\bar{c} \neq c^{j_2}$, $c^{j_1} \neq c^{j_2}$. Since $\langle c^{j_1}, c^{j_2} \rangle < 1$,

$$\begin{aligned} 1 = \langle \bar{c}, \bar{c} \rangle &= \langle \lambda_1 c^{j_1} + \lambda_2 c^{j_2}, \lambda_1 c^{j_1} + \lambda_2 c^{j_2} \rangle \\ &= {\lambda_1}^2 \langle c^{j_1}, c^{j_1} \rangle + {\lambda_2}^2 \langle c^{j_2}, c^{j_2} \rangle + 2\lambda_1\lambda_2 \langle c^{j_1}, c^{j_2} \rangle \\ &< {\lambda_1}^2 + {\lambda_2}^2 + 2\lambda_1\lambda_2 \\ &= (\lambda_1 + \lambda_2)^2. \end{aligned}$$

Hence, we have

$$\begin{aligned} \langle v^k - y^k, \bar{c} \rangle &= \lambda_1 \langle v^k - y^k, c^{j_1} \rangle + \lambda_2 \langle v^k - y^k, c^{j_2} \rangle \\ &\geq (\lambda_1 + \lambda_2) \min\{\langle v^k - y^k, c^{j_1} \rangle, \langle v^k - y^k, c^{j_2} \rangle\} \\ &> \min\{\langle v^k - y^k, c^{j_1} \rangle, \langle v^k - y^k, c^{j_2} \rangle\}. \end{aligned}$$

This is a contradiction to the optimality of $\bar{c}$ for problem (E_k). This completes the proof. □

From the result of Theorem 4.4, it follows that $\bar{c}$ is an extreme direction of $N_D(y^k)$ provided that $\bar{c}$ solves problem (E_k). In order to get over the incomplete point III and settle question (A), in the case of the algorithm OAM for problem (P), we construct an affine function $h_k : R^n \to R$ by

$$h_k(x) := \langle \bar{c}, x - y^k \rangle \tag{17}$$

where $y^k \in]v^k, \hat{x}[\cap \mathrm{bd}\, D$ for some $\hat{x} \in \mathrm{int}\, D$ and $\bar{c}$ is a minimizing point of $\langle v^k - y^k, c \rangle$ over $\{c \in R^n : c = \nabla g_i(y^k)/\|\nabla g_i(y^k)\|,\ i \in L(y^k)\}$. Clearly, $h_k : R^n \to R$ satisfies that $h_k(v^k) > 0$ and that $h_k(x) \leq 0$ for any $x \in D$. Hence, the affine function $h_k(x)$ defined by condition (17) satisfies conditions (1) and (2). Consequently, we can implement the proposed algorithm OAM by using such function $h_k(x)$.

5. Conclusions

In this paper, we have presented two kinds of algorithms of an outer approximation method for a quasi-concave programming problem. One of them generates an infinite sequence which is contained in the feasible set of the problem and whose accumulation points are optimal solutions for the problem, or it generates a finite sequence which is contained in the feasible set and whose terminal point is an optimal solution for the problem. Implementing the algorithm we can get an approximation value with its error for the optimal value less than a given positive constant (tolerance), and the approximate solution is always contained in the feasible set.

At each step, the other algorithm generates an essential inequality which is needed to produce a feasible set of each relaxed problems for the original problem whose feasible set is defined by a finite number differentiable convex functions. Therefore, the algorithm does not have any redundant iterations. Consequently, the algorithm does not need to have an additional procedure of identifying redundant constraints.

Finally, by combining such two algorithms with **(NSC)** and by constructing an affine function defined by condition (17) in each step at the same time, we can propose one algorithm getting over the incomplete points I, II and III for the algorithm OAM for problem (P) simultaneously.

Acknowledgments

The authors are very grateful to Professor Panos M. Pardalos for his encouragement to this paper, and to the referees for their valuable comments and suggestions. The authors would like to thank Professors T. Tanino and M. Inuiguchi of Graduate School of Engineering of Osaka University for their useful suggestions and helps on this research.

References

1. Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993), *Nonlinear Programming: Theory and Algorithms*, 2nd ed., John Wiley, New York.
2. Cheney, E.W. and Goldstein, A.A. (1959), Newton's Method of Convex Programming and Tchebycheff Approximation, *Numer. Math.* 1, 253–268.
3. Falk, J.E. and Hoffman, K.R. (1976), A Successive Underestimation Method for Concave Minimization Problems, *Mathematics of Operations Research* 1, 251–259.
4. Horst, R., Thiau, N.V. and De Vries, J. (1988), On Finding New Vertices and Redundant Constraints in Cutting Plane Algorithms for Global Optimization, *Operations Research Letters* 7, 85–90.
5. Horst, R. and Tuy, H. (1987), On the Convergence of Global Methods in Multiextremal Optimization, *Journal of Optimization Theory and Applications* 54, 253–271.
6. Horst, R. and Tuy, H. (1990), *Global Optimization*, Springer-Verlag, Berlin.
7. Horst, R. and Pardalos, P.M. (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht.
8. Kelley, J.E., Jr. (1960), The Cutting-Plane Method for Solving Convex Programs, *SIAM Journal* 8, 703–712.
9. Kuno, T., Yajima, Y. and Konno, H. (1993), An Outer Approximation Method for Minimizing the Product of Several Convex Functions on a Convex Set, *Journal of Global Optimization* 3, 325–335.

10. Nemhauser, G.L., Kan, A.H.G.R. and Todd, M.J. (1989), *Optimization; Handbooks in Operations Research and Management Science, vol.1,* Elsevier Science Publishers, B.V.
11. Rockafellar, R.T. (1970), *Convex Analysis,* Princeton University Press, Princeton, N.J.
12. Thieu, T.V., Tam, B.T. and Ban, V.T. (1983), An Outer Approximation Method for Globally Minimizing a Concave Function over a Compact Convex Set, *Acta Mathematica Vietnamica* 8, 21–40.
13. Veinott, A.F., Jr. (1967), The Supporting Hyperplane Method for Unimodal Programming, *Operations Research* 15, 147–152.