A Conical Branch-and-Bound Algorithm for a Class of Reverse Convex Programs

Hidetoshi Nagai Takahito Kuno*

Abstract

The purpose of this paper is to construct a conical branch-and-bound algorithm for solving linear programming problems with an additional reverse convex constraint. We propose an inexpensive bound-tightening procedure, which is based on the surrogate relaxation. We show that this procedure considerably tightens lower bounds yielded by the usual linear programming relaxation. We also report numerical results, which indicate that the proposed algorithm is much promising, compared with existing ones.

1 Introduction

Let us consider a class of reverse convex programs, i.e., linear programs with an additional reverse convex constraint (LPARC). The feasible set of this class is a difference of a polyhedron and an open convex set. We need to optimize a linear function on such a nonconvex set, which might be disconnected. Therefore, LPARC can have multiple locally optimal solutions, many of which fail to be globally optimal. Although LPARC is just a subclass of the reverse convex program, it involves a wide variety of problems (see e.g., [2]). Among others, of importance is the linear complementarity problem: find $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathbf{x} \geq \mathbf{0}, \quad \mathbf{M}\mathbf{x} + \mathbf{q} \geq \mathbf{0}, \quad \mathbf{x}^{\mathsf{T}}(\mathbf{M}\mathbf{x} + \mathbf{q}) = 0,$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\mathbf{q} \in \mathbb{R}^n$. Even this well-known problem is an instance of LPARC:

$$\left|\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & \mathbf{M}\mathbf{x} - \mathbf{y} = -\mathbf{q}, \quad (\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \\ & z - \displaystyle\sum_{j=1}^{n} \min\{x_j, y_j\} \geq 0. \end{array}\right.$$

*The author was partially supported by the Grand-in-Aid for Scientific Research (C) 17560050 from the Japan Society for the Promotion of Science.

2000 Mathematics Subject Classification. Primary 90C26, Secondary 90C57.

Key words and phrases. reverse convex program, global optimization, nonconvex optimization, conical branch-and-bound algorithm, surrogate relaxation.

To solve LPARC, various algorithms have been proposed since the pioneer work by Hillestad [1]. In this paper, we focus on the conical branch-and-bound algorithm, which was originally proposed by Tuy [9] for concave minimization problems and applied to LPARC later in, e.g., [6, 7]. In the bounding process, we usually relax each subproblem into a linear program and solve it to obtain a lower bound on the optimal value. We will show that the lower bound yielded by this linear programming relaxation can be tighten considerably using a nonlinear surrogate relaxation. Recently, it was reported in [4, 5] that a similar procedure works well in simplicial branch-and-bound algorithms for concave minimization problems. After giving our problem settings of LPARC in Section 2, we explain basic workings of the standard conical branch-and-bound algorithm in Section 3. We then describe the nonlinear surrogate relaxation and incorporate it into the branch-and-bound algorithm in Section 4. Section 5 is devoted to a report of numerical results on the proposed algorithm.

2 Problem settings

The problem we consider in this paper is the following LPARC:

$$\left|\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x}+\mathbf{D}\mathbf{y}=\mathbf{b}, \quad (\mathbf{x},\mathbf{y})\geq\mathbf{0} \\ & g(\mathbf{x})\geq 0, \end{array}\right. \tag{2.1}$$

where $\mathbf{A}\in\mathbb{R}^{m\times r}$, $\mathbf{D}\in\mathbb{R}^{m\times(n-r)}$, $\mathbf{b}\in\mathbb{R}^m$, $\mathbf{c}\in\mathbb{R}^r$, and $g:\mathbb{R}^r\to\mathbb{R}$ is a convex function. In many applications, we can assume that $r\geq 0$ is much smaller than n because of the low-rank nonconvexity [3]. Low-rank-nonconvex structured instances of LPARC are generally formulated into

$$\left|\begin{array}{lll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x}+\mathbf{d}^{\mathsf{T}}\mathbf{y} & \\ \text{subject to} & \mathbf{A}\mathbf{x}+\mathbf{D}\mathbf{y}=\mathbf{b}, & (\mathbf{x},\mathbf{y})\geq\mathbf{0} \\ & g(\mathbf{x})+\mathbf{h}^{\mathsf{T}}\mathbf{y}\geq 0, & \end{array}\right. \tag{2.2}$$

where $\mathbf{d}$, $\mathbf{h}\in\mathbb{R}^{n-r}$ and $n>>r$. If we introduce auxiliary variables ζ_-, ζ_+, η_- and η_+, then (2.2) reduces to the form of (2.1):

$$\left|\begin{array}{lll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x}+\zeta_+-\zeta_- & \\ \text{subject to} & \mathbf{A}\mathbf{x}+\mathbf{D}\mathbf{y}=\mathbf{b}, & (\mathbf{x},\mathbf{y})\geq\mathbf{0} \\ & \zeta_+-\zeta_--\mathbf{d}^{\mathsf{T}}\mathbf{y}=0, & (\zeta_-,\zeta_+)\geq\mathbf{0} \\ & \eta_+-\eta_--\mathbf{h}^{\mathsf{T}}\mathbf{y}=0, & (\eta_-,\eta_+)\geq\mathbf{0} \\ & g(\mathbf{x})+\eta_+-\eta_-\geq 0. & \end{array}\right.$$

Let

$$\begin{array}{rcl} F & = & \{\mathbf{x}\in\mathbb{R}^r \mid \exists\mathbf{y}\geq\mathbf{0},\ \mathbf{A}\mathbf{x}+\mathbf{D}\mathbf{y}=\mathbf{b},\ \mathbf{x}\geq\mathbf{0}\} \\ G & = & \{\mathbf{x}\in\mathbb{R}^r \mid g(\mathbf{x})<0\}, \end{array}$$

and assume that both F and G are bounded and have interior points. Then (2.1) is embedded in the $\mathbf{x}$-space as $\min\{\mathbf{c}^\mathsf{T}\mathbf{x} \mid \mathbf{x} \in F \setminus G\}$. We assume that at least one optimal solution $\mathbf{x}^\circ$ to the associated linear program $\min\{\mathbf{c}^\mathsf{T}\mathbf{x} \mid \mathbf{x} \in F\}$ is a point in G. This condition makes (2.1) nontrivial, but provides us with a valuable information about its optimality [2, 10]:

Proposition 2.1. *If $F \setminus G \neq \emptyset$, there exists a globally optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$ to (2.1) such that $\mathbf{x}^*$ is located at the intersection of an edge of the polyhedron F with the boundary of the set G.*

For simplicity, we assume $\mathbf{x}^\circ = \mathbf{0}$ in the sequel.

3 Overview of the conical algorithm

In this section, we will overview the basic workings of the standard conical branch-and-bound algorithm on (2.1) [2, 10].

Let $\Delta^1 = \{\mathbf{x} \in \mathbb{R}^r \mid \mathbf{x} \geq \mathbf{0}\}$. Then Δ^1 is a cone vertexed at $\mathbf{x}^\circ = \mathbf{0}$ and includes the polytope F. Starting from this cone Δ^1, we recursively divide it into subcones, each vertexed at $\mathbf{x}^\circ$, satisfying

$$\Delta^k = \Delta^{2k} \cup \Delta^{2k+1}, \quad \mathrm{int}(\Delta^{2k}) \cap \mathrm{int}(\Delta^{2k+1}) = \emptyset, \quad k = 1, 2, \dots,$$

where $\mathrm{int}(\,\cdot\,)$ denotes the interior. This procedure generates an infinite sequence of cones $\{\Delta^{k_\ell} \mid \Delta^{k_\ell} \supset \Delta^{k_{\ell+1}},\ \ell = 1, 2, \dots\}$. To guarantee the convergence of the algorithm, we need to subdivide Δ^1 in such an exhaustive manner that $\bigcap_{\ell=1}^{\infty} \Delta^{k_\ell}$ becomes a half line emanating from $\mathbf{x}^\circ$. Suppose that Δ^k is spanned by r linearly independent vectors $\mathbf{w}_i \in \mathbb{R}^r$, $i = 1, \dots, r$, and denote $\Delta^k = \mathrm{cone}(\{\mathbf{w}_1, \dots, \mathbf{w}_r\})$. The easiest exhaustive subdivision rule is *bisection*, i.e., we may divide the longest edge of $\Delta^k = \mathrm{conv}(\{\mathbf{w}_1, \dots, \mathbf{w}_r\})$, say $\mathbf{w}_p$-$\mathbf{w}_q$, at a fixed ratio of $\alpha \in (0, 1/2]$, where $\mathrm{conv}(\,\cdot\,)$ denotes the convex hull. Letting $\mathbf{w} = (1-\alpha)\mathbf{w}_p + \alpha\mathbf{w}_q$, then we have

$$\Delta^{2k} = \mathrm{cone}(\{\mathbf{w}_i \mid i \neq p\} \cup \{\mathbf{w}\}), \quad \Delta^{2k+1} = \mathrm{cone}(\{\mathbf{w}_i \mid i \neq q\} \cup \{\mathbf{w}\}).$$

For each subcone $\Delta = \Delta^k$, we have a subproblem of (2.1):

$$\mathrm{P}(\Delta) \left| \begin{array}{ll} \text{minimize} & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} & \mathbf{x} \in (F \setminus G) \cap \Delta. \end{array} \right.$$

This problem is essentially the same as (2.1) and cannot be solved directly. We instead compute a lower bound on the optimal value of $\mathrm{P}(\Delta)$. If the bound is greater than or equal to the value of the best feasible solution $\mathbf{x}^*$ to (2.1) obtained so far, we can discard $\mathrm{P}(\Delta)$ from further consideration. For each $i = 1, \dots, r$, let β_i be a positive number such that $g(\beta_i\mathbf{w}_i) = 0$, and let

$$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r], \quad \mathbf{v}_i = \beta_i\mathbf{w}_i.$$

Then we have $\Delta = \{\mathbf{x} \in \mathbb{R}^r \mid \mathbf{x} = \mathbf{V}\boldsymbol{\lambda},\ \boldsymbol{\lambda} \geq \mathbf{0}\}$. We also see from the convexity of g that

$$\Delta \setminus G \subset \{\mathbf{x} \in \mathbb{R}^r \mid \mathbf{x} = \mathbf{V}\boldsymbol{\lambda},\ \mathbf{e}^{\mathsf{T}}\boldsymbol{\lambda} \geq 1,\ \boldsymbol{\lambda} \geq \mathbf{0}\},$$

where $\mathbf{e}$ is an all-ones vector. This implies that a lower bound of $\mathrm{P}(\Delta)$ is given as the optimal value of a linear program:

$$\overline{\mathrm{P}}(\Delta) \left| \begin{array}{lll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x} & \\ \text{subject to} & \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b}, & \mathbf{y} \geq \mathbf{0} \\ & \mathbf{x} - \mathbf{V}\boldsymbol{\lambda} = \mathbf{0}, & \boldsymbol{\lambda} \geq \mathbf{0} \\ & \mathbf{e}^{\mathsf{T}}\boldsymbol{\lambda} \geq 1, & \end{array} \right.$$

which is known as the *linear programming relaxation* of $\mathrm{P}(\Delta)$. Let $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ denote an optimal solution to $\overline{\mathrm{P}}(\Delta)$, and let $z(\overline{\mathrm{P}}) = \mathbf{c}^{\mathsf{T}}\overline{\mathbf{x}}$.

4 Surrogate relaxation and the proposed algorithm

To tighten the lower bound $z(\overline{\mathrm{P}})$, we propose here a kind of surrogate relaxation of $\mathrm{P}(\Delta)$.

Let us consider the dual problem of $\overline{\mathrm{P}}(\Delta)$:

$$\left| \begin{array}{lll} \text{maximize} & \mathbf{b}^{\mathsf{T}}\boldsymbol{\pi} + \eta & \\ \text{subject to} & \mathbf{A}^{\mathsf{T}}\boldsymbol{\pi} + \boldsymbol{\rho} = \mathbf{c}, & \mathbf{D}^{\mathsf{T}}\boldsymbol{\pi} \leq \mathbf{0} \\ & \mathbf{e}\eta - \mathbf{V}^{\mathsf{T}}\boldsymbol{\rho} \leq \mathbf{0}, & \eta \geq 0. \end{array} \right. \tag{4.1}$$

We can obtain an optimal solution $(\overline{\boldsymbol{\pi}}, \overline{\eta}, \overline{\boldsymbol{\rho}})$ to (4.1) as a byproduct in solving $\overline{\mathrm{P}}(\Delta)$. For this $\overline{\boldsymbol{\pi}} \in \mathbb{R}^m$, let us define the following:

$$\mathrm{S}(\Delta) \left| \begin{array}{lll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x} & \\ \text{subject to} & \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{D}\mathbf{y} = \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{b}, & (\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \\ & \mathbf{x} - \mathbf{V}\boldsymbol{\lambda} = \mathbf{0}, & \boldsymbol{\lambda} \geq \mathbf{0} \\ & g(\mathbf{x}) \geq 0, & \end{array} \right.$$

where $\mathbf{x} \geq \mathbf{0}$ is redundant and can be eliminated. Let us denote by $z(\mathrm{S})$ the optimal value of this problem.

Proposition 4.1. *Between $z(S)$ and $z(\overline{P})$, there exists a relationship:*

$$z(S) \geq z(\overline{P}).$$

Proof. Consider the linear programming relaxation of $\mathrm{S}(\Delta)$:

$$\left| \begin{array}{lll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x} & \\ \text{subject to} & \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{D}\mathbf{y} = \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{b}, & \mathbf{y} \geq \mathbf{0} \\ & \mathbf{x} - \mathbf{V}\boldsymbol{\lambda} = \mathbf{0}, & \boldsymbol{\lambda} \geq \mathbf{0} \\ & \mathbf{e}^{\mathsf{T}}\boldsymbol{\lambda} \geq 1. & \end{array} \right. \tag{4.2}$$

The dual of this problem is

$$\left| \begin{array}{ll} \text{maximize} & \mathbf{b}^{\mathsf{T}}\overline{\boldsymbol{\pi}}\zeta + \eta \\ \text{subject to} & \mathbf{A}^{\mathsf{T}}\overline{\boldsymbol{\pi}}\zeta + \boldsymbol{\rho} = \mathbf{c}, \quad \mathbf{D}^{\mathsf{T}}\overline{\boldsymbol{\pi}}\zeta \le \mathbf{0} \\ & \mathbf{e}\eta - \mathbf{V}^{\mathsf{T}}\boldsymbol{\rho} \le \mathbf{0}, \quad \eta \ge 0. \end{array} \right. \tag{4.3}$$

Then $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ and $(1, \overline{\eta}, \overline{\boldsymbol{\rho}})$ are feasible for (4.2) and (4.3), respectively. Moreover, we have $\mathbf{c}^{\mathsf{T}}\overline{\mathbf{x}} = \mathbf{b}^{\mathsf{T}}\overline{\boldsymbol{\pi}} + \overline{\eta}$, and see that $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ and $(1, \overline{\eta}, \overline{\boldsymbol{\rho}})$ are optimal for these problems. Thus, even the relaxed problem of $\mathrm{S}(\Delta)$ has the same optimal value $z(\overline{\mathrm{P}})$ as $\overline{\mathrm{P}}(\Delta)$. $\square$

Problem $\mathrm{S}(\Delta)$ belongs to the same class of (2.1), but we can solve it in polynomial time if the value of g is given by oracle. Let $F' = \{\mathbf{x} \in \mathbb{R}^r \mid \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{A}\mathbf{x} \ge \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{b}\} \cap \Delta$. Then we have $\mathbf{x}^\circ \in \arg\min\{\mathbf{c}^{\mathsf{T}}\mathbf{x} \mid \mathbf{x} \in F'\}$, and further

$$F' = \{\mathbf{x} \in \mathbb{R}^r \mid \exists(\mathbf{y}, \boldsymbol{\lambda}) \ge \mathbf{0},\ \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{D}\mathbf{y} = \overline{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{b},\ \mathbf{x} - \mathbf{V}\boldsymbol{\lambda} = \mathbf{0},\ \mathbf{x} \ge \mathbf{0}\}$$

by noting $\mathbf{D}^{\mathsf{T}}\overline{\boldsymbol{\pi}} \le \mathbf{0}$ and $\mathbf{y} \ge \mathbf{0}$. We see from Proposition 2.1 that $\mathrm{S}(\Delta)$ has an optimal solution $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ such that $\widetilde{\mathbf{x}}$ lies on some edge of F'. Since F' is an intersection of the cone Δ with r edges and a halfspace, the maximum number of its edges is $r(r+1)/2$. This implies that $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ can be found if we evaluate g at most $O(r^2)$ times.

We are now ready to give a detailed description of our proposed algorithm for solving (2.1). Here, $\epsilon \ge 0$ is a given tolerance.

algorithm Conical_BB
begin
  $\Delta^1 := \mathrm{cone}(\{\mathbf{e}_1, \ldots, \mathbf{e}_r\})$, where $\mathbf{e}_i$ is the ith unit vector;
  $\mathcal{H} := \{\Delta^1\}$; $z^* := +\infty$;
  while $\mathcal{H} \ne \emptyset$ do begin
    select a cone $\Delta^k \in \mathcal{H}$; $\mathcal{H} := \mathcal{H} \setminus \{\Delta^k\}$;
    $\Delta := \Delta^k = \mathrm{cone}(\{\mathbf{w}_1, \ldots, \mathbf{w}_r\})$;
    for $i = 1, \ldots, r$ do compute β_i such that $g(\beta_i\mathbf{w}_i) = 0$ and $\beta_i > 0$;
    $\mathbf{V} := [\beta_1\mathbf{w}_1, \ldots, \beta_r\mathbf{w}_r]$;
    let Δ denote $\{\mathbf{x} \in \mathbb{R}^r \mid \mathbf{x} = \mathbf{V}\boldsymbol{\lambda},\ \boldsymbol{\lambda} \ge \mathbf{0}\}$;
                            /* *bounding operation* */
    solve $\overline{P}(\Delta)$, and obtain a lower bound $z^{\overline{P}}$;
    let $(\overline{\mathbf{x}}^k, \overline{\mathbf{y}}^k, \overline{\boldsymbol{\lambda}}^k)$ be an optimal solution to $\overline{P}(\Delta)$;
    if $g(\overline{\mathbf{x}}^k) \ge -\epsilon$ then begin
      if $z^{\overline{P}} < z^*$ then $z^* := z^{\overline{P}}$; $\mathbf{x}^* := \overline{\mathbf{x}}^k$; $\mathbf{y}^* := \overline{\mathbf{y}}^k$
    else if $z^{\overline{P}} < z^*$ then
      define $S(\Delta, \overline{\boldsymbol{\pi}})$ for the dual optimal solution $(\overline{\boldsymbol{\pi}}, \overline{\eta}, \overline{\boldsymbol{\rho}})$ to $\overline{P}(\Delta)$;
      solve $S(\Delta, \overline{\boldsymbol{\pi}})$, and obtain a lower bound z^S;
      search for a local optimal solution $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\lambda}})$ to $P(\Delta)$;
      if $\mathbf{c}^{\mathsf{T}}\widetilde{\mathbf{x}} < z^*$ then $z^* := \mathbf{c}^{\mathsf{T}}\widetilde{\mathbf{x}}$; $\mathbf{x}^* := \widetilde{\mathbf{x}}$; $\mathbf{y}^* := \widetilde{\mathbf{y}}$;
      if $z^S < z^*$ then begin

/* *branching operation* */

select the longest edge $\mathbf{w}_p$–$\mathbf{w}_q$ of $\mathrm{conv}(\{\mathbf{w}_1, \ldots, \mathbf{w}_r\})$;
let $\mathbf{w} := (1-\alpha)\mathbf{w}_p + \alpha\mathbf{w}_q$ for a fixed ratio $\alpha \in (0, 1/2]$;
$\Delta^{2k} := \mathrm{cone}(\{\mathbf{w}_i \mid i \neq p\} \cup \{\mathbf{w}\})$;
$\Delta^{2k+1} := \mathrm{cone}(\{\mathbf{w}_i \mid i \neq q\} \cup \{\mathbf{w}\})$;
$\mathcal{H} := \mathcal{H} \cup \{\Delta^{2k}, \Delta^{2k+1}\}$
end
end
end
end;

We refer to $(\mathbf{x}, \mathbf{y})$ satisfying the following as an *ϵ-feasible solution* to (2.1):

$$\mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b}, \quad (\mathbf{x}, \mathbf{y}) \geq \mathbf{0}, \quad g(\mathbf{x}) + \epsilon \geq \mathbf{0}.$$

Theorem 4.2. *If $\epsilon > 0$, then algorithm Conical_BB terminates after finitely many iterations and yields an ϵ-feasible solution $(\mathbf{x}^*, \mathbf{y}^*)$ to (2.1) such that $\mathbf{c}^\mathsf{T}\mathbf{x}^* \leq \mathbf{c}^\mathsf{T}\mathbf{x}$ for all $(\mathbf{x}, \mathbf{y})$ feasible to (2.1).*

Proof. When Conical_BB terminates in a finite number of iterations, the assertion is obvious. Suppose that the algorithm does not terminate and generates an infinite sequence of nested cones $\{\Delta^{k_\ell} \mid \Delta^{k_\ell} \supset \Delta^{k_{\ell+1}}, \ell = 1, 2, \ldots\}$ such that $g(\overline{\mathbf{x}}^{k_\ell}) < -\epsilon < 0$, $\ell = 1, 2, \ldots$. Let $\mathbf{v}_i^{k_\ell}$ be the ith column of $\mathbf{V}$ for each Δ^{k_ℓ}. Since $\bigcap_{\ell=1}^{\infty} \Delta^{k_\ell}$ is a half line, we have $\mathbf{v}_i^{k_\ell} \to \mathbf{v}$ as $\ell \to \infty$ for all $i = 1, \ldots, r$, where $g(\mathbf{v}) = 0$. Here, we should notice that $\overline{\mathrm{P}}(\Delta^{k_\ell})$ is also an instance of LPARC because $g'(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{e}^\mathsf{T}\boldsymbol{\lambda} - 1$ is a convex function. Let

$$\begin{aligned} F' &= \{(\mathbf{x}, \boldsymbol{\lambda}) \mid \exists \mathbf{y} \geq \mathbf{0},\ \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b},\ \mathbf{x} - \mathbf{V}\boldsymbol{\lambda} = \mathbf{0},\ (\mathbf{x}, \boldsymbol{\lambda}) \geq \mathbf{0}\} \\ G' &= \{(\mathbf{x}, \boldsymbol{\lambda}) \mid \mathbf{e}^\mathsf{T}\boldsymbol{\lambda} - 1 < 0\}. \end{aligned}$$

Then we have $(\mathbf{x}^\circ, \boldsymbol{\lambda}^\circ) = (\mathbf{0}, \mathbf{0}) \in \arg\min\{\mathbf{c}^\mathsf{T}\mathbf{x} \mid (\mathbf{x}, \boldsymbol{\lambda}) \in F'\}$ by assumption, and besides $(\mathbf{x}^\circ, \boldsymbol{\lambda}^\circ) \in G'$. Therefore, for the optimal solution $(\overline{\mathbf{x}}^{k_\ell}, \overline{\mathbf{y}}^{k_\ell}, \overline{\boldsymbol{\lambda}}^{k_\ell})$ of $\overline{\mathrm{P}}(\Delta^{k_\ell})$ we have $(\overline{\mathbf{x}}^{k_\ell}, \overline{\boldsymbol{\lambda}}^{k_\ell}) \in \partial G'$ by Proposition 2.1, where $\partial\cdot$ denotes the boundary; and $\mathbf{e}^\mathsf{T}\overline{\boldsymbol{\lambda}}^{k_\ell} = 1$ holds for each $\ell = 1, 2, \ldots$. This means that $\overline{\mathbf{x}}^{k_\ell}$ is given as convex combination of $\mathbf{v}_i^{k_\ell}$'s, and then $\overline{\mathbf{x}}^{k_\ell} \to \mathbf{v}$ as $\ell \to \infty$. Therefore, we have $g(\overline{\mathbf{x}}^{k_\ell}) \to 0$ as $\ell \to \infty$, which is a contradiction. □

5 Numerical results

In this section, we present numerical results of having compared our algorithm incorporating the surrogate relaxation $\mathrm{S}(\Delta)$ with a standard algorithm only using the linear programming relaxation $\overline{\mathrm{P}}(\Delta)$. We refer to those codes as cbb_s and cbb_lp, respectively. Both adopted the depth first rule in selecting $\Delta^k \in \mathcal{H}$, $\alpha = 1/2$, $\epsilon = 10^{-4}$ and were written using GNU Octave (version 2.1.34) [8] and C++ (GCC version 2.96) in part.

Table 5.1: Average numbers of branchig operations and CPU seconds

	LP relaxation		Surrogate relaxation	
$m,\ n,\ r$	branch	time(sec)	branch	time(sec)
10, 30, 10	4031	58	489	8
10, 40, 10	10959	129	1687	23
10, 50, 10	21998	334	5177	91
20, 50, 10	9891	190	3062	50
30, 50, 10	11392	237	3179	67
10, 40, 15		(3)		(3)
10, 40, 20		(7)		(3)

The test problem we solved is as follows:

$$\left| \begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} + [\mathbf{D}', \mathbf{I}]\mathbf{y} = \mathbf{e}, \quad (\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \\ & \displaystyle\sum_{j=1}^{r} \gamma_j x_j^2 \geq 1, \end{array} \right. \tag{5.1}$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$ is an identity matrix; $\mathbf{e} \in \mathbb{R}^m$ is an all-ones vector; each component in the last row of $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{D}' \in \mathbb{R}^{m\times(n-r-m)}$ was fixed at $1.0/(n-m)$, and other componets of $[\mathbf{A}, \mathbf{D}']$ were all random numbers in $[-2.0, 8.0]$, and about 50% of them were zeros; each component of $\mathbf{c} \in \mathbb{R}^r$ was drawn randomly from the uniform distribution on $[10.0, 11.0]$; and each number γ_j was positive and selected so that (5.1) was feasible but not trivial. Selecting seven sets of parameters (m, n, r), we solved ten instances of (5.1) for each set using cbb_s and cbb_lp on a Linux workstation (Linux 2.4.18, Itanium 2 processor 1GHz).

Table 5.1 shows the average number of branching operations and CPU seconds for each set of (m, n, r). Each figure in brackets represents the number of instances not solved in two hours. We see from this table that the surrogate relaxation $\mathrm{S}(\Delta)$ is of help to cut down the number of branching operations considerably, which also implies that the inequality in Proposition 4.1 held strictly in many iteration of cbb_s. As a consequence, cbb_s is much faster than cbb_lp in CPU seconds. Figure 5.1 compares the difference of the average CPU seconds taken by cbb_s and cbb_lp when $(m, n) = (10, 40)$. As for the instances not solved in two hours, the CPU seconds are plotted at $7,200$ seconds expediently.

Even though we tested the codes on rather limited instances, the performance of the proposed algorithm was promising, compared with the standard one. For some instances, however, both codes were numerically unstable due to rounding errors, and failed to terminate in two hours. In the next paper, we will discuss how to resolve this troublesome issue.

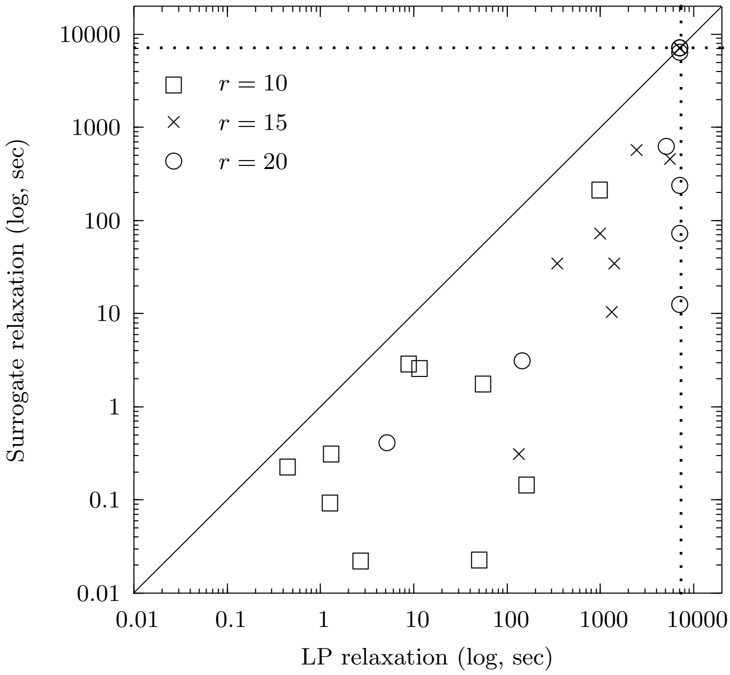

Figure 5.1: CPU seconds when $(m, n) = (10, 40)$

References

[1] R. J. Hillestad. Optimization problems subject to a budget constraint with economies of scale. *Operations Research* **23** (1975), 1091–1098.

[2] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*, 3 ed. Springer-Verlag, Berlin, 1996.

[3] H. Konno, P. T. Thach and H. Tuy. *Optimization on Low Rank Nonconvex Structures.* Kluwer Academic Publishers, Dordrecht, 1997.

[4] T. Kuno and H. Nagai. A simplicial algorithm with two-phase bounding operation for a class of concave minimization problems. *Pacific Journal of Optimization* **1** (2005), 297–313.

[5] T. Kuno and H. Nagai. A simplicial branch-and-bound algorithm conscious of special structures in concave minimization problems. Technical Report CS-TR-05-2, University of Tsukuba, Ibaraki, 2005.

[6] K. Moshirvaziri and M. A. Amouzegar. A subdivision scheme for linear programs with an additional reverse convex constraint. *Asia-Pacific Journal of Operations Research* **15** (1998), 179–192.

[7] L. D. Muu. A convergent algorithm for solving linear programs with an additional reverse convex constraint. *Kybernetika* **21** (1985), 428–435.

[8] Octave home page. http://www.octave.org/.

[9] H. Tuy. Concave programming under linear constraints. *Soviet Mathematics* **5** (1964), 1437–1440.

[10] H. Tuy. *Convex Analysis and Global Optimization.* Kluwer Academic Publishers, Dordrecht, 1998.

Hidetoshi Nagai
Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1, Ten'noudai, Tsukuba, Ibaraki 305-8573, Japan
E-mail address: `nagai@syou.cs.tsukuba.ac.jp`

Takahito Kuno
Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1, Ten'noudai, Tsukuba, Ibaraki 305-8573, Japan
E-mail address: `takahito@cs.tsukuba.ac.jp`