International Journal of Industrial Engineering and Operational Research (IJIEOR)

Contents lists available at IJIEOR
International Journal of Industrial Engineering and Operational Research
journal homepage: http://ijieor.ir
Volume 7, No. 1, 2025

A review of an efficient algorithm for solving linear semi-infinite optimization

Alireza Ataei [a], Fatemeh Daryaee [b]

[a] *Department of Mathematics, Faculty of Intelligent Systems Engineering and Data Science, Persian Gulf University, Bushehr, 7516913817, Iran*

[b]*Department of Education, Bushehr, 7551866897, Iran*

ARTICLE INFO

Received: 2024/08/24

Revised: 2024/10/19

Accept: 2025/01/12

Keywords:

Semi-infinite programming, Interior point algorithm Barrier function, Efficient algorithm, Interior point constraint generation algorithm.

ABSTRACT

In this paper, we try to introduce an interior point constraint generation (IPCG) algorithm for semi-infinite linear optimization that has a faster convergence than other algorithms. We will convert the semi-infinite optimization problem to the finite linear optimization problem using discretization, and we will convert the linear problem to the nonlinear problem using the barrier function. In each iteration, we find a point near the central path. We identify the finite number of the constraints violated at the mentioned point from the feasibility set of the semi-infinite problem. We update the feasible region and the barrier parameter simultaneously. We are improving the feasible point for the new feasible region and updating the central path; then, we use the Newton method to find a point near the new central path. We continue this process until the barrier parameter reaches the desired accuracy. Numerical results show that this algorithm has better speed and accuracy than others.

1. Introduction

In 1924, research on semi-infinite problems by Haar [11] began. From that year on, many algorithms were proposed to solve this problem [1,4,6] until, in 1997, Kalyski et al. [7] used a barrier decomposition algorithm for this kind of problem. This method is based on a cutting plane

---

[a] Corresponding author email address: ataei@pgu.ac.ir (Alireza Ataei).

and starts with a finite set of constraints. In the algorithm process, some of the constraints that interfere with the calculation of the optimal answer will be removed, on the other hand, some useful and effective constraints will be added. Less than a year later, Leo et al. [7] improved the algorithm by modifying the added constraints such that it will be created at most $O(\frac{m^6}{\delta^2}e^{\frac{4\sqrt{m}}{\varepsilon}})$ cuts where m is the number of variables, δ is the radius of the largest full dimensional ball contained in feasible region and only one cut will be added in each iteration. The algorithm that we will introduce is superior to its similar algorithm by storing more data and adding multiple cuts simultaneously such that at most $O(\frac{m^6\hat{p}^2}{\delta^2}e^{\frac{3\sqrt{m}}{\varepsilon}})$ cuts where $\hat{p}$ is the maximum number of constraints added simultaneously, will be added. This algorithm is basically a method based the analysis center cutting plane methods (ACCPM), introduced by Sonnevend [12, 13] and further developed by Goffin et al. [16], Ye [17], and Luo & Sun [18]. We will introduce this algorithm in the second part and in the third part we will compare the results of applying this algorithm with the results of the algorithm taken of [5].

2. Interior Point Constraint Generation (IPCG) Algorithm

In this section we present our IPCG algorithm for solving semi-infinite linear optimization. Consider the following optimization problem:

$$max\{ b^T y : a_t^T y \leq c_t , t \in \Omega \}, \tag{SILP}$$

where Ω is a compact set, $b \in \mathbb{R}^m$, $c_t \in \mathbb{R}$ and $a_t \in \mathbb{R} m$ for $t \in \Omega$.

The compressible feasible region of the above problem is as follows:

$$\mathcal{F}_\Omega = \{ y \in \mathbb{R}^m : a_t^T y \leq c_t , t \in \Omega \},$$

We make the following assumptions:

Assumption 1. The set Ω is compact, and the mappings $t \rightarrow a_t$ and $t \rightarrow b_t$ are continuous in t.

Assumption 2. The feasible region $\mathcal{F}_\Omega$ contains a δ-radius full-dimensional ball.

Assumption 3. $\mathcal{F}_\Omega$ is contained in the unit cube $[0, 1]^m$ and all m-vectors b and a_t are normalized ($\| a_t \| = \| b \| = 1$).

Consider semi-infinite linear programming (SILP), where the feasible region is an outer approximation of $\mathcal{F}_\Omega$ and is defined by a finite number of constraints:

$$max\{\, b^T y : A^T y \le c \,\}. \tag{D}$$

The above discretization of (D) is constructed by choosing n $(n \ge 2m)$ linear constraints from the constraint set $\mathcal{F}_\Omega$. In fact, for any subset $Q \subset \Omega$, we can define a corresponding discretization (or relaxation) of SILP by considering only those constraints indexed by Q. We shall use the simplified notation $A^{m\times n}$ to denote the $m \times n$ matrix whose $i$the vector gives h column a_i , with $i = 1, 2, \ldots, n$; Likewise, we use c n to signify the vector whose ith component is equal to c_i , for $i = 1, 2, \ldots, n$. Since $\mathcal{F}_\Omega$ is contained in the hypercube $[0, 1]^m$, we assume for convenience that the first $2m$ constraints chosen in (D) are of the form $0 \le y_i \le 1$, $i = 1, \ldots, m$. Thus, A^{mn} contains an $m \times m$ identity submatrix, therefore the matrix $A^{m\times n}$ has full row rank.

We denote the dual feasible region of the discretization problem:

$$\mathcal{F}_d = \{\, s \in \mathbb{R}^n : A^T y + s = c, \qquad s \ge 0 \,\}$$

Problem D is a relaxation of the original problem. Let us call it the dual problem. The corresponding primal problem reads.

$$min\{\, c^T x : Ax = b,\ x \ge 0 \,\}, \tag{P}$$

where

$$\mathcal{F}_p = \{x \in \mathbb{R}^n : Ax = b,\ x \ge 0\}.$$

Definition 1. A θ-approximate μ-center $(\bar{x}, \bar{s})$ is a point in the vicinity of the central path that satisfies

$$A^T \bar{y} + \bar{s} = c, \qquad A\bar{x} = b, \qquad \| \frac{\bar{x}\bar{s}}{\mu} - e \| \le 1$$

where $e \in \mathbb{R}^n$ is the vector with all its components equal to 1 and $\bar{x}\bar{s}$ is the Hadamard product of x and s.

We now state some technical lemmas needed in the IPCG algorithm. The proofs of those lemmas that are not given here can be found in the interior point methods book, such as [2].

Lemma 1. Let $(\bar{x}, \bar{s})$ be a θ-approximate μ-center.

Then:

$$n - \theta\sqrt{n} \leq \bar{x}^T\bar{s}\,\mu \leq n + \theta\sqrt{n}.$$

Moreover if $\mu^+ = (1 - \eta)\mu$ with $0 < \eta < 1$, then

$$\| \frac{\bar{x}\bar{s}}{\mu} - e \| \leq \frac{1}{1 - \eta}(\theta + \eta\sqrt{n}).$$

Corollary 1. For $n \geq 2, \eta = 1\ 8\sqrt{n}$, and arbitrary $\theta \leq 1/2$, one has

$$\| \frac{\bar{x}\bar{s}}{\mu^+} - e \| \leq 0.57.$$

We start from problem D with only an artificial box constraint with dynamically updated bounds (RHS). Let $\bar{y}$ be a point in the vicinity of the central path of $\mathcal{F}_d$ and $\bar{a}_j^T\ y \leq \bar{c}_j$, for $j = 1, 2, .., p$ be p constraints in $\mathcal{F}_\Omega$ such that $\bar{c}_j < \bar{a}_j^T\ \bar{y}$. The feasible region of the updated discretization, therefore reads as:

$$\mathcal{F}_d^+ = \{ s \in \mathbb{R}_+^n, \quad r \in \mathbb{R}_+^p : A^T y + s = c, \quad \bar{A}y + r = \bar{c} \},$$

$$\mathcal{F}_p^+ = \left\{ x \in \mathbb{R}_+^n, \quad t \in \mathbb{R}_+^p : Ax + \bar{A}t = b \right\},$$

where $\bar{A} \in \mathbb{R}^{p\times n}$ is composed of the p column vectors $\bar{a}_i$ s and$\bar{c} = (\bar{c}_1; \bar{c}_2; \ldots; \bar{c}_p)$. Let $\mu^+ = (1 - \eta)\mu$ be the updated barrier parameter for a later-specified value $0 < \eta < 1$. Now, we have to find a point in the vicinity of the central path of the updated discretization, close to the μ^+-center of $\mathcal{F}_d^+$. Before that, we first need to derive a strictly feasible point for $\mathcal{F}_d^+$ [14-16].

Let:

$$\bar{t} = \arg\min \{\tfrac{p}{2}\, t^T\, V\, t - \textstyle\sum_{i=1}^{p} \log t_i \}, \tag{1}$$

Where $V = \bar{A}^T\ (A\,\bar{X}A^T\ )^{-1}\,\bar{A},\ \bar{X}$ is a diagonal $n \times n$ matrix with the components of vector $\bar{x}$ as its diagonal elements.

Also define:

$$\bar{d} = p(\bar{A}^T\,\bar{y} - \bar{c})\bar{t}. \tag{2}$$

Such that $\bar{d} > 0$.

Let $\alpha < 1 - \theta$ be fixed. We consider two cases:

1. **Moderately deep constraints:** $\bar{d} < \alpha e$. In this case we show that all violated constraints cross the Dikin ellipsoid around $\bar{y}$, and the dual feasibility can be recovered using the current point $\bar{y}$.

2. **Very deep constraints:** There exists a constraint for which $\bar{d}_i \geq \alpha$. In this case, dual feasibility cannot be recovered, but primal feasibility can be recovered.

Below, Lemma shows that if the violated constraints are moderately deep, then Newton's method can be initiated from x^+ and s^+ to obtain a point in the vicinity of the new central path. When at least one very deep inequality exists, dual feasibility cannot be recovered because it is unclear how far the constraint is away from the Dikin ellipsoid. In this situation, one can still recover primal feasibility by using x^+, and Newton's method can be applied in the primal space to update the μ^+-center. This procedure is repeated until the barrier parameter falls within the desired accuracy.

Lemma 2. Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be the primal and dual feasible regions of the discretization problem, respectively. Let μ be the barrier parameter, and $(\bar{x}, \bar{s})$ be a point in the vicinity of the central path. Let p violated constraints be added to $\mathcal{F}_d^+$ and $\Delta x = -\bar{X}^2 A^T (A\bar{X}^2 A^T)^{-1} \bar{A}\bar{t}$,Then $x^+ = (\bar{x} + \alpha\Delta x;\ \alpha\bar{t})$ is strictly feasible for $\mathcal{F}_d^+$. Furthermore, define $\Delta s = A^T (A\bar{X}^2 A^T)^{-1} \bar{A}\bar{t}$ and $\bar{t} = 1/p\big(\alpha e - \bar{d}\big)\bar{t}^{-1}$ where the p-vector $\bar{t}^{-1}$ is the component-wise inverse of the vector $\bar{t}$. Then $s^+ = (\bar{s} + \alpha\Delta s;\ \bar{r})$ is strictly feasible for $\mathcal{F}_d^+$.

Proof. Let's put x^+ in $\mathcal{F}_p$:

$$A(\bar{x} + \alpha x) + \alpha\bar{A}\bar{t} = A\bar{x} + \alpha\bar{A}\bar{t}$$

Now by putting Δx in $A\Delta x + \bar{x}\bar{t}$, we will have:

$$\begin{aligned} A(\bar{x} + \alpha\Delta x) + \alpha\bar{A}\bar{t} &= A\bar{x} - \alpha A\bar{X}^2 A^T (A\bar{X}^2 A^T)^{-1} \bar{A}\bar{t} + \alpha\bar{A}\bar{t}, \\ &= A\bar{x} - \alpha\bar{A}\bar{t} + \alpha\bar{A}\bar{t}, \\ &= A\bar{x}, \\ &= b \end{aligned}$$

By the fact $\alpha < 1 - \theta$, one can show that $\bar{x} + \alpha\Delta x > 0$ and $\bar{s} + \alpha\Delta s > 0$. now, we will proof $\bar{r} > 0$ and $\bar{A}^T y^+ + \bar{r} = \bar{c}$. Because $\bar{d} < \alpha e$, so $\bar{r} > 0$. Notice because $A^T (\bar{y} + \Delta y) + \bar{s} + \Delta s = c$ we can say $A^T \Delta y = -\Delta s$, so $\Delta y = -(A\bar{X} - 2A^T)\bar{A}\bar{t}$. Then:

$$\bar{A}^T y^+ r^+ = \bar{A}^T \bar{y} + \alpha \bar{A}^T \Delta y + \bar{r} = \bar{A}^T \bar{y} - \alpha V \bar{t} + \bar{t}.$$

If we run KKT condition for (1), we can write:

$$\begin{aligned} \bar{A}^T y + \bar{r} &= \bar{A}^T \bar{y} - \alpha\, p\, \bar{t}^{-1} + \frac{1}{p}\left(\alpha e - \bar{d}\right)\bar{t}^{-1}, \\ &= \bar{A}^T \bar{y} - \frac{1}{p}\ \bar{d}\bar{t}^{-1}, \\ &= \bar{c} \end{aligned}$$

The next lemma, from Luo et al. [9], shows that the constraint generation algorithm delivers a ϵ-optimal solution for SILP.

Lemma 3. Let $\epsilon > 0$ be ginven. Under Assumption 1, if $\bar{y} \in \mathcal{F}_\Omega$ is in the vicinity of $\mu < \epsilon/(n + \sqrt{n})$, then $\bar{y}$ is an ϵ-maximizer of SILP.

According to what has been said, the algorithm is as follows:

```
Algorithm 1 interior point constraint generation Algorithm.
Require: $\mathcal{F}_d^0 = [0,1]^m$ , $y^0 = \frac{1}{2} e$, $s^0 = \frac{1}{2} e$, $n^0 = 2m$, $\eta^0 = \frac{1}{8}\sqrt{n_0}$ , $\theta = 0.5$, $k = 1$
Ensure: $x^k$     $s^k$
while $( n_{k-1} + \sqrt{n_{k-1}} )\mu_{k-1} \geq \varepsilon$ do
    Identify $p_k$ violated constraints in $\mathcal{F}_\Omega$ that $y^k$ is not feasible.
        $(\bar{A}^k)^T y \leq \bar{c}^k$ in $\mathcal{F}_\Omega$ such that $(\bar{A}^k)^T y^k > \bar{c}^k$
    Update
        $n_k = n_{k-1} + p_k$,    $\eta_k = \frac{1}{8}\sqrt{n_k}$ ,    $\mu_k = (1 - \eta_k)\mu_{k-1}$
        $A^k = [\, A^{k-1}\ \ \bar{A}^k \,]$ , $c^k = (c^{k-1} ; \bar{c}^k)$
    Compute $\bar{t}$ from (1) and $\bar{d}$ from (2)
    if $\bar{d} < \alpha e$ then
        use $s^+$ to start a primal newton method to obtain $s^k$ and define $x^k = x(s^k)$ in the vicinity of the $\mu_k$-center of $\mathcal{F}_d^k$ .
    else
```

```
            use $x^+$ to start a primal Newton method to obtain $x^k$ and define $s^k = s(x^k)$ in the vicinity of the $\mu_k$-center of $\mathcal{F}_p^k$ .
        end if
        k = k + 1
end while
```

3. Comparison of two algorithms of interior point constraint generation and projected Lagrangian

In this section, we have solved some examples using the introduced algorithm. We have run the examples with MATLAB 2019. We took example 1 from [10], example 2 from [8], and example 3 from [6].

Example 1.

$$max \sum_{i=1}^{10} r_{2i-1} x_i$$

$$s.t.\ \ 2\sum_{i=1}^{20} cos\big((2i-1)\pi t\big) x_i \geq -1$$

$$\Omega \in [0, 0.5]$$

$$where\ r_i = 0.95^i$$

Example 2.

$$max\ -\sum_{i=1}^{10} i^{-1} x_i$$

$$s.t:$$

$$\sum_{i=1}^{8} t^{i-1} x_i \geq \frac{1}{2-t}$$

$$\Omega \in [0, 1]$$

Example 3.

$$max\ -\sum_{i=1}^{9} i^{-1} x_i$$

$$s.t:$$

$$\sum_{i=1}^{9} i^{i-1} x_i$$

$$\Omega \in [0,1]$$

Numerical results are presented in Table 1.

Table 1: Results of IPCG algorithms

Example	Objective value	CPU time
1	-0.5105165	19.7
2	-0.6655548	3.2
3	-0.7325134	5.8

The computational results are reported in Table 1. Comparing the results in Table 1 and Table 2 reveals that the algorithm mentioned is more efficient than the algorithm presented in [5].

Table 2: Results of [5]

Example	Objective value	CPU time
1	-0.5105203821	27.41
2	-0.6655548	5.94
3	-0.7325144332	6.23

4. Conclusion

This paper introduces an algorithm for solving semi-infinite linear programming problems. Numerical results show interior point constraint generation algorithms have a higher speed and accuracy than others [16-19]. The tolerance considered in this paper is $\varepsilon = 10^{-8}$.

References

[1] Coope I. D., Watson G. A. (1985). A projected Lagrangian algorithm for semi-infinite programming, Math. Programming 32, 337-356.

[2] Hertog D. (1994). Interior Point Approach to Linear, Quadratic, and Convex Programming, Kluwer Academic Publishers, Dordrecht, The Netherlands

[3] Glashoff K., Gustafson S.A. (1983). Linear Optimization and Approximation, Springer-Verlag, Berlin.

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

[5] Fakharzadeh J. A. R., Khosravi A. (2015). A hybrid method for solving fuzzy semi-infinite linear programming problems, Journal of Intelligent Fuzzy Systems 28, 879884.

[6] Ferris M. C., Philpott A. B. (1989). An interior point algorithm for semi-definite linear programming. Math. Programming Ser. A 43(3), 257276.

[7] Kaliski J., Haglin D., Roos C. and Terlaky T. (1997). A logarithmic barrier decomposition method for semi-infinite programming, Internat. Trans. Oper. Res. 4 (4), 285-303.

[8] Leon T. and Vercher E. (1992). A purification algorithm for semi-infinite programming, European Journal of Operational Research 57, 412420.

[9] Luo Z. Q., Roos C., Terlaky T. (1999). Complexity analysis of logarithmic barrier decomposition method for semi-infinite linear programming, Applied Numerical Mathematics 29, 379394.

[10] Moulin P., Anitescu M., Kortanek K.O. and Potra F.A. (1997). The role of linear semi-infinite programming in signal adapted QMF bank design, IEEE Trans Signal Processing 45**,** 21602174.

[11] Haar, D. T. (1924). Onze Vlinders: 149-151.

[12] Mahdavimanshadi, M., Anaraki, M. G., Mowlai, M., & Ahmadirad, Z. (2024, May). A Multistage Stochastic Optimization Model for Resilient Pharmaceutical Supply Chain in COVID-19 Pandemic Based on Patient Group Priority. In 2024 Systems and Information Engineering Design Symposium (SIEDS) (pp. 382-387). IEEE.

[13] Sonnevend, G. 1988. New algorithms in convex programming based on a notation of center and on rational extrapolations. International Series of Numerical Mathematics, Vol. 84. Birkhauser Verlag, Basel, Switzerland, 311–327.

[14] Salehi, S., Ghatrehsamani, S., Akbarzadeh, S. et al. Application of Continuum Damage Mechanics for Prediction of Wear with Provision for Sequential Speed Operation. Tribol Lett 70, 105 (2022). https://doi.org/10.1007/s11249-022-01645-9

[15] Salehi, S., Ghatrehsamani, S., Akbarzadeh, S. et al. Prediction of Wear in Start–Stop Systems Using Continuum Damage Mechanics. Tribol Lett 73, 11 (2025). https://doi.org/10.1007/s11249-024-01943-4

[16] Ghatrehsamani, S., Salehi, S., Akbarzadeh, S., & Khonsari, M. M. (2023). On the wear of coated surfaces under variable speed. Tribology International, 187, 108677.

[17] Goffin, J.-L., & Vial, J.-P. (2000). Multiple Cuts in the Analytic Center Cutting Plane Method. SIAM Journal on Optimization, 11(1), 266–288

[18] Ye, Y. 1997. Complexity analysis of the analytic center cutting plane method that uses multiple cuts. Math. Programming, Ser. A 78(1) 85–104.

[19] Luo, Z.-Q., J. Sun. 2000. A polynomial cutting surfaces algorithm for the convex feasibility problem defined by self-concordant inequalities. Comput. Optim. Appl. 15(2) 167–191.