Annals of Operations Research 103, 193–211, 2001

Solving General Capacity Problem by Relaxed Cutting Plane Approach

SOON-YI WU * soonyi@mail.ncku.edu.tw
Department of Mathematics, National Cheng Kung University, Tainan, Taiwan, R.O.C.

SHU-CHERNG FANG ** fang@eos.ncsu.edu
Department of Industrial Engineering and Operations Research, North Carolina State University, Raleigh, NC 27695-7906, USA

CHIH-JEN LIN *** cjlin@csie.ntu.edu.tw
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract. This paper studies a class of infinite dimensional linear programming problems known as the general capacity problem. A relaxed cutting plane algorithm is proposed. A convergence proof together with some analysis of the results produced by the algorithm are given. A numerical example is also included to illustrate the computational procedure.

Keywords: capacity problem, infinite linear programming, cutting plane method, duality theory

1. Introduction

The general capacity problem studied in this paper is a special infinite linear program defined on some measure spaces [1]. Let X and Y be two compact Hausdorff spaces, $M(X)$ and $M(Y)$ be the spaces of regular Borel measures on X and Y, respectively, and $M^+(X)$ and $M^+(Y)$ be the subsets of $M(X)$ and $M(Y)$ consisting of the nonnegative Borel measures on X and Y, respectively. Given that $g(\cdot)$ and $f(\cdot)$ are real valued continuous functions defined on X and Y, respectively, and $\phi(\cdot,\cdot)$ is a real valued continuous function defined on $X \times Y$, a pair of primal and dual general capacity problems can be formulated in the following form:

$$(GCAP) \qquad \text{Minimize} \int_Y f(y)\,\mathrm{d}\mu(y) \qquad (1.1)$$

* Corresponding author.
** Supported by the National Textile Center and NC Furniture Manufacturing and Management Center.
*** Supported by the National Science Council of Taiwan, ROC, Grant No. NSC-88-2213-E-002-097.

$$\text{subject to} \quad \int_Y \phi(x, y)\, \mathrm{d}\mu(y) \geqslant g(x), \quad \forall x \in X,$$
$$\mu \in M^+(Y),$$

(*DGCAP*)
$$\text{Maximize} \int_X g(x)\, \mathrm{d}\nu(x) \tag{1.2}$$
$$\text{subject to} \quad \int_X \phi(x, y)\, \mathrm{d}\nu(x) \leqslant f(y), \quad \forall y \in Y,$$
$$\nu \in M^+(X).$$

The origin of the capacity problem is from the study of electrostatics in determining the capacity of a conducting body. As pointed out in Anderson and Nash [1], the electrostatic capacity problem and related potential theory have been intensively studies for many years. Some early studies on the capacity theory can be found in Choquet [2] and Fugled [5]. The recognition of the capacity problem as a general linear program is due to Yamasaki [18] and Ohtsuka [11]. In particular, Ohtsuka [12] showed that the duality gap between the pair of primal and dual general capacity problems may vanish under some conditions. However, when the conditions are not met, Yosida [19] provided an example showing that the optimal value of (*GCAP*) may exist while the feasible domain of (*DGCAP*) is empty. A related duality theorem was established by Wu [15]. The characterization of the extreme points of the feasible domain and the optimal solutions of the general capacity problem can be found in Wu [16] and Lai and Wu [10].

Note that (*GCAP*) and (*DGCAP*) are virtually identical, so we can choose to study either. A concrete example with X and Y being the unit interval of the real line and $\phi(x, y)$, $g(y)$ and $f(x)$ being analytic functions can be found in [1]. Moreover, an interesting connection between the general capacity problem and the two-person zero-sum infinite game was also identified by Wu [16]. By letting $P(Y) = \{\mu \in M^+(Y) \mid \mu(Y) = 1\}$, Wu showed that the following two-person zero-sum infinite game

(*TZG*)
$$\text{Minimize } \alpha$$
$$\text{subject to} \quad \int_Y \phi(x, y)\, \mathrm{d}\mu(y) \leqslant \alpha, \quad \forall x \in X,$$
$$\mu \in P(Y),$$

can be formulated as a general capacity problem of form (1.1). Related applications based on the (*TZG*) model can be found in Karlin [9].

In this paper, we concentrate on the algorithmic issues for solving the general capacity problem. Like Anderson and Nash [1], for concreteness and ease of exposition, throughout the rest of the paper we consider X and Y to be a closed real interval $[a, b]$, although they do not have to be an interval. When the optimal objective values of (*GCAP*) and (*DGCAP*) exist, they are denoted by $V(GCAP)$ and $V(DGCAP)$, respectively.

Like other infinite linear programming problems, the general capacity problem usually does not have a "closed form" solution. Some approximation schemes are applicable. But as pointed out by Hernandez-Lerma and Lasserre [7], most, if not all, approximation schemes for solving infinite linear programs are specifically designed for particular problems. We found very little literature containing computational algorithms for solving the general capacity problem.

Along the direction of approximation, one direct way to solve (*GCAP*) is using discretization techniques. Let $\{x_1, \ldots, x_n\}$ be a finite set of grid points in X and $\{y_1, \ldots, y_m\}$ in Y, then an approximation solution of (*GCAP*) can be obtained by solving a regular linear programming problem with m nonnegative variables and n inequality constraints. However, other than knowing the fact that "a discretization with more grid points in general results in a better approximation at the cost of the computational effort spent on solving a larger size linear program", no quality statement can be made about the approximation. A more sophisticated scheme is based on the observation that if the points of support for a feasible solution of (*GCAP*) are fixed, then the optimal values can be found by solving a semi-infinite linear program. Lai and Wu [10] introduced a cutting plane approach which generates a sequence of optimal extreme points of corresponding semi-infinite linear programming problems in a systematic way and showed that the sequence converges to an optimal solution of (*GCAP*).

With the recent advance in designing relaxation schemes for solving semi-infinite programming problems [3,17], in this paper, we intend to construct a relaxed cutting plane approximate algorithm for solving the general capacity problem. The proposed algorithm is introduced in section 2. A convergence proof and related analysis of the proposed algorithm are presented in section 3. Then a numerical example is used for illustration in section 4. Some concluding remarks are made in section 5.

2. Proposed algorithm

A cutting plane approach was proposed by Lai and Wu [10] for solving the general capacity problem. Basically, it discretizes the X space in to a finite number of grid points to construct a semi-infinite programming problem for an approximate solution. If the result is good enough, the algorithm stops. Otherwise, it refines the set of grid points for a better solution. The approach leads to the following iterative scheme:

Step 1. Set $k = 1$, choose any $x^1 \in X$, and set $T_1 = \{x^1\}$.

Step 2. Find an optimal solution μ^k for the following semi-infinite programming problem:

(*SIP*$_k$)
$$\text{Minimize} \int_Y f(y)\,\mathrm{d}\mu(y) \tag{2.1}$$
$$\text{subject to} \quad \int_Y \phi\big(x^j, y\big)\,\mathrm{d}\mu(y) \geqslant g\big(x^j\big), \quad j = 1, \ldots, k,$$
$$\mu \in M^+(Y).$$

Step 3. Find a minimizer x^{k+1} of $\phi_k(x)$ over X where

$$\phi_k(x) \equiv \int_Y \phi(x, y)\, \mathrm{d}\mu^k(y) - g(x). \tag{2.2}$$

Step 4. If $\phi_k(x^{k+1}) \geqslant 0$, then stop and μ^k is optimal for (*GCAP*). Otherwise, set $T_{k+1} = T_k \cup \{x^{k+1}\}$, increment $k \leftarrow k + 1$, and go to step 2.

In the iterative scheme, one constraint is added at a time and the computational bottleneck falls either in solving a semi-infinite program (SIP_k) or in finding a global minimizer x^{k+1} of $\phi_k(x)$.

To solve (SIP_k), we may consider a dual problem:

$$(DSIP_k) \qquad \text{Maximize } \sum_{j=1}^{k} g\left(x^j\right) v_j \tag{2.3}$$

$$\text{subject to } \quad \sum_{j=1}^{k} \phi\left(x^j, y\right) v_j \leqslant f(y), \quad \forall y \in Y,$$

$$v_j \geqslant 0, \quad j = 1, \ldots, k.$$

A solution to ($DSIP_k$) can be obtained by either discretizing Y or applying another level of cutting plane approach on Y. Related method can be found in [4,6,8,13]. A solution to (SIP_k) can then be obtained from the primal–dual relation.

The intention of this paper is to design a better cutting plane algorithm for solving the general capacity problem. The new algorithm will relax the computational requirements for potential bottlenecks. In particular, the following issues will be directly addressed:

1. Relaxing the computational work required for finding a global minimizer x^{k+1} in step 3. Finding a minimizer of a continuous function $\phi_k(x)$ over X may involve in heavy computation. Here we explore an idea whereby the subproblem (SIP_{k+1}) is constructed by choosing a point $\bar{x}^{k+1}$ at which the infinite constraints are violated, i.e., $\phi_k(\bar{x}^{k+1}) < 0$, rather than choosing the point x^{k+1} where the violation is maximized, i.e., $\phi_k(x)$ is minimized.

2. Relaxing the computation requirement for finding an optimal solution μ_k to (SIP_k) in step 2. The idea of taking μ_k to be an approximate solution to (SIP_k) will be explored.

3. Relaxing the computational work required for solving a potentially large scale (SIP_k). The idea of dropping unnecessary constraints while a new constraint is added in each iteration will be explored.

Because of the third consideration, at the kth iteration, we may have a set of n ($n \leqslant k$) "critical" grid points, say $T_k = \{\bar{x}^1, \ldots, \bar{x}^n\} \subset \{x^1, \ldots, x^k\}$, which defines a semi-infinite programming problem:

$$(SIP_{T_k}) \qquad \text{Minimize} \int_Y f(y)\,\mathrm{d}\mu(y) \tag{2.4}$$

$$\text{subject to} \quad \int_Y \phi(\bar{x}^j, y)\,\mathrm{d}\mu(y) \geqslant g(\bar{x}^j), \quad \bar{x}^j \in T_k,$$
$$\mu \in M^+(Y).$$

Its dual is of the following form:

$$(DSIP_{T_k}) \qquad \text{Maximize} \sum_{j=1}^{n} g(\bar{x}^j) v_j \tag{2.5}$$

$$\text{subject to} \quad \sum_{\bar{x}_j \in T_k} \phi(\bar{x}^j, y) v_j \leqslant f(y), \quad \forall y \in Y,$$
$$v_j \geqslant 0, \quad \bar{x}_j \in T_k.$$

A relaxed cutting plane algorithm is proposed as follows:

Step 0. Let $\delta > 0$ be a sufficiently small number, $\Delta_1 > 0$, $0 < \varepsilon < 1$, and $n_0 = 0$.
Step 1. Set $k = 1$, choose any $x_1^0 \in X$, and set $T_1 = \{x_1^0\}$.
Step 2. Find an approximate (feasible) solution μ^k of (SIP_{T_k}), where μ^k is a discrete measure concentrated on $y_1^k, \ldots, y_{m_k}^k$, and an approximate (maybe infeasible) solution $v^k = (v_1^k, \ldots, v_{n_{k-1}+1}^k)$ of ($DSIP_{T_k}$) such that the following conditions are met:

$$\sum_{j=1}^{n_k} \left|\min(s_j^k, 0)\nu^k(x_j^k)\right| < \Delta_k, \tag{2.6}$$

$$\sum_{i=1}^{m_k} \left|\max(0, \bar{\phi}_{k-1}(y_i^k))\mu^k(y_i^k)\right| < \Delta_{k-1} \quad (\text{if } k \geqslant 2), \tag{2.7}$$

$$\bar{\phi}_k(y) \leqslant \Delta_k, \quad \forall y \in Y, \tag{2.8}$$

$$0 \leqslant V\big(SIP_{T_k}(\mu^k)\big) - V\big(DSIP_{T_k}(\nu^k)\big) < \Delta_k, \tag{2.9}$$

where ν^k is a nonnegative discrete measure defined on X with

$$\nu^k(x) = \begin{cases} v_j^k\ (\geqslant 0) & \text{if } x = x_j^{k-1} \in T_k, \\ 0 & \text{if } x \in X \setminus T_k, \end{cases} \tag{2.10}$$

$$B_k \equiv \{x \in T_k \mid \nu^k(x) > 0\} = \{x_1^k, \ldots, x_{n_k}^k\},$$

$$\bar{\phi}_k(y) \equiv \int_X \phi(x, y)\,\mathrm{d}\nu^k(x) - f(y), \tag{2.11}$$

$$s_j^{k-1} \equiv \sum_{i=1}^{m_k} \phi\big(x_j^{k-1}, y_i^k\big)\mu^k\big(y_i^k\big) - g\big(x_j^{k-1}\big),$$
$$j = 1, \ldots, n_{k-1} \text{ (if } k \geqslant 2). \tag{2.12}$$

Define $\phi_k(x)$ according to (2.2).

Step 3. Find any $x_{n_k+1}^k \in X$ such that $\phi_k(x_{n_k+1}^k) < -\delta$. If such an $x_{n_k+1}^k$ does not exist and $\Delta_k < \delta$, stop and output μ^k as the solution.

If $k = 1$ or $V(DSIP_{T_k}(\upsilon^k)) \geqslant V(DSIP_{T_{k-1}}(\upsilon^{k-1}))$ but $x_{n_k+1}^k$ exists, set $T_{k+1} = B_k \cup \{x_{n_k+1}^k\}$; otherwise, set $\Delta_k = (1-\varepsilon)\Delta_k$ and go to step 2.

Step 4. Update k by $k+1$ and go to step 2.

Note that the initial point selected from X is denoted by x_1^0 because only if its corresponding measure $\upsilon_1^1 > 0$, x_1^0 is included in B_1 as a member. Also note that $x_{n_k+1}^k \notin B_k$, $\forall k$. In step 2, a pair of approximate solutions of (SIP_{T_k}) and ($DSIP_{T_k}$) satisfying (2.6)–(2.9) can be obtained by using a relaxed scheme like [17]. Moreover, if the algorithm terminates in step 3, then the output solution μ^k is an approximation solution of (*GCAP*) when δ is small enough. The more detailed properties including a convergence proof of the proposed algorithm will be analyzed in the next section.

3. Analysis of algorithm

To simplify the analysis, we assume that (SIP_{T_k}) is solvable with an optimal value $V(SIP_{T_k})$ and ($DSIP_{T_k}$) is also solvable with an optimal value $V(DSIP_{T_k})$. Since μ^k and ν^k may not be optimal solutions to (SIP_{T_k}) and ($DSIP_{T_k}$), respectively, we denote their objective values by $V(SIP_{T_k}(\mu^k))$ and $V(DSIP_{T_k}(\upsilon^k))$, respectively. In this way, we have

$$V\big(SIP_{T_k}\big(\mu^k\big)\big) = \sum_{i=1}^{m_k} f\big(y_i^k\big)\mu^k\big(y_i^k\big),$$

and

$$V\big(DSIP_{T_k}\big(\upsilon^k\big)\big) = \sum_{j=1}^{n_k} g\big(x_j^k\big)\upsilon^k\big(x_j^k\big).$$

Similar to the relation between ν^k and υ^k as defined in (2.10), on the primal side, we define $u_i^k \equiv \mu(y_i^k)$, for $i = 1, \ldots, m_k$. We also define the matrix H_k to be an $n_k \times m_k$ matrix with its rth row vector being

$$\big(\phi\big(x_r^k, y_1^k\big), \phi\big(x_r^k, y_2^k\big), \ldots, \phi\big(x_r^k, y_{m_k}^k\big)\big) \quad \text{for } r = 1, \ldots, n_k,$$

and let $\bar{\Delta}_k = V(SIP_{T_k}(\mu^k)) - V(DSIP_{T_k}(\upsilon^k))$.

Then we have the following result:

Theorem 3.1. Let ν^k, ν^{k+1}, μ^k, μ^{k+1} and s^k be generated in the kth and the $(k+1)$st iteration of the proposed algorithm. Then

$$V\big(DSIP_{T_{k+1}}(\nu^{k+1})\big) - V\big(DSIP_{T_k}(\nu^k)\big) = \sum_{j=1}^{n_k} s_j^k \nu^k\big(x_j^k\big) - \sum_{i=1}^{m_{k+1}} \bar{\phi}_k\big(y_i^{k+1}\big)\mu^{k+1}\big(y_i^{k+1}\big) - \bar{\Delta}_{k+1}. \tag{3.1}$$

Proof. By the definition of $\bar{\phi}_k(y)$, we have

$$\begin{aligned}
&\sum_{i=1}^{m_{k+1}} \bar{\phi}_k\big(y_i^{k+1}\big)\mu^{k+1}\big(y_i^{k+1}\big)\\
&= \sum_{i=1}^{m_{k+1}} \left(\sum_{j=1}^{n_k} \phi\big(x_j^k, y_i^{k+1}\big)\nu^k\big(x_j^k\big) - f\big(y_i^{k+1}\big)\right)\mu^{k+1}\big(y_i^{k+1}\big)\\
&= \sum_{j=1}^{n_k}\sum_{i=1}^{m_{k+1}} \phi\big(x_j^k, y_i^{k+1}\big)\mu^{k+1}\big(y_i^{k+1}\big)\nu^k\big(x_j^k\big) - \sum_{i=1}^{m_{k+1}} f\big(y_i^{k+1}\big)\mu^{k+1}\big(y_i^{k+1}\big)\\
&= \sum_{j=1}^{n_k}\big(g\big(x_j^k\big) + s_j^k\big)\nu^k\big(x_j^k\big) - V\big(SIP_{T_{k+1}}\big(\mu^{k+1}\big)\big) \quad \text{(from definition (2.12))}\\
&= \sum_{j=1}^{n_k} g\big(x_j^k\big)\nu^k\big(x_j^k\big) + \sum_{j=1}^{n_k} s_j^k\nu^k\big(x_j^k\big) - V\big(SIP_{T_{k+1}}\big(\mu^{k+1}\big)\big)\\
&= V\big(DSIP_{T_k}\big(\nu^k\big)\big) + \sum_{j=1}^{n_k} s_j^k\nu^k\big(x_j^k\big) - V\big(DSIP_{T_{k+1}}\big(\nu^{k+1}\big)\big) - \bar{\Delta}_{k+1}.
\end{aligned}$$

It follows that

$$V\big(DSIP_{T_{k+1}}(\nu^{k+1})\big) - V\big(DSIP_{T_k}(\nu^k)\big) = \sum_{j=1}^{n_k} s_j^k \nu^k\big(x_j^k\big) - \sum_{i=1}^{m_{k+1}} \bar{\phi}_k\big(y_i^{k+1}\big)\mu^{k+1}\big(y_i^{k+1}\big) - \bar{\Delta}_{k+1}. \qquad \square$$

One of our major objectives is to establish a convergence proof for the proposed algorithm. From the previous work [10], we know that this is not a straight-forward job. Some technical conditions have to be introduced for this purpose. In the subsequent analysis, we let $C^\infty(S)$ be the set of all analytic functions defined on a space S, M^* denotes a sufficiently large positive number, and ε^* denotes a sufficiently small number. We focus our analysis on the case that functions $f, g \in C^\infty([a, b])$ and $\phi(x, y) \in C^\infty([a, b] \times [a, b])$, with $a < 0$ and $b > 1$, such that $|\partial\phi/\partial y|$, $|\phi|$, $|f|$ and $|g|$ are bounded by M^*.

Note from theorem 3.1 that the duality gap in each iteration of the proposed algorithm is somehow depends on the functions $\bar{\phi}_k(\cdot)$, $\nu^k(\cdot)$, and $\mu^k(\cdot)$. In our analysis, we will impose two technical conditions on $\bar{\phi}_k(\cdot)$ and one condition on $\nu^k(\cdot)$ and $\mu^k(\cdot)$ to prove the convergence of the proposed algorithm.

By the definition of (2.11), we know that $\bar{\phi}_k(y) \in C^\infty([a,b])$. The first technical condition is imposed on $\bar{\phi}_k(y)$, for k being large enough.

Technical condition 1.

(i) $\bar{\phi}_k(y) \leqslant \varepsilon^*$, $\forall y \in [a,b]$.

(ii) $\bar{\phi}_k(y)$ achieves local optimality only at a finite number of points $y_p \in [a,b]$, (say $p = 1,\ldots,m$), and for each p there exists an i_p ($< M^*$) such that if $i_p \neq 1$, the jth derivative $\bar{\phi}_k^{(j)}(y_p) = 0$, for $j = 1,\ldots,i_p - 1$, but $\bar{\phi}_k^{(i_p)}(y_p) \neq 0$. If $i_p = 1$, then $\bar{\phi}_k^{(i_p)}(y_p) \neq 0$.

(iii) For $p = 1,2,\ldots,m$, if $y \in N_{\varepsilon^*}(y_p)$ (the ε^*-neighborhood of y_p), then $\bar{\phi}_k^{(i_p)}(y)$ has the same sign as $\bar{\phi}_k^{(i_p)}(y_p)$ and $M^* \geqslant |\bar{\phi}_k^{(i_p)}(y)| \geqslant \varepsilon^*$.

(iv) If $\bar{y}$ is a local maximum or minimum of $\bar{\phi}_k(y)$ over $[a,b]$ and $\bar{\phi}_k(\bar{y}) < 0$, then $\bar{\phi}_k(\bar{y}) \leqslant -\varepsilon^*$.

Note that technical condition 1 is the so-called "RA condition" used in [10]. If $\bar{\phi}(y)$ is a smooth function with $\bar{\phi}(y) \leqslant \varepsilon^*$, $\forall y \in [a,b]$, in general, (ii)–(iv) of technical condition 1 can be satisfied without much difficulty. In the proposed algorithm, $\bar{\phi}_k(y) \leqslant \Delta_k$, $\forall y \in [a,b]$, is required by (2.8) and Δ_k is updated in step 3 to be monotonically decreasing to 0. Therefore, there exists $\overline{N} > 0$ such that $\Delta_k \leqslant \varepsilon^*$, $\forall k \geqslant \overline{N}$. Also remember that $\bar{\phi}_k(y)$ is an analytic function. Hence for $k \geqslant \overline{N}$, $\bar{\phi}_k(y)$ will in general satisfy this technical condition without much trouble.

Parallel to the proof of lemma 6.3 of [10], the following lemma can be derived and will be used in the convergence proof.

Lemma 3.2. Assume that, for $k \geqslant \overline{N}$, $\bar{\phi}_k(y)$ satisfies technical condition 1 and it has only $\overline{m}_k$ local maxima $\bar{y}_i^k \in [a,b]$, for $i = 1,\ldots,\overline{m}_k$, such that $0 \leqslant \bar{\phi}_k(\bar{y}_i^k) \leqslant \Delta_k$. If there exist $y \in [a,b]$ and a sufficiently small $\varepsilon > 0$ such that $|\bar{\phi}_k(y)| < \varepsilon$, then there exists $j \in \{1,\ldots,\overline{m}_k\}$ such that $|y - \bar{y}_j^k| < r(\Delta_k,\varepsilon)$ with $r(\Delta_k,\varepsilon) \to 0$ whenever $\Delta_k \to 0$ and $\varepsilon \to 0$.

The second technical condition is imposed on μ^k and ν^k. Note that, in the kth iteration, the proposed algorithm generates $\mu^k(\cdot)$, $\nu^k(\cdot)$, x_j^k, $j = 1,\ldots,n_k$, and y_i^k, $i = 1,\ldots,m_k$. Based on these $x's$ and $y's$, we can construct a pair of primal and dual finite linear programs, which can be used to approximate (SIP_{T_k}) and ($DSIP_{T_k}$), respectively:

$$(LP_k) \qquad \text{Minimize} \sum_{i=1}^{m_k} f\big(y_i^k\big)u_i \tag{3.2}$$
$$\text{subject to} \quad \sum_{i=1}^{m_k} \phi\big(x_j^k, y_i^k\big)u_i \geqslant g\big(x_j^k\big), \quad j = 1, \dots, n_k,$$
$$u_i \geqslant 0, \quad i = 1, \dots, m_k.$$

$$(DLP_k) \qquad \text{Maximize} \sum_{j=1}^{n_k} g\big(x_j^k\big)v_j \tag{3.3}$$
$$\text{subject to} \quad \sum_{j=1}^{n_k} \phi\big(x_j^k, y_i^k\big)v_j \leqslant f\big(y_i^k\big), \quad i = 1, \dots, m_k,$$
$$v_j \geqslant 0, \quad j = 1, \dots, n_k.$$

The second technical condition is described as below.

Technical condition 2. For $k \geqslant \overline{N}$, μ^k and ν^k satisfy that

(i) $\|\mu^k\| \leqslant M^*$;

(ii) $\nu^k(x_j^k) \geqslant \varepsilon^*$, for $j = 1, \dots, n_k$, and $\nu^{k+1}(x_{n_k+1}^k) \geqslant \varepsilon^*$;

(iii) $\{\mu^k(y_i^k) \mid i = 1, \dots, m_k\}$ is an optimal solution to (LP_k) and $\{\nu^k(x_j^k) \mid j = 1, \dots, n_k\}$ is optimal for (DLP_k).

Note that the violation of (ii) of technical condition 2 will lead to dual degeneracy.

The condition of $\mu^k(\cdot)$ being optimal to (LP_k) implies that μ^k is a feasible solution of (SIP_{T_k}). This explains why the primal feasibility is required in step 2 of the proposed algorithm. Also note the fact that $\mu^k(y_i^k) > 0$ for each i, which, together with (iii), implies that $\bar{\phi}_k(y_i^k) = 0$, for $i = 1, \dots, m_k$.

The third technical condition is further imposed on $\bar{\phi}_k(\cdot)$, for k being large enough, to regularize its shape from being "ill conditioned". Remember that $y_1^k, \dots, y_{m_k}^k$ are generated by the proposed algorithm in the kth iteration, and $\bar{y}_1^k, \dots, \bar{y}_{\overline{m}_k}^k$ are the local maxima of $\bar{\phi}_k(\cdot)$ as stated in lemma 3.2. It is important to realize that they are related, but different. Let $\bar{\delta} > 0$ be sufficiently small, the required technical condition is stated as below.

Technical condition 3. For $k \geqslant \overline{N}$, if $\bar{y}_i^k$ is a local maximum of $\bar{\phi}_k(\cdot)$ with $0 < \bar{\phi}_k(\bar{y}_i^k) < \bar{\delta}$, then

(i) the values of $\bar{\phi}_k(\cdot)$ on the left sides of $\bar{y}_i^k$ strictly decrease to either the boundary point a or a root of $\bar{\phi}_k(\cdot)$ and the function values on the right sides of $\bar{y}_i^k$ strictly decrease to either the boundary point b or a root of $\bar{\phi}_k(\cdot)$;

(ii) one and only one of the above two limiting points is in $\{y_1^k, \ldots, y_{m_k}^k\}$;

(iii) except these limiting points and $y_1^k, \ldots, y_{m_k}^k$, $\bar{\phi}_k(y) = 0$ has no other possible solution.

A result similar to lemma 3.2 immediately follows.

Lemma 3.3. Assume that, for $k \geqslant \overline{N}$, $\bar{\phi}_k(y)$ satisfies technical conditions 1 and 3, and it has only $\overline{m}_k$ local maxima $\bar{y}_i^k \in [a, b]$, for $i = 1, \ldots, \overline{m}_k$, such that $0 \leqslant \bar{\phi}_k(\bar{y}_i^k) \leqslant \Delta_k$. For any y_j^k, $j = 1, \ldots, m_k$, with $\phi_k(y_j^k) = 0$, if $\bar{y} \in \{\bar{y}_1^k, \ldots, \bar{y}_{\overline{m}_k}^k\}$ is a neighboring local maximum of $\bar{\phi}_k$ (i.e., from one side of $\bar{y}$, the function value of $\bar{\phi}_k$ decreases to zero), then $|\bar{y} - y_j^k| < r(\Delta_k, \varepsilon)$ with $r(\Delta_k, \varepsilon) \to 0$ whenever $\Delta_k \to 0$ and $\varepsilon \to 0$.

We are now ready for the convergence proof.

Theorem 3.4. Given any $\delta > 0$ in step 0, for $\overline{N}$ being a sufficiently large integer such that technical conditions 1–3 are met for $k \geqslant \overline{N}$, if there exists a square sub-matrix D_k of H_k having rank m_k such that $|\det(D_k)|$ is bounded away from zero, then the proposed algorithm stops in a finite number of iterations.

Proof. Suppose the algorithm does not stop in a finite number of iterations. By step 3 of the proposed algorithm, we have

$$V\big(DSIP_{T_1}(v^1)\big) < V\big(DSIP_{T_2}(v^2)\big) < \cdots \leqslant V(GCAP).$$

Thus $\lim_{k\to\infty} V(DSIP_{T_k}(v^k)) = \alpha \leqslant V(GCAP)$. We claim this is impossible.

Let $K = \{\mu \in M([a, b]) \mid \|\mu\| \leqslant M^*\}$. Since $C([a, b])$ (the space of all continuous functions defined on $[a, b]$) is separable, the weak* compact set K is sequential compact. Technical condition 2 guarantees that the infinite sequences $\{\mu^k\}$ and $\{v^k\}$ are confined in the weak* compact set K. Note that $[a, b]$ is a compact Hausdorff space, hence there exists a subsequence $\{\mu^{k_r}\}$ of $\{\mu^k\}$ which is weak* convergent to a nonnegative measure μ^*. Similarly, there exists a subsequence $\{v^{k_r}\}$ of $\{v^k\}$ which is weak* convergent to a nonnegative measure v^*. Moreover, the sequence $\{x_{n_{k_r}+1}^{k_r}\}_{r=1}^{\infty}$ converges to a point x_*, as $r \to \infty$. By letting

$$\phi_*(x) = \int_Y \phi(x, y)\,\mathrm{d}\mu^*(y) - g(x),$$

we see that $\phi_{k_r}(x_{n_{k_r}+1}^{k_r})$ converges to $\phi_*(x_*)$. Since $\phi_{k_r}(x_{n_{k_r}+1}^{k_r}) < -\delta$, $\forall r$, we have $\phi_*(x_*) \leqslant -\delta < 0$.

Let $\varepsilon \in (0, \delta)$ be an arbitrary number. We can find a large enough integer $N \in \{k_r\}_{r=1}^{\infty}$ ($N \geqslant \overline{N}$) such that

$$\big|V\big(DSIP_N(v^N)\big) - \alpha\big| < \varepsilon^2, \quad \|\Delta_N\| \leqslant \varepsilon^2 \quad \text{and} \quad \big|\phi_N\big(x_{n_N+1}^N\big) - \phi_*(x_*)\big| < \varepsilon^2.$$

Implied by theorem 3.1, we have

$$\begin{aligned}&\left|V\left(DSIP_{N+1}\left(v^{N+1}\right)\right)-V\left(DSIP_{N}\left(v^{N}\right)\right)\right|\\&\quad=\left|\sum_{j=1}^{n_N} s_j^N \nu^N\left(x_j^N\right)-\sum_{i=1}^{m_{N+1}} \bar{\phi}_N\left(y_i^{N+1}\right)\mu^{N+1}\left(y_i^{N+1}\right)-\bar{\Delta}_{N+1}\right|<\varepsilon^2. \qquad (3.4)\end{aligned}$$

Let

$$\begin{aligned}T_N^{P'}&=\left\{j \mid s_j^N<0,\ j=1,\dots,n_N\right\} \quad \text{and}\\ T_N^{D'}&=\left\{i \mid \bar{\phi}_N\left(y_i^{N+1}\right)>0,\ i=1,\dots,m_{N+1}\right\}.\end{aligned}$$

We have, by (2.6) and (2.7),

$$\Delta_k^{P'} \equiv \sum_{j\in T_k^{P'}}\left|s_j^k \nu^k\left(x_j^k\right)\right|<\Delta_k \quad \text{and} \quad \Delta_k^{D'} \equiv \sum_{i\in T_k^{D'}}\left|\bar{\phi}_k\left(y_i^{k+1}\right)\mu^{k+1}\left(y_i^{k+1}\right)\right|<\Delta_k, \quad \forall k. \qquad (3.5)$$

Thus

$$\begin{aligned}&\left|\sum_{j\in\{1,\dots,n_N\}\setminus T_N^{P'}} s_j^N \nu^N\left(x_j^N\right)-\sum_{i\in\{1,\dots,m_{N+1}\}\setminus T_N^{D'}} \bar{\phi}_N\left(y_i^{N+1}\right)\mu^{N+1}\left(y_i^{N+1}\right)\right|\\&\quad<\varepsilon^2+\bar{\Delta}_{N+1}+\Delta_N^{P'}+\Delta_N^{D'}. \qquad (3.6)\end{aligned}$$

Without loss of generality, we may assume that $\mu^{N+1}(y_i^{N+1}) \geqslant \varepsilon$, for $i=1,2,\dots,T$, with $T \leqslant m_{N+1}$ and $\mu^{N+1}(y_i^{N+1})<\varepsilon$, for $i=T+1,\dots,m_{N+1}$.

Note that

$$\bar{\phi}_N\left(y_i^{N+1}\right) \leqslant 0,\ \ \forall i \in \{1,2,\dots,m_{N+1}\}\setminus T_N^{D'} \quad \text{and} \quad s_j^N \geqslant 0,\ \ \forall j \in \{1,\dots,n_N\}\setminus T_N^{P'},$$

each term in (3.6) is nonnegative and less then $4\varepsilon^2$. Therefore,

$$\left|\bar{\phi}_N\left(y_i^{N+1}\right)\right|<\frac{4\varepsilon^2}{\mu^{N+1}(y_i^{N+1})}, \quad \forall i \in \{1,\dots,T\}\setminus T_N^{D'}.$$

Since $\mu^{N+1}(y_i^{N+1}) \geqslant \varepsilon, \forall\, i=1,\dots,T$, we further have

$$\left|\bar{\phi}_N\left(y_i^{N+1}\right)\right|<4\varepsilon, \quad \forall i \in \{1,\dots,T\}\setminus T_N^{D'}. \qquad (3.7)$$

By (2.8),

$$0<\bar{\phi}_N\left(y_i^{N+1}\right)<\varepsilon^2<\varepsilon, \quad \forall i \in T_N^{D'}. \qquad (3.8)$$

Since $\|\Delta_N\| \leqslant \varepsilon$, combining (3.7) and (3.8) to meet the conditions of lemma 3.2, we can choose sufficiently small $\varepsilon>0$ and find $s_i(\Delta_N,\varepsilon)>0$, for $i=1,\dots,T$, such that $y_i^{N+1} \in N_{s_i(\Delta_N,\varepsilon)}(\bar{y}_{\bar{r}_i}^N)$ (the neighborhood of $\bar{y}_{\bar{r}_i}^N$), where $\bar{r}_i \in \{1,\dots,\overline{m}_N\}$, $\bar{y}_{\bar{r}_i}^N$ is a local maximum of $\bar{\phi}_k$, and $s_i(\Delta_N,\varepsilon) \to 0$ as $\varepsilon \to 0$. Following (ii) of technical condition 3, there exist $y_{r_i}^N \in \{y_1^N,\dots,y_{m_N}^N\}$ such that $y_{r_i}^N$ is a neighboring root of $\bar{y}_{\bar{r}_i}^N$.

Then from lemma 3.3, there exist $\bar{s}_i(\Delta_N, \varepsilon) > 0$ such that $|y_{r_i}^N - \bar{y}_{\bar{r}_i}^N| < \bar{s}_i(\Delta_N, \varepsilon)$, where $\bar{s}_i(\Delta_N, \varepsilon) \to 0$ as $\varepsilon \to 0$. Therefore, $|y_i^{N+1} - y_{r_i}^N| < s_i(\Delta_N, \varepsilon) + \bar{s}_i(\Delta_N, \varepsilon)$. In other words, we can choose a sufficiently small $\varepsilon > 0$ and find $q_i(\Delta_N, \varepsilon) \equiv s_i(\Delta_N, \varepsilon) + \bar{s}_i(\Delta_N, \varepsilon)$, for $i = 1, \ldots, T$, such that $y_i^{N+1} \in N_{q_i(\Delta_N,\varepsilon)}(y_{r_i}^N)$ and $y_{i'}^{N+1} \in N_{q_{i'}(\Delta_N,\varepsilon)}(y_{r_{i'}}^N)$ with $N_{q_i(\Delta_N,\varepsilon)}(y_{r_i}^N)$ and $N_{q_{i'}(\Delta_k,\varepsilon)}(y_{r_{i'}}^N)$ being disjoint when $i \neq i'$.

Similarly, by the fact that $\bar{\phi}_{N+1}$ satisfies the technical conditions 1 and 3, there exists a positive integer M' such that $m_{N+1} \leqslant M'$, $\forall N$. Since, for $i = 1, \ldots, T$, $|y_i^{N+1} - y_{r_i}^N| < q_i(\Delta, \varepsilon)$, and $\phi \in C^\infty([a, b] \times [a, b])$, we know that, as $\varepsilon \to 0$,

$$\max_{i=1,\ldots,T} \max_{j=1,\ldots,n_N} \left|\phi\left(x_j^N, y_i^{N+1}\right) - \phi\left(x_j^N, y_{r_i}^N\right)\right| \to 0 \quad \text{as } N \to \infty. \tag{3.9}$$

Technical condition 2 guarantees that $\nu^N(x_j^N) \geqslant \varepsilon^*$, for $j = 1, \ldots, n_N$. It follows from (3.6) that

$$\left|s_j^N\right| \leqslant \varepsilon, \quad \forall j \in \{1, \ldots, n_N\} \setminus T_N^{P'}. \tag{3.10}$$

Moreover, by (3.5), we have $|s_j^N| \leqslant \varepsilon,\ \forall j \in T_N^{P'}$. Since

$$\sum_{i=1}^{m_{N+1}} \phi\left(x_j^N, y_i^{N+1}\right)\mu^{N+1}\left(y_i^{N+1}\right) = g\left(x_j^N\right) + s_j^N \quad \text{for } j = 1, \ldots, n_N,$$

it follows that

$$\sum_{i=1}^{T} \phi\left(x_j^N, y_i^{N+1}\right)\mu^{N+1}\left(y_i^{N+1}\right) = g\left(x_j^N\right) + \varepsilon_j^N(\varepsilon) \tag{3.11}$$

with $\max_{j=1,\ldots,n_N} \varepsilon_j^N(\varepsilon) \to 0$, as $\varepsilon \to 0$.

Technical condition 2 also asserts that $\{\mu^k(y_i^k) \mid i = 1, \ldots, m_k\}$ is optimal to (LP_k) and $\{\nu^k(x_j^k) \mid j = 1, \ldots, n_k\}$ is optimal to (DLP_k). Since $\mu^k(y_j^k)$ and $\nu^k(x_j^k)$ are all positive, the complementarity slackness condition implies

$$\sum_{i=1}^{T} \phi\left(x_j^N, y_{r_i}^N\right)\mu^N\left(y_{r_i}^N\right) + \sum_{k=1}^{m_N - T} \phi\left(x_j^N, y_{t_k}^N\right)\mu^N\left(y_{t_k}^N\right) = g\left(x_j^N\right) \quad \text{for } j = 1, \ldots, n_N, \tag{3.12}$$

where $t_k \in \{1, \ldots, m_N\} \setminus \{r_1, \ldots, r_T\}$, with $t_{k_1} \neq t_{k_2}$ when $k_1 \neq k_2$.

Let W_N be an n_N by T matrix with row vectors

$$\left(\phi\left(x_j^N, y_1^{N+1}\right), \phi\left(x_j^N, y_2^{N+1}\right), \ldots, \phi\left(x_j^N, y_T^{N+1}\right)\right) \quad \text{for } j = 1, \ldots, n_N$$

and

$$v_{N+1} = \left(\mu^{N+1}\left(y_1^{N+1}\right), \ldots, \mu^{N+1}\left(y_T^{N+1}\right)\right)^{\mathrm{T}}$$

be a column vector. Also let G_N be a matrix with row vectors

$$\left(\phi\left(x_j^N, y_{r_1}^N\right), \ldots, \phi\left(x_j^N, y_{r_T}^N\right)\right) \quad \text{for } j = 1, \ldots, n_N,$$

and G'_N be a matrix with row vectors

$$\big(\phi\big(x_j^N, y_{t_1}^N\big), \ldots, \phi\big(x_j^N, y_{t_{m_N-T}}^N\big)\big) \quad \text{for } j = 1, \ldots, n_N.$$

A column vector v'_N is set by

$$v'_N = \big(\mu^N\big(y_{r_1}^N\big), \ldots, \mu^N\big(y_{r_T}^N\big), \mu^N\big(y_{t_1}^N\big), \ldots, \mu^N\big(y_{t_{m_N-T}}^N\big)\big)^{\mathrm{T}}.$$

In this way, (3.11) and (3.12) can be represented by

$$\big[W_N \mid G'_N\big]\begin{bmatrix} v_{N+1} \\ \hline 0 \end{bmatrix} = \big(g\big(x_1^N\big) + \varepsilon_1(\varepsilon), \ldots, g\big(x_{n_N}^N\big) + \varepsilon_{n_N}(\varepsilon)\big)^{\mathrm{T}} \tag{3.13}$$

and

$$\big[G_N \mid G'_N\big]v'_N = \big(g\big(x_1^N\big), \ldots, g\big(x_{n_N}^N\big)\big)^{\mathrm{T}}. \tag{3.14}$$

Because $H_N = [G_N \mid G'_N]$, our assumption says that $[G_N \mid G'_N]$ has rank m_N. Then from (3.9), we see that $[W_N \mid G'_N]$ has rank m_N and

$$\mu^N\big(y_i^N\big) < \varepsilon'_i(\varepsilon), \quad \forall i \in \{1, \ldots, m_N\} \setminus \{r_1, \ldots, r_T\}, \tag{3.15}$$

$$\big|\mu^N\big(y_{r_i}^N\big) - \mu^{N+1}\big(y_i^{N+1}\big)\big| < \varepsilon'_{r_i}(\varepsilon), \quad \forall i \in \{1, \ldots, T\}, \tag{3.16}$$

where $\varepsilon'_i(\varepsilon) \to 0$, for $i = 1, \ldots, m_N$, as $\varepsilon \to 0$. Noting that $\phi_{N+1}(x_{n_N+1}^N) = 0$ and $m_{N+1} \leqslant M'$, we have, by (3.15), and (3.16),

$$\phi_N\big(x_{n_N+1}^N\big) \to 0 \quad \text{as } \varepsilon \to 0. \tag{3.17}$$

But it was shown that $\phi_N(x_{n_N+1}^N) \to \phi_*(x_*) \neq 0$ (as $N \to \infty$). Therefore we have a contradiction which completes the proof. □

The technical conditions required in the convergence proof are directly related to the relaxation schemes adopted in the proposed algorithm. For example, in step 2 of the proposed algorithm, if μ^k and ν^k are required to be optimal solutions of (SIP_{T_k}) and ($DSIP_{T_k}$), respectively, then the second part of (ii) and (iii) of technical condition 2 can be removed. In step 3 of the proposed algorithm, if $\bar{\phi}_k(y) \leqslant 0$ is required, then technical condition 3 can be greatly simplified as "$\bar{\phi}_k(y) = 0$ is achieved only at $y_1^k, \ldots, y_{m_k}^k$".

When the proposed algorithm is applied to solve the general capacity problem (GCP), it stops in a finite number, say N^*, of iterations. Let the primal iterate at this iteration be a discrete measure μ^{N^*} concentrated at points $y_1^{N^*}, \ldots, y_{m_{N^*}}^{N^*}$ and the dual iterate be a discrete measure ν^{N^*} concentrated at $x_1^{N^*}, \ldots, x_{n_{N^*}}^{N^*}$. The following result tells how good μ^{N^*} can be as an approximate solution of ($GCAP$).

Theorem 3.5. Suppose that the proposed algorithm terminates at the N^*th iteration with μ^{N^*} and ν^{N^*}. Assume that $\bar{\mu}^{N^*} \in M^+(Y)$ is an optimal solution of ($SIP_{T_{N^*}}$) satisfying

$$\int_Y \phi(x, y)\,\mathrm{d}\bar{\mu}^{N^*}(y) - g(x) \geqslant -\delta, \quad \forall x \in X. \tag{3.18}$$

If (i) there exists a measure $\bar{\mu} \in M(Y)$ with $\bar{\mu}(y_i^{N^*}) \geqslant -\bar{\mu}^{N^*}(y_i^{N^*})/\delta$, for $i = 1, 2, \ldots, m_{N^*}$, and $\bar{\mu}$ is a nonnegative measure at $Y \setminus \{y_1^{N^*}, \ldots, y_{m_{N^*}}^{N^*}\}$ such that

$$\int_Y \phi(x, y)\,\mathrm{d}\bar{\mu}(y) \geqslant 1, \quad \forall x \in X, \tag{3.19}$$

and (ii) there exists a measure $\bar{\nu} \in M(X)$ with $\bar{\nu}(x_j^{N^*}) \geqslant -\nu^{N^*}(x_j^{N^*})/\delta$, for $j = 1, 2, \ldots, n_{N^*}$, and $\bar{\nu}$ is a nonnegative measure at $X \setminus \{x_1^{N^*}, \ldots, x_{n_{N^*}}^{N^*}\}$ such that

$$\int_Y \phi(x, y)\,\mathrm{d}\bar{\nu}(x) \leqslant -1, \quad \forall y \in Y, \tag{3.20}$$

then

$$\left|V\big(SIP_{T_{N^*}}\big(\mu^{N^*}\big)\big) - V(GCAP)\right| \leqslant \delta + \delta\left|\int_X g(x)\,\mathrm{d}\bar{\nu}(x)\right| + \delta\left|\int_Y f(y)\,\mathrm{d}\bar{\mu}(y)\right|.$$

Proof. Following (3.19), we have

$$\delta\int_Y \phi(x, y)\,\mathrm{d}\bar{\mu}(y) \geqslant \delta, \quad \forall x \in X. \tag{3.21}$$

By (3.18) and (3.21),

$$\int_Y \phi(x, y)\,\mathrm{d}\big(\bar{\mu}^{N^*} + \delta\bar{\mu}\big)(y) \geqslant g(x), \quad \forall x \in X. \tag{3.22}$$

Our assumptions imply that $\bar{\mu}^{N^*} + \delta\bar{\mu} \in M^+(Y)$ and it is a feasible solution for $(GCAP)$.

Therefore, we have

$$\begin{aligned}
&\left|V\big(SIP_{T_{N^*}}\big) - V\big(GCAP\big)\right| \\
&\leqslant \left|\int_Y f(y)\,\mathrm{d}\bar{\mu}^{N^*}(y) - \left(\int_Y f(y)\,\mathrm{d}\bar{\mu}^{N^*}(y) + \delta\int_Y f(y)\,\mathrm{d}\bar{\mu}(y)\right)\right| \\
&= \delta\left|\int_Y f(y)\,\mathrm{d}\bar{\mu}(y)\right|.
\end{aligned} \tag{3.23}$$

Similarly, from (2.5) and the definition of $(DSIP_{T_{N^*}})$,

$$\int_X \phi(x, y)\,\mathrm{d}\nu^{N^*}(x) - f(y) \leqslant \delta, \quad \forall y \in Y.$$

Again, from (3.20),

$$\delta\int_X \phi(x, y)\,\mathrm{d}\bar{\nu}(x) \leqslant -\delta, \quad \forall y \in Y.$$

Therefore,

$$\int_Y \phi(x, y)\,\mathrm{d}\big(\nu^{N^*} + \delta\bar{\nu}\big)(x) \leqslant f(y), \quad \forall y \in Y.$$

Our assumptions implies that $\nu^{N^*} + \delta\bar{\nu} \in M^+(Y)$. By (3), we know $\nu^{N^*} + \delta\bar{\nu}$ is a feasible solution of $(DSIP_{T_{N^*}})$. Note that

$$\begin{aligned} &\left|V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(GCAP)\right| \\ &\quad \leqslant \left|V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(SIP_{T_{N^*}})\right| + \left|V(SIP_{T_{N^*}}) - V(GCAP)\right| \qquad (3.24) \\ &\quad \leqslant \left|V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right)\right| + \left|V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right)\right. \\ &\qquad \left. - V(DSIP_{T_{N^*}})\right| + \left|V(SIP_{T_{N^*}}) - V(GCAP)\right|. \qquad (3.25) \end{aligned}$$

There are two possible cases to analyze, namely, $V(DSIP_{T_{N^*}}(\nu^{N^*})) \leqslant V(DSIP_{T_{N^*}})$ and $V(DSIP_{T_{N^*}}(\nu^{N^*})) > V(DSIP_{T_{N^*}})$.

For the first case, since μ^{N^*} is feasible to $(SIP_{T_{N^*}})$,

$$V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(DSIP_{T_{N^*}}) \geqslant 0.$$

Furthermore,

$$V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) \leqslant V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right) + \delta \leqslant V(DSIP_{T_{N^*}}) + \delta.$$

Thus,

$$0 \leqslant V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(DSIP_{T_{N^*}}) = V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(SIP_{T_{N^*}}) \leqslant \delta. \qquad (3.26)$$

Combining (3.24), (3.26) and (3.23), we see that

$$\left|V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(GCAP)\right| \leqslant \delta + \delta\left|\int_Y f(y)\,\mathrm{d}\bar{\mu}(y)\right|.$$

For the second case, we have

$$0 \leqslant V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right) \leqslant \delta,$$

and

$$V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right) > V(DSIP_{T_{N^*}}) \geqslant V\left(DSIP_{T_{N^*}}\left(\nu^{N^*} + \delta\bar{\nu}\right)\right),$$

since $(\nu^{N^*} + \delta\bar{\nu})$ is a feasible solution of $(DSIP_{T_{N^*}})$. Then

$$\begin{aligned} &\left|V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right) - V(DSIP_{T_{N^*}})\right| \\ &\quad \leqslant \left|V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right) - V\left(DSIP_{T_{N^*}}\left(\nu^{N^*} + \delta\bar{\nu}\right)\right)\right| \\ &\quad = \left|\int_X g(x)\,\mathrm{d}\nu^{N^*}(x) - \int_X g(x)\,\mathrm{d}(\nu^{N^*} + \delta\bar{\nu})(x)\right| = \delta\left|\int_X g(x)\,\mathrm{d}\bar{\nu}(x)\right|. \qquad (3.27) \end{aligned}$$

Combining (3.25), (3.23) and (3.27), we have

$$\begin{aligned} &\left|V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V(GCAP)\right| \leqslant \left|V\left(SIP_{T_{N^*}}\left(\mu^{N^*}\right)\right) - V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right)\right| \\ &\qquad + \left|V\left(DSIP_{T_{N^*}}\left(\nu^{N^*}\right)\right) - V(DSIP_{T_{N^*}})\right| + \left|V(DSIP_{T_{N^*}}) - V(GCAP)\right| \\ &\quad \leqslant \delta + \delta\left|\int_X g(x)\,\mathrm{d}\bar{\nu}(x)\right| + \delta\left|\int_Y f(y)\,\mathrm{d}\bar{\mu}(y)\right|. \end{aligned}$$

□

Note that the assumption (3.18) is in general achievable since after taking a large enough number of iterations, $\int_Y \phi(x, y)\,\mathrm{d}\bar{\mu}^{N^*}(y) - g(x)$ goes to zero.

4. A numerical example

The following example taken from page 149 of [1] is used to illustrate the computational behavior of the proposed algorithm:

$$(GCAP) \qquad \text{Minimize} \int_{-1}^{1} \mathrm{d}\mu(y) \tag{4.1}$$
$$\text{subject to} \quad \int_{-1}^{1} \big((y-x)^2 - 2\big)^2 \,\mathrm{d}\mu(y) \geqslant 1, \quad x \in [-1, 1],$$
$$\mu \in M^+\big([-1, 1]\big).$$

Its dual becomes

$$(DGCAP) \qquad \text{Maximize} \int_{-1}^{1} \mathrm{d}\nu(x) \tag{4.2}$$
$$\text{subject to} \quad \int_{-1}^{1} \big((y-x)^2 - 2\big)^2 \,\mathrm{d}\nu(y) \leqslant 1, \quad y \in [-1, 1],$$
$$\nu \in M^+\big([-1, 1]\big).$$

The optimal solution of the primal problem is given by a three-point measure $\mu(\cdot)$ with $\mu(-1) = 1/9$, $\mu(1) = 1/9$, and $\mu(0) = 2/9$. The dual optimal solution is given by $\nu(1/\sqrt{2}) = 2/9$ and $\nu(-1/\sqrt{2}) = 2/9$. The optimal objective value is $4/9 \approx 0.4444$.

We have implemented two algorithms for a simple comparison. The traditional cutting plane method is denoted by "Alg. 1" and the proposed algorithm by "Alg. 2". No matter using the traditional cutting plane method or the proposed algorithm, in the kth iteration, we face the following pair of linear semi-infinite programming problems:

$$(SIP_{T_k}) \qquad \text{Minimize} \int_{-1}^{1} \mathrm{d}\mu(y)$$
$$\text{subject to} \quad \int_{-1}^{1} \big((y - x_j^{k-1})^2 - 2\big)^2 \,\mathrm{d}\mu(y) \geqslant 1, \quad j = 1, \ldots, n_{k-1}+1,$$
$$\mu \in M^+\big([-1, 1]\big).$$

$$(DSIP_{T_k}) \qquad \text{Maximize} \sum_{j=1}^{n_{k-1}+1} v_j$$
$$\text{subject to} \quad \sum_{j=1}^{n_{k-1}+1} \big((y - x_j^{k-1})^2 - 2\big)^2 v_j \leqslant 1, \quad \forall y \in [-1, 1],$$
$$v_j \geqslant 0, \quad j = 1, \ldots, n_{k-1}+1.$$

Remember that for the traditional cutting plane method, $T_k = \{1, \ldots, k\}$.

To solve ($DSIP_{T_k}$) in both algorithms, we discretize the interval $[-1, 1]$ to 200 intervals and then solve a finite dimensional linear program with $n_{k-1} + 1$ nonnegative variables and 201 inequality constraints. To solve (SIP_k), first we find the vector y^k with y_i^k satisfying

$$\sum_{j=1}^{n_{k-1}+1} \left(\left(y_i^k - x_j^{k-1}\right)^2 - 2\right)^2 v_j^k = 1,$$

where $\{v_j^k\}$ is the optimal solution of ($DSIP_k$). A vector $\bar{x}^k$ is selected from the set $\{x_j^{k-1} \mid v_j^k \neq 0,\ j = 1, \ldots, n_{k-1}+1\}$. Then a linear system $M^k\mu = [1, \ldots, 1]^{\mathrm{T}}$ is solved to get the discrete measure of μ where M^k is a matrix with $M_{j,i}^k = ((y_i^k - \bar{x}_j^k)^2 - 2)^2$.

We use MATLAB version 5.0 [14] on a SUN Ultra 1 workstation as the test environment. In step 3 of Alg. 1, a minimizer $x_{n_k+1}^k$ of $\phi_k(x)$ is found by using the MATLAB function `fmin` and in step 4, the stopping criterion is set to be

$$\phi_k\left(x_{n_k+1}^k\right) > -0.0001. \tag{4.3}$$

For Alg. 2, we set $\delta = 0.02$ in step 3. There are different methods to find an $x_{n_k+1}^k$ such that $\phi(x_{n_k+1}^k) < -\delta$. We use a very simple implementation strategy by partitioning interval $[-1, 1]$ into 20 intervals and sequentially testing each end-point of the intervals. If the $\phi(\cdot)$ value is less then $-\delta$ at any of the end-points, we stop. Otherwise, $[-1, 1]$ is further partitioned into 200 intervals and then 2000 intervals. If all of end-points do not satisfy the inequality, we use MATLAB's `fmin` to get $x_{n_k+1}^k$. In our experiment, only in the last two iterations, this case happened and `fmin` was used. In this manner, even though $x_{n_k+1}^k$ obtained by this simple procedure is not the global minimizer of $\phi_k(x)$, it is still good enough to produce final results. In each iteration, technical conditions 1–3 are checked to validate the use of the proposed algorithm. The stopping criterion of Alg. 2 again uses (4.3). Alg. 2 is implemented by a C program linked by using `cmex` of MATLAB with `-O` compiler option.

Both algorithms were tested with two different initial point $x^1 = 0.5$ and $x^1 = 0$. The results are reported in table 1.

We set the maximal number of iterations to be $k = 30$. In our experiment, only the case of using Alg. 1 with $x^1 = 0$ exceeded this limit. The reason is related to the

Table 1
Numerical results with two different initial points.

	$x^1 = 0.5$		$x^1 = 0$	
	Alg. 1	Alg. 2	Alg. 1	Alg. 2
No. of iterations	10	10	30*	15
Objective value	0.444284	0.444343	0.439947	0.444343
Time spent in step 3	0.17 sec.	0.13 sec.	0.49 sec.	0.15 sec.

* Exceed the pre-set maximal number of iterations.

numerical error in calling `fmin` to find a global minimizer of $\phi_k(x)$. After the 10th iteration, the new $x^k_{n_k+1}$ is very close to those already in the set T_k. The algorithm was trapped thereafter. This also shows that, instead of finding a minimizer of $\phi_k(x)$, using $x^k_{n_k+1}$ with a relaxed condition of $\phi_k(x^k_{n_k+1}) < -\delta$ could be more stable.

The last row of table 1 shows the computational time spent in step 3 of both algorithms. The proposed algorithm has clearly reduced the computational effort required for finding a global minimizer of $\phi(\cdot)$ in each iteration. It is also interesting to report that, the number of constraints in (SIP_{T_k}) (i.e., $n_{k-1}+1$), of using the Alg. 2 with $x^1 = 0$ is

```
1, 2, 3, 3, 3, 4, 4, 4, 4, 4, 3, 4, 4, 4, 4,
```

in each iteration. Compared to solving (SIP_k) with k constraints in the kth iteration, the effectiveness of dropping unnecessary constraints in the proposed algorithm becomes significant.

5. Conclusion

In this paper, we have presented a relaxed cutting plane algorithm to solve the general capacity problem. Under proper technical conditions, it has been shown that the proposed algorithm indeed generates an approximate solution in a finite number of iterations. The quality of the approximate solution has also been studied.

Compared to the traditional cutting plane scheme, in each iteration, the proposed algorithm requires only inexact solutions to its subprograms and allows unnecessary constraints to be dropped to reduce the size of subprograms. It also relaxes the requirement of finding an optimizer of a potentially nonlinear and nonconvex program from one iteration to another. The potential of the proposed algorithm is illustrated by using one simple example.

References

[1] E.J. Anderson and P. Nash, *Linear Programming in Infinite-Dimensional Spaces* (Wiley, Chichester, 1987).

[2] G. Choquet, Theory of capacities, Annales de l'Institut Fourier 5 (1954) 131–295.

[3] S.-C. Fang, C.-J. Lin and S.-Y. Wu, Solving quadratic semi-infinite programming problems by using relaxed cutting plane scheme, Journal of Computational and Applied Mathematics (December 2000), to appear.

[4] S.-C. Fang and S.-Y. Wu, An inexact approach to solving linear semi-infinite programming problems, Optimization 28 (1994) 291–299.

[5] B. Fugled, On the theory of potentials in locally compact spaces, Acta Mathematica 103 (1960) 139–215.

[6] S.A. Gustafson and K.O. Kortanek, Numerical treatment of a class of semi-infinite programming problems, Naval Research Logistics Quarterly 20 (1973) 477–504.

[7] O. Hernandez-Lerma and J.B. Lasserre, Approximation schemes for infinite linear programs, SIAM J. Optimization 8 (1998) 973–988.

[8] R. Hettich and K.O. Kortanek, Semi-infinite programming: theory, method and applications, SIAM Review 35 (1993) 380–429.
[9] S. Karlin, *Mathematical Methods and Theory in Games, Programming and Economica* (Pergamon Press, London, 1959).
[10] H.-C. Lai and S.-Y. Wu, Extremal points and optimal solutions for general capacity problems, Mathematical Programming 54 (1992) 87–113.
[11] M. Ohtsuka, Generalized capacity and duality theorem in linear programming, Journal of Science, Hiroshima University Series A-I 30 (1966) 45–56.
[12] M. Ohtsuka, A generalization of duality theorem in the theory of linear programming, Journal of Science, Hiroshima University Series A-I 30 (1966) 31–39.
[13] R. Reemtsen and S. Gorner, Numerical methods for semi-infinite programming: a survey, in: *Semi-Infinite Programming*, eds. R. Reemtsen and J.-J. Ruckmann (Kluwer Academic, Netherlands, 1998).
[14] The MathWorks, Inc., *MATLAB User Guide* (Natick, MA, 1998).
[15] S.-Y. Wu, The general capacity problem, in: *Methods of Operations Research*, eds. W. Oettli et al. (Oelgeschlager, Gunn and Hain, Cambridge, MA, 1985).
[16] S.-Y. Wu, Linear programming in measure space, Ph.D. Dissertation, Cambridge University, UK (1985).
[17] S.-Y. Wu, S.-C. Fang, and C.-J. Lin, Relaxed cutting plane method for solving linear semi-infinite programming problems, Journal of Optimization Theory and Applications 99 (1998) 759–779.
[18] M. Yamasaki, On a capacity problem raised in connection with linear programming, Journal of Science, Hiroshima University Series A-I 30 (1966) 57–73.
[19] M. Yosida, Some examples related to duality theorem in linear programming, Journal of Science, Hiroshima University Series A-I 30 (1966) 41–43.