Hindawi
Journal of Mathematics
Volume 2022, Article ID 1765385, 14 pages
https://doi.org/10.1155/2022/1765385

Research Article

An Exact Algorithm Based on the Kuhn–Tucker Conditions for Solving Linear Generalized Semi-Infinite Programming Problems

Abraham Barragán and José-Fernando Camacho-Vallejo

Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico-Matemáticas, Av. Universidad s/n,
San Nicolás de Los Garza CP 66450, Nuevo León, Mexico

Correspondence should be addressed to José-Fernando Camacho-Vallejo; jose.camachovl@uanl.edu.mx

Received 22 September 2021; Revised 1 February 2022; Accepted 28 February 2022; Published 25 April 2022

Academic Editor: Ram Jiwari

Optimization problems containing a finite number of variables and an infinite number of constraints are called semi-infinite programming problems. Under certain conditions, a class of these problems can be represented as bi-level programming problems. Bi-level problems are a particular class of optimization problems, in which there is another optimization problem embedded in one of the constraints. We reformulate a semi-infinite problem into a bi-level problem and then into a single-level nonlinear one by using the Kuhn–Tucker optimality conditions. The resulting reformulation allows us to employ a branch and bound scheme to optimally solve the problem. Computational experimentation over well-known instances shows the effectiveness of the developed method concluding that it is able to effectively solve linear semi-infinite programming problems. Additionally, some linear bi-level problems from literature are used to validate the robustness of the proposed algorithm.

1. Introduction

The more general case of semi-infinite programming (SIP) problems occurs when a finite number of semi-infinite constraints are considered. Each constraint consists in a set of constraints indexed by a set of infinite cardinality, in this case, a generalized semi-infinite programming (GSIP) problem arises. Previous work has been reported for problems that involve continuous and differentiable functions [1–3]. Therefore, linear generalized semi-infinite programming (LGSIP) problems can be considered as a particular case of GSIPs, where all the defining functions are sufficiently smooth. In [4], an inverse optimization approach is proposed to solve a LGSIP. In general, semi-infinite programming is a very challenging field and keeps attracting the attention of researchers. For instance, a review of solution methods and different applications appear in [5]. Similarly, GSIPs have been proposed to model a wide variety of real-life applications, such as design centering, optimization of assembly lines, time minimal control, and robust optimization [6].

A basic reference on this topic is [7], and a review of classical methods in GSIP appears in [8]. When the set $Y(\mathbf{x}) = Y$ does not depend on the variable $\mathbf{x}$, we have a standard semi-infinite programming (SSIP) problem which is another especial case in GSIP. Regarding the latter problems, in [9], an extensive study on their main characteristics is presented, and, in [10, 11], some algorithmic aspects are reviewed. Recent results for a linear case are presented in [12–15]. For a wide overview of SSIP applications, we refer to [7]. Some applications of GSIP can be found in the maneuverability problem in robotics (see [16]) and the reverse Chebyshev approximation (see [17, 18]).

Theory and solution methods for SIPs are strongly related to their corresponding bi-level formulation. The connection between both classes of problems relies in the fact that, under certain conditions, semi-infinite problems can be reformulated as bi-level programming ones [1, 3]. Bi-level programming (BLP) problems have been used to model applications widely, for example, in location, network design, pricing, humanitarian logistics, and telecommunication systems, among others [19, 20]. BLP can be seen as a hierarchized situation in which the decision maker with a higher hierarchy (leader) aims to optimize his/her own objective function depending on its decisions and the decisions taken by the decision maker with a lower hierarchy (follower). In this case, the follower considers his/her own optimization problem but parameterized according to the leader's decision. It is essential to mention that BLP cannot be seen as two-stage optimization problems, since the leader's decision restricts the follower's problem, and then, the follower's decision impacts the leader's objective function. Hence, both decisions are interrelated but respecting a predefined hierarchy among the decision makers.

As it is mentioned above, there exist few methods for solving generalized semi-infinite optimization that exploit the closely relationship obtained from the bi-level structure of the problems [8]. Despite the abovementioned, the proposed methods that solve GSIPs are scarce and the computational experimentations are limited. The latter motivated us to conduct this research. Hence, in order to benefit from the relationship between GSIPs and BLPs, the following idea could be followed: reformulate the lower-level problem by some of the well-known methods. For instance, in the convex case, the Kuhn–Tucker optimality conditions are commonly used. After that, some of the classical algorithms used to solve linear bi-level problems, such as the *K*th-best algorithm, Kuhn–Tucker approach, complementarity approach, variable elimination algorithm, and penalty function approach [21, 22], could be implemented to solve the reformulated GSIP. In this case, we decided to implement a different exact approach, which is based on the branch and bound method, which have been used to successfully solve nonlinear SIPs (see [23]).

Other methods that have been used to solve GSIPs via the bi-level formulation differ from each other on the lower-level reformulations. For instance, Stein [2] uses equilibrium constraints to model the lower level. In [8], complementarity constraints are employed to replace the lower level. Numerical methods to approximate the original problem are generated by substituting these complementarity constraints by smooth ones [3, 24]. That method has been applied to solve centering design and lapidary cutting problems (see [1, 25], respectively). Another approach is to reformulate the lower level by its corresponding dual problem. In [24], Wolfe's dual problem is considered. It is shown that, under additional conditions, the resulting problem is numerically tractable. Theoretically, feasible direction methods and discretization algorithms [6] have been also proposed to solve GSIPs. However, the computational experimentation is complicated when having multidimensional set of indices. Most of these methods are developed to solve particular class of GSIPs or made significant assumptions preventing their reproducibility, in general. On the contrary, the algorithm herein proposed is motivated by the ideas given in [3] for solving a general case of semi-infinite programming problems but neither a detailed description nor results are reported there. Hence, a detailed step-by-step description of the algorithm is included to facilitate its implementation and reproducibility.

The main contribution of this research is the proposal of an algorithm that can be used to solve a particular class of generalized semi-infinite programming problems. In particular, linear problems consider a semi-infinite constraint. The proposed method exploits the conditions that allow a generalized semi-infinite problem to be handled as a bi-level programming one. In order to apply this method, the representation of the semi-infinite problem as a bi-level one is required. Then, its Kuhn–Tucker optimality conditions are used to reformulate the bi-level problem into a single-level one, which is solved via a branch-and-bound scheme. The effectiveness of the algorithm is shown through computational experimentation. To validate the obtained results, we analytically solve the tested linear semi-infinite programming problems, showing the algebraic and graphical solutions coincide with the solution obtained by the algorithm.

The remainder of this study is organized as follows. In Section 2, some preliminaries and the notation considered throughout this research is presented. In Section 3, the developed algorithm is described. Therefore, the computational experimentation, in which the effectiveness of the algorithm is validated, is presented in Section 4. Additionally, some examples are solved in a geometric and algebraic way to corroborate the obtained solution. Conclusions that include important remarks regarding the topic investigated in this study and further research directions are presented in Section 5.

2. Preliminaries

Let us introduce the necessary notation for this study, as well as some concepts, and necessary results on GSIP and BLP that will be used.

Throughout this study, we consider $\{\mathbf{a}, \mathbf{c}, \mathbf{c}_1, \mathbf{c}_2\} \subset \mathbb{R}^n$, $\{\mathbf{b}, \mathbf{d}_1, \mathbf{d}_2\} \subset \mathbb{R}^m$, $b_0 \in \mathbb{R}$, $\mathbf{b}_1 \in \mathbb{R}^p$, $\mathbf{b}_2 \in \mathbb{R}^q$, A_1 is a $p\times n$-matrix, B_1 is a $p \times m$-matrix, A_2 is a $q \times n$-matrix, and B_2 is a $q \times m$-matrix.

We consider linear generalized semi-infinite programming (LGSIP) problems which are defined as

$$\begin{aligned} \text{SIP: } & \min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \\ & \text{s.t. } \mathbf{a}^T \mathbf{x} + \mathbf{b}^T \mathbf{y} \le b_0, \text{for all } \mathbf{y} \text{ in } Y(\mathbf{x}), \end{aligned} \tag{1}$$

where

$$Y(\mathbf{x}) = \{\mathbf{y} | A_2 \mathbf{x} + B_2 \mathbf{y} \le \mathbf{b}_2\}. \tag{2}$$

Unless otherwise stated, we will consider $\mathbf{x} \in X$ and $\mathbf{y} \in Y_0$, where the sets X and Y_0 will represent additional restrictions on the decision variables. In particular, $X \subseteq \mathbb{R}^n_+$ and $Y_0 \subseteq \mathbb{R}^m_+$, that is, $x_i \ge 0$ and $y_j \ge 0$, for all $i = 1, \dots, n$ and $j = 1, \dots, m$, respectively. Furthermore, we will consider that $X = do\, mY$, where $\operatorname{dom} Y = \{\mathbf{x} | Y(\mathbf{x}) \ne \emptyset\}$; this condition can always be satisfied by adding in X appropriate extra conditions (such as $x_i \le \delta$ or others depending only on the variables x_i).

Following the notation presented in [21], a linear bi-level programming (LBLP) problem is defined as

$$
\begin{aligned}
&\text{BL: } \min_{\mathbf{x}} \mathbf{c}_1^T\mathbf{x} + \mathbf{d}_1^T\mathbf{y},\\
&\text{s.t. } A_1\mathbf{x} + B_1\mathbf{y} \le b_1,\\
&\text{and } \mathbf{y} \text{ solves},\\
&Q(\mathbf{x})\text{: } \min_{\mathbf{y}} \mathbf{c}_1^T\mathbf{x} + \mathbf{d}_1^T\mathbf{y},\\
&\text{s.t. } \mathbf{y} \in Y(\mathbf{x}),
\end{aligned}
\tag{3}
$$

where $Y(\mathbf{x})$ is defined as in *SIP*.

With the conditions on X, the problem SIP can be considered as a LBLP problem with the special structure that lower-level objective function is obtained with the upper-level constraint. The problem that results is presented next (see [3]):

$$
\begin{aligned}
&\text{BL}_{SIP}\text{: } \min_{\mathbf{x}} \mathbf{c}^T\mathbf{x},\\
&\text{s.t. } \mathbf{a}^T\mathbf{x} + b^T\mathbf{y} \le b_0,\\
&\text{and } \mathbf{y} \text{ solves},\\
&Q(\mathbf{x})\text{: } \min_{\mathbf{x}} -\mathbf{a}^T\mathbf{x} - \mathbf{b}^T\mathbf{y},\\
&s.t.\ \mathbf{y} \in Y(\mathbf{x}).
\end{aligned}
\tag{4}
$$

In order to have well-defined considered problem, we assume that the set-valued mapping $Y\colon \mathbb{R}^n \longrightarrow 2^{\mathbb{R}^m}$ is uniformly compact on $\mathbb{R}^n$. In other words, there exists a ball $B_\varepsilon(\overline{x}) = \{\mathbf{x} \in \mathbb{R}^n \,|\mathbf{x} - \overline{x} \le \varepsilon\}$ with $\varepsilon > 0$ for any $\overline{\mathbf{x}} \in \mathbb{R}^n$ such that $cl\,(\cup_{B_\varepsilon(\overline{\mathbf{x}})} Y(\mathbf{x}))$ is compact, where $cl\,X$ denotes the topological closure of X. Therefore, under these assumptions, the mapping Y is closed and upper semicontinuous in the sense of Berge, and X is a closed set. It is important to remark that, by considering the latter condition, the lower-level problems $Q(\mathbf{x})$, $\mathbf{x} \in X$, always has an optimal solution.

The following sets are associated with a bi-level problem. Let $S(\mathbf{x}) = \{\mathbf{y}|\mathbf{y} \text{ is a solution global of } Q(\mathbf{x})\}$ be the set of solutions of $Q(\mathbf{x})$. Consider $S = \{(\mathbf{x},\mathbf{y})|\mathbf{y} \in S(\mathbf{x})\}$ as the solution graph of Q, and $G = \{(\mathbf{x},\mathbf{y}) \,|\, \mathbf{a}^T\mathbf{x} + \mathbf{b}^T\mathbf{y} \le b_0\}$ as the constrained set for the upper level. Also, let $M_{BL} = \{(\mathbf{x},\mathbf{y})|\,(\mathbf{x},\mathbf{y}) \in G, \mathbf{y} \in S(\mathbf{x})\}$ be the feasible set of *BLP*.

Now, the concepts associated with the semi-infinite problem defined in the introduction are presented. First, its feasible set is defined as follows:

$$
M_{SIP} = \{\mathbf{x} \in X|\mathbf{a}^T\mathbf{x} + \mathbf{b}^T\mathbf{y} \le b_0, \mathbf{y} \in Y(\mathbf{x})\}. \tag{5}
$$

Under the assumptions made on the set X, we have (see [3]) $M_{SIP} = \text{pr}_{\mathbf{x}}(M_{BL_{SIP}})$, where $pr_{\mathbf{x}}$ denotes the orthogonal projection onto the space $\mathbb{R}^n$.

The latter equality allows the use of methods proposed to solve LBLP problems to obtain the solution of LGSIP problems.

As it is mentioned in the introduction, the main idea of the algorithms designed to solve linear bi-level problems is to reformulate the bi-level problem into an equivalent single-level one. The latter can be achieved by using the important result described next.

Consider $\mathbf{u} \in \mathbb{R}^q$ and $\mathbf{v} \in \mathbb{R}^m$ as the corresponding dual variables of the constraints associated with the lower-level problem, that is, constraints represented by equation (3). Using the notation of problem BL_{SIP}, we adapt the following proposition given in [21].

Proposition 1. *A necessary condition for* $(\mathbf{x}^*, \mathbf{y}^*)$ *to solve* BL_{SIP} *is that there exist (column) vectors* $\mathbf{u}^*$ *and* $\mathbf{v}^*$ *such that* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*, \mathbf{v}^*)$ *solves*

$$
\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x}, \tag{6}
$$

$$
\text{s.t. } \mathbf{a}^T\mathbf{x} + \mathbf{b}^T\mathbf{y} \le b_0, \tag{7}
$$

$$
\mathbf{u}^T B_2 - \mathbf{v} = \mathbf{b}, \tag{8}
$$

$$
\mathbf{u}^T(\mathbf{b}_2 - A_2\mathbf{x} - B_2\mathbf{y}) + \mathbf{v}^T\mathbf{y} = 0, \tag{9}
$$

$$
A_2\mathbf{x} + B_2\mathbf{y} \le \mathbf{b}_2, \tag{10}
$$

$$
\mathbf{x} \ge 0, \mathbf{y} \ge 0, \mathbf{u} \ge 0, \mathbf{v} \ge 0. \tag{11}
$$

This single-level reformulation is valid in bi-level programming when the lower-level problem is convex. Hence, it not always exists for all the classes of BLP problems. However, in this study, we are considering the linear case, so the problems can be reformulated under this scheme.

3. An Algorithm that Uses the Kuhn–Tucker Optimality Conditions and a Branch-and-Bound Scheme for Solving LGSIPs

It is well known that one of the most common approaches to solve a linear bi-level problem is by using its equivalent single-level reformulation, which is given in equation (5)–(10). It can be appreciated that the resulting single-level model is a nonlinear one (see equation (9)), which complicates the optimization process. Noting that equation (9) is the only nonlinear constraint, it is common to try to handle the reformulation without that equation instead of solving the reformulated nonlinear single-level model.

One option to address the nonlinearity of the reformulated model is by omitting the nonlinear term (this results in a relaxation of the problem) and solve the resulting relaxed problem. For achieving this, let us consider all the inequalities that appear in the lower-level problem in the following manner: $g_i(\mathbf{x},\mathbf{y}) \ge 0, i = 1, \ldots, q + m$. Also, it is identified that the nonlinear constraints $u_i g_i(\mathbf{x},\mathbf{y}) = 0$ $(i = 1, \ldots, q + m$, where $u_{q+j} = v_j$, $j = 1, \ldots, m)$ imply the complementarity slackness constraints. Recall that the relaxed reformulation yields to a linear subproblem.

After the resulting linear subproblem has been solved, it is checked if equation (9) holds. If so, the optimal solution obtained is a feasible solution of problem given by equation (5)–(10), which implies that it is a feasible solution for the bi-level problem derived from the SIP; if not, an approach based on a branch and bound scheme is implemented to try to satisfy the complementarity slackness constraints. To achieve the latter, an examination of all the possible combinations' values that guarantee the complementarity slackness implicitly is conducted. A complete cycle of this procedure is called an iteration.

The notation introduced in [21], which is the classical notation for an implicit enumeration, is used to describe the solution method herein considered. Let $W = \{1, \ldots, q+m\}$ be the set of indexes associated with the complementarity slackness terms (see equation (9)). Consider $F(\cdot)$ as the objective function of the single-level reformulation, which corresponds to the upper-level's objective function of the bi-level problem, and consider $\overline{F}$ as the incumbent upper bound on the upper-level's objective function. A subset of indexes $W_k \subset W$ is considered at the kth level of the search tree, and also, a path P_k is used. The path indicates if either $u_i = 0$ or $g_i = 0$, for $i \in W_k$. The common sets S_k^+, S_k^-, and S_k^0 of the implicit branch and bound are used. Therefore,

$$\begin{aligned} S_k^+ &= \{i \,|\, i \in W_k \text{ and } u_i = 0\}, \\ S_k^- &= \{i \,|\, i \in W_k \text{ and } g_i = 0\}, \\ S_k^0 &= \{i | i \notin W_k\}. \end{aligned} \tag{12}$$

The variables u_i and g_i, for which $i \in S_k^0$, are considered as nonnegative when solving the relaxation of the single-level reformulated problem defined by equations (6)–(11). Recall that, in the aforementioned relaxation, equation (9) is omitted. Hence, the complementarity slackness constraint may be satisfied or not.

The description of the algorithm based on a branch-and-bound scheme that considers the complementarity slackness constraints is presented next. It is important to emphasize that this algorithm is based in the Kuhn–Tucker approach for solving linear bi-level problems (see [21]). In nonlinear problems, Kuhn–Tucker conditions are used in globalization [26].

(i) Initialization: set $k = 0$, $S_k^+ = \varnothing$, $S_k^- = \varnothing$, $S_k^0 = \{1, \ldots, q+m\}$, and $\overline{F} = \infty$.

(ii) Main step (in the **k**th iteration): set $u_i = 0$, for $i \in S_k^+$, and $g_i = 0$, for $i \in S_k^-$. Try to solve the relaxed single-level problem, that is, the problem defined by problem defined in equations (6)–(11) omitting equation (9). If an optimal solution is obtained, do $k \leftarrow k+1$, record the solution $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{u}^k)$, and continue to the Fathom step; otherwise, go to the Backtrack step.

Fathom: if the obtained solution yields an objective function value worst that the incumbent, a backtrack is needed. That is, if $F(\mathbf{x}^k, \mathbf{y}^k) \geq \overline{F}$, go to the Backtrack step; otherwise, continue to the Branch step.

Branch: if $u_i^k g_i(\mathbf{x}^k, \mathbf{y}^k) = 0$, $i = 1, \ldots, q+m$, go to the next step; otherwise, select the ith for which the product $u_i^k g_i(\mathbf{x}^k, \mathbf{y}^k)$ is the largest one and label it as i_1. Update the sets as follows: $S_k^+ \leftarrow S_k^+ \bigcup \{i_1\}$, $S_k^0 \leftarrow S_k^0 \setminus \{i_1\}$, and $S_k^- \leftarrow S_k^-$. Append i_1 to the current path P_k, and return to the main step.

Update: set the incumbent objective function value as $\overline{F} = F(\mathbf{x}^k, \mathbf{y}^k)$.

Backtrack: if all the nodes in the current path P_k has been explored already, go to the next step; otherwise, branch to the ith component most recently introduced in the path P_k and update S_k^+, S_k^-, and S_k^0. Return to the main step.

Termination: if the initial incumbent value remains the same ($\overline{F} = \infty$), hence, there is no feasible solution to the bi-level problem. Otherwise, consider the feasible solution associated with $\overline{F}$ as the optimal one.

This algorithm based in the branch-and-bound scheme solves exactly the equivalent single-level reformulation of the linear bi-level programming problem. The latter arises from linear generalized semi-infinite problems. Therefore, this algorithm is used to solve LGSIPs optimally.

4. Numerical Experimentation

In this section, the algorithm described above is used to solve a set of ten linear semi-infinite problems with different characteristics. Its effectiveness and suitability to solve LGSIPs are shown. Since various examples will be shown, the semi-infinite problem, its feasible set, the optimal value, and the optimal set are distinguished as P_i, M_{SIP}^i, v_i, and F_i^*, respectively, where i denotes the number of example. Moreover, for the bi-level form, we also distinguish its corresponding sets $S_i(\mathbf{x})$, S_i, G_i, and $M_{BL_{SIP}}^i$.

The aim of this section is to exhibit that the proposed algorithm is able to solve LGSIPs in an efficient manner. The optimal solution of the tested problems is compared with the optimal solution obtained analytically. Also, eight additional linear bi-level problems are tested to robust the computational section showing that the algorithm is able to efficiently solve these problems. Since the majority of the semi-infinite problems herein tested are proposed by us, we do not compare the efficiency results of our algorithm against other methods. However, a comparison regarding the quality of the solution is made.

The first three examples show the particular case when $Y(\mathbf{x}) = Y$. Although the study is only of interest in the general case (since in standard SIPs, they are inherently finite by the vertex theorem), standard SIPs are considered to show that the algorithm is effective in both cases.

Example 1. Consider the following problem:

$$\begin{aligned} P_1: \ & \min_{\mathbf{x}} x_1 + 2x_2, \\ & \text{s.t. } x_1 + x_2 \geq y, \; y \in Y = [0, 1]. \end{aligned} \tag{13}$$

Figure 1 shows that the problem has feasible solution, $v_1 = 1$ and $F_1^* = \left\{(1, 0)'\right\}$.

The bi-level formulation of P_1 is

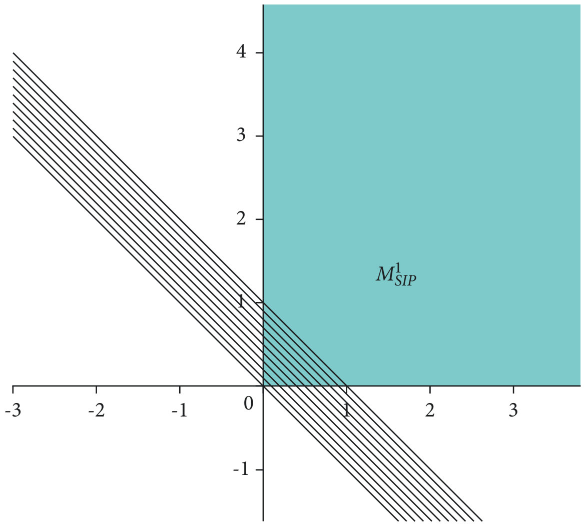

FIGURE 1: Feasible set of P_1.

$$\begin{aligned} &\min_{\mathbf{x}} x_1 + 2x_2 \\ &\text{s.t. } x_1 + x_2 \geq y, \\ &\text{and } y \text{ solves,} \\ &Q(\mathbf{x}):\ \min_{y} -y \\ &\text{s.t. } 0 \leq y \leq 1. \end{aligned} \tag{14}$$

The solution obtained by the algorithm coincides with the solution obtained geometrically.

Example 2. Consider the following problem:

$$\begin{aligned} &P_2:\ \min_{\mathbf{x}} 2x_1 + x_2 \\ &\text{s.t. } x_1 + x_2 \geq y_1 + y_2,\ \mathbf{y} \in Y = [0,1] \times [0,1]. \end{aligned} \tag{15}$$

The corresponding bi-level formulation is

$$\begin{aligned} &\min_{\mathbf{x}} 2x_1 + x_2 \\ &\text{s.t. } x_1 + x_2 \geq y_1 + y_2, \\ &\text{and } \mathbf{y} \text{ solves,} \\ &Q(\mathbf{x}):\ \min_{\mathbf{y}} -y_1 - y_2 \\ &\text{s.t. } 0 \leq y_1 \leq 1, \\ &0 \leq y_2 \leq 1. \end{aligned} \tag{16}$$

The algorithm shows that $F_2^* = \left\{(0,2)^{'}\right\}$ and $v_2 = 2$.

Example 3. Consider the following problem:

$$\begin{aligned} &P_3:\ \min_{\mathbf{x}} 2x_1 + x_2 \\ &s.t.\ x_1 + x_2 \geq y_1 + y_2,\ \mathbf{y} \in Y, \end{aligned} \tag{17}$$

where

$$Y = \{\mathbf{y} | 0 \leq y_1 + y_2 \leq 1\}. \tag{18}$$

The bi-level formulation is

$$\begin{aligned} &\min_{\mathbf{x}} 2x_1 + x_2 \\ &\text{s.t. } x_1 + x_2 \geq y_1 + y_2, \\ &\text{and } \mathbf{y} \text{ solves,} \\ &Q(\mathbf{x}):\ \min_{\mathbf{y}} -y_1 - y_2 \\ &\text{s.t. } 0 \leq y_1 + y_2 \leq 1. \end{aligned} \tag{19}$$

The results obtained by the algorithm are $F_3^* = \left\{(0,1)^{'}\right\}$ and $v_3 = 1$.

The following examples show that the proposed algorithm is also effective to solve semi-infinite problems in the general case.

Example 4. Consider the following problem:

$$\begin{aligned} &P_4:\ \max_{x} x \\ &\text{s.t. } 2y + x - 1 \leq 0,\ y \in Y(x), \end{aligned} \tag{20}$$

where

$$Y(x) = \left\{ y | 0 \leq y \leq \frac{x}{2} \right\}. \tag{21}$$

This example is presented in [23], where the bi-level problem has optimal set $F^* = \left\{((1/2),(1/4))^{'}\right\}$ and its optimal value is $v_4 = (1/2)$.

To solve the semi-infinite problem using the algorithm herein proposed, its bi-level formulation is deduced:

$$\begin{aligned} &P_{4_{BL}}:\ \min_{x} -x \\ &\text{s.t. } x + 2y \leq 1, \\ &\text{and } y \text{ solves,} \\ &Q(x):\ \min_{y} -2y \\ &\text{s.t. } 0 \leq y \leq \frac{x}{2}. \end{aligned} \tag{22}$$

Then, its equivalent single-level reformulation is as follows:

$$\begin{aligned} &\min_{x} -x \\ &\text{s.t. } x + 2y \leq 1, \\ &u - v = 2, \\ &u\left(\frac{x}{2} - y\right) + vy = 0, \\ &-\frac{x}{2} + y \leq 0, \\ &x \geq 0,\ y \geq 0,\ u \geq 0,\ v \geq 0. \end{aligned} \tag{23}$$

To solve that problem, the nonlinear constraint (complementarity slackness constraint) is omitted and the relaxed linear problem is solved:

$$\begin{aligned}
&\min_{x} - x\\
&\text{s.t. } x + 2y \leq 1,\\
&u - v = 2, \qquad (24)\\
&-\frac{x}{2} + y \leq 0,\\
&x \geq 0, y \geq 0, u \geq 0, v \geq 0.
\end{aligned}$$

The optimal solution of the relaxed problem is $(x, y, u, v)^{'} = (1, 0, 2, 0)^{'}$, with an optimal value of $\overline{F} = -1$. Now, the relaxed complementarity slackness constraints are verified. The algorithm first explores the branch when $u = 0$. In this case, the resulting problem is infeasible, and that branch is fathomed. Then, the branch that considers $(x/2) - y = 0$ is explored. In this case, the optimal solution $(x, y, u, v)^{'} = (0.5, 0.25, 2, 0)^{'}$ is obtained, yielding to $\overline{F} = -0.5$. The search to the global optimum continues, and $v = 0$ is imposed. By doing this, the same solution of the latter problem is obtained. At this point, the obtained solution is a candidate to be the global optimum of the semi-infinite problem since all the complementarity slackness constraints are hold. To finish the search, the constraint $v = 0$ is substituted by $y = 0$ and the resulting problem is solved. The obtained solution is $(x, y, u, v)^{'} = (0, 0, 2, 0)^{'}$ and the objective function value of $\overline{F} = 0$. The algorithm stops and the optimal solution of the bi-level problem is $(x, y, u, v)^{'} = (0.5, 0.25, 2, 0)^{'}$ with $\overline{F} = -0.5$. Hence, the optimal solution for the semi-infinite problem is $x = 0.5$ y $v_4 = 0.5$.

In Figure 2, the representation of the search tree is shown.

Example 5. Consider the following problem:

$$\begin{aligned}
&P_5\colon\ \min_{x} x\\
&\text{s.t. } x - y \leq 0,\ y \in Y(x),
\end{aligned} \qquad (25)$$

where

$$Y(x) = \left\{ y \mid -x - y \leq -3,\ x - y \leq 3,\ \frac{x}{2} + y \leq 3 \right\}. \qquad (26)$$

The corresponding bi-level formulation is

$$\begin{aligned}
&\min_{x} x\\
&\text{s.t. } x - y\\
&\text{and } y \text{ solves,}\\
&Q(x)\colon\ \min_{y} y \qquad (27)\\
&\text{s.t. } -x - y \leq -3,\\
&x - y \leq 3,\\
&\frac{x}{2} + y \leq 3.
\end{aligned}$$

Its single-level reformulation is as follows:

$$\begin{aligned}
&\min_{x} x,\\
&\text{s.t. } x - y \leq 0,\\
&u_1 + u_2 - u_3 + v = 1,\\
&u_1(-3 + x + y) + u_2(3 - x + y)\\
&+ u_3\left(3 - \frac{x}{2} - y\right) + vy = 0, \qquad (28)\\
&-x - y \leq -3,\\
&x - y \leq 3,\\
&\frac{x}{2} + y \leq 3,\\
&x, y, u_1, u_2, u_3, v \geq 0.
\end{aligned}$$

Similarly, as in the previous example, the branch and bounding scheme explores the search tree depicted in Figure 3 to achieve the optimal solution.

The solution that the algorithm obtains is $\left\{(0, 3)^{'}\right\}$ as optimal set and 0 as optimal value, which shows that $F_5^* = \{0\}$ and $v_5 = 0$. The results are validated by solving P_5 in a geometric way. We have

$$S_5(x) = \begin{cases} 3 - x \text{ if } 0 \leq x \leq 3,\\ x - 3 \text{ if } 3 < x \leq 4. \end{cases} \qquad (29)$$

The solution graph of Q is shown in Figure 4. On the contraru, in Figure 5, we show that

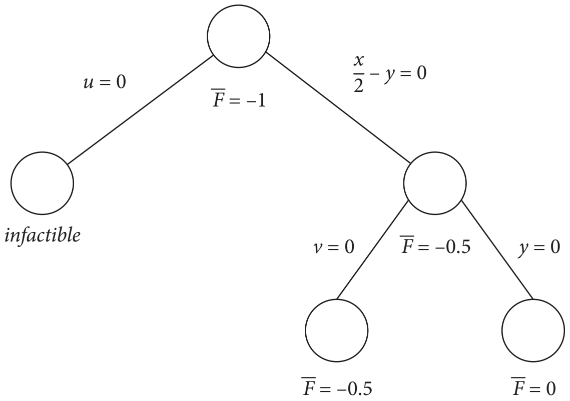

Figure 2: Search tree for Example 4.

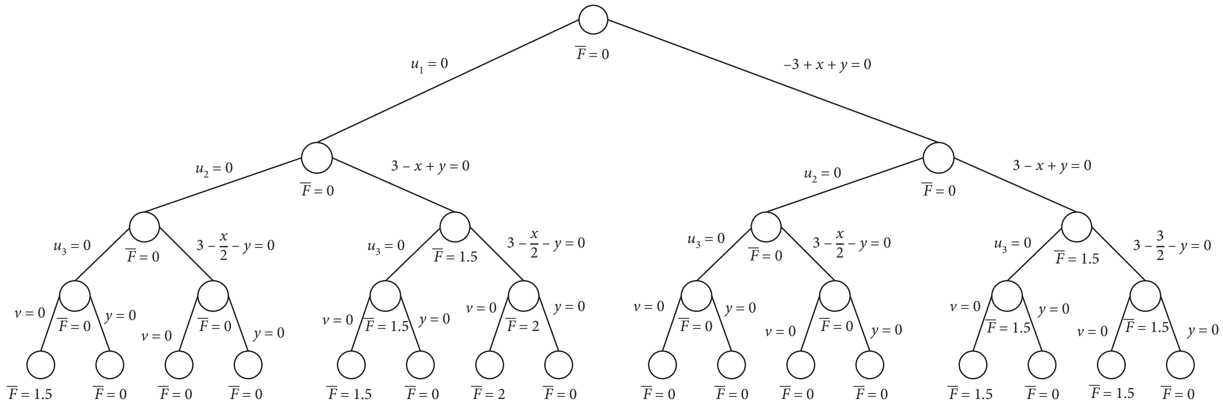

Figure 3: Search tree for Example 5.

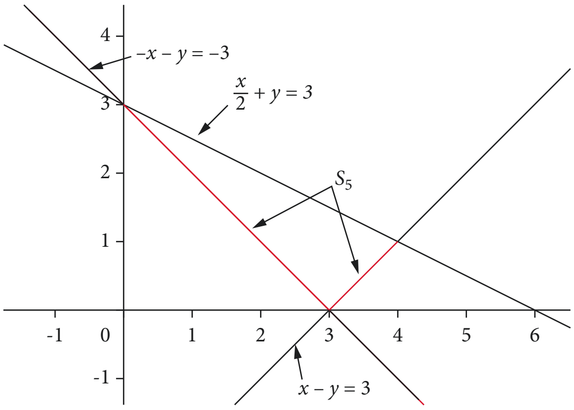

Figure 4: Solution graph of Q which results of P_5.

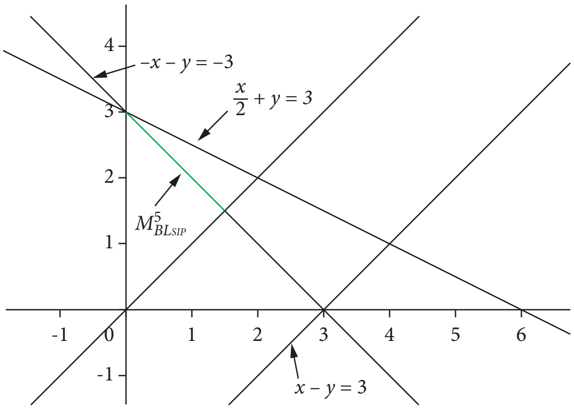

Figure 5: Feasible set of BL_{SIP} which is results of P_5.

$$M^5_{BL_{SIP}} = \left\{ (x, y)' \mid y = 3 - x, 0 \le x \le \frac{3}{2} \right\}. \quad (30)$$

As $M^5_{SIP} = pr_x(M^5_{BL_{SIP}})$, the feasible set is

$$M^5_{SIP} = \left[0, \frac{3}{2} \right]. \quad (31)$$

The optimal solution of the *SIP* problem obtained from the above described feasible set is the same as the one obtained by the algorithm.

Example 6. Consider the following problem:

$$\begin{aligned} P_6\colon\ & \min_{x} x \\ & \text{s.t. } x + y \le 0,\ y \in Y(x), \end{aligned} \tag{32}$$

where

$$Y(x) = \left\{ y \middle| -x - y \le -3,\ x - y \le 3,\ \frac{x}{2} + y \le 3 \right\}. \tag{33}$$

The bi-level formulation is

$$\begin{aligned} & \min_{x} x \\ & \text{s.t. } x + y \le 0, \\ & \text{and } y \text{ solves}, \\ & Q(x)\colon\ \min_{y} -y, \\ & \text{s.t. } -x - y \le -3, \\ & x - y \le 3, \\ & \frac{x}{2} + y \le 3. \end{aligned} \tag{34}$$

The algorithm shows that the problem does not have feasible solution (from the Backtrack step goes to the Termination one). The result is justified geometrically because the intersection of G_6 and S_6 is empty.

Example 7. Consider the following problem:

$$\begin{aligned} P_7\colon\ & \min_{x} x \\ & \text{s.t. } -x + y \le 0,\ y \in Y(x), \end{aligned} \tag{35}$$

where

$$Y\ (x) = \{ y | x + y \le 3 \}. \tag{36}$$

The bi-level form is

$$\begin{aligned} & \min_{x} x \\ & \text{s.t. } -x + y \le 0, \\ & \text{and } y \text{ solves}, \\ & Q(x)\colon\ \min_{y} -y \\ & \text{s.t. } x + y \le 3. \end{aligned} \tag{37}$$

The solution graph of Q is shown in Figure 6. On the contrary, in Figure 7, we show that

$$M^7_{BL_{SIP}} = \{ (x, y)' | y = 3 - x, 1.5 \le x \le 3 \}. \tag{38}$$

Since $M^7_{SIP} = px_x(M^7_{BL_{SIP}})$, the feasible set is $M^7_{SIP} = [1.5, 3]$. The optimal solution is $x = 1.5$. We have the same result with the algorithm.

Example 8. Consider the following problem:

$$\begin{aligned} P_8\colon\ & \min_{\mathbf{x}} -x_1 + x_2 \\ & \text{s.t. } y \ge 0,\ y \in Y(\mathbf{x}), \end{aligned} \tag{39}$$

where

$$Y\ (\mathbf{x}) = \{ y | y \ge x_1,\ y \ge x_2,\ y \le 1 \}. \tag{40}$$

The corresponding bi-level formulation is

$$\begin{aligned} & \min_{\mathbf{x}} -x_1 + x_2 \\ & \text{s.t. } y \ge 0, \\ & \text{and } y \text{ solves}, \\ & Q(\mathbf{x})\colon\ \min_{y} y \\ & \text{s.t. } y \ge x_1, \\ & y \ge x_2, \\ & y \le 1. \end{aligned} \tag{41}$$

The solution obtained by the algorithm is $(x_1, x_2)' = (1, 0)'$ and $v_8 = -1$.

If the restriction $y \le 1$ is omitted in the problem P_8, then the problem is unbounded. Moreover, if x_1 and x_2 have no sign constraints, the feasible set exhibits a so-called re-entrant corner point at the origin (see [2]).

Example 9. Consider the following problem:

$$\begin{aligned} P_9\colon\ & \min_{\mathbf{x}} x_1 - x_2 \\ & \text{s.t. } y \ge 0,\ y \in Y(\mathbf{x}), \end{aligned} \tag{42}$$

where

$$Y(\mathbf{x}) = \{ y | x_1 - y \le -1,\ x_2 - y \le 1, y \le 2 \}. \tag{43}$$

The corresponding bi-level formulation is

$$\begin{aligned} & \min_{\mathbf{x}} x_1 - x_2 \\ & \text{s.t. } y \ge 0 \\ & \text{and } y \text{ solves}, \\ & Q(\mathbf{x})\colon\ \min_{y} y \\ & \text{s.t. } x_1 - y \le -1, \\ & x_2 - y \le 1, \\ & y \le 2. \end{aligned} \tag{44}$$

The solution obtained by the algorithm is $(x_1, x_2)' = (0, 3)'$ and $v_9 = -3$.

Example 10. Consider the following problem:

$$\begin{aligned} P_{10}\colon\ & \min_{x_2 \le 3} -x_1 - x_2 \\ & \text{s.t. } y \ge 0,\ y \in Y(\mathbf{x}), \end{aligned} \tag{45}$$

where

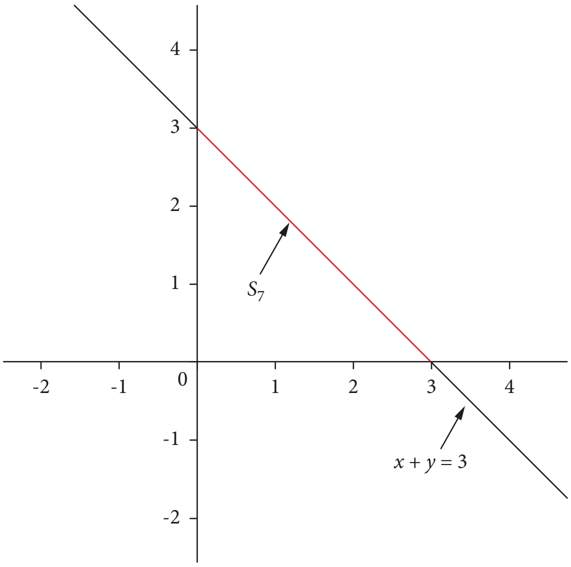

Figure 6: Solution graph of Q which results of P_7.

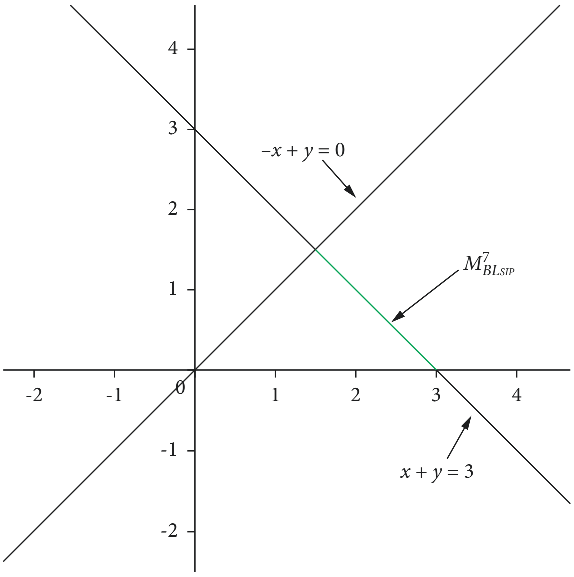

Figure 7: Feasible set of BL_{SIP} which results of P_7.

$$Y\ (\mathbf{x}) = \{y | y \ge x_1, y \le x_2\}. \tag{46}$$

The corresponding bi-level formulation is

$$\begin{aligned} &\min_{x_2 \le 3} -x_1 - x_2 \\ &\text{s.t. } y \ge 0 \\ &\text{and } y \text{ solves,} \\ &Q(\mathbf{x}):\ \min_{y} y \\ &\text{s.t. } y \ge x_1, \\ &y \le x_2. \end{aligned} \tag{47}$$

The solution obtained by the algorithm is $(x_1, x_2)' = (3, 3)'$ and $v_{10} = -6$.

If the restriction $x_2 \le 3$ is omitted in the problem P_{10}, then the problem is unbounded. Moreover, if x_1 and x_2 have no sign constraints, the feasible set is not closed (see [2]).

The examples and computational results are summarized in Table 1, in which the problem is shown in the first column, the optimal solution is reported in the second one, the number of mathematical programming problems solved until the optimal solution obtained is presented in the third column, the total number of mathematical programming problems solved to guarantee optimality is shown in the fourth column, and the total CPU required time is reported in the last column. All tests were performed using a personal computer having an Intel (R) Core processor with 8 GB of RAM, and the proposed algorithm was coded and run using the FICO XPRESS 8.5 commercial software. Also, as mentioned in each example, the optimal solutions are taken from the cited references in which each problem is proposed.

The effectiveness of the algorithm is shown by solving an additional set of problems that appears in [3, 21, 22, 27]. Here, bi-level problems are considered to show the robustness of the developed method. The algorithm starts from the point in which a bi-level reformulation of the linear generalized semi-infinite programming problem is obtained. As it is explained above, the algorithm aims to find the optimal solution, but then, global optimality must be guaranteed. As a consequence, additional iterations are required. However, the required time to solve the tested problems is very short. The obtained results are presented in Table 2 in which we follow the notation of the authors.

5. Application to a Real-Life Problem

In this section, a case study regarding a supply chain, in which distribution centers and manufacturers are involved, is used to demonstrate the applicability of the developed algorithm. The problem herein considered is proposed in [28] and considered in [27]. At the upper level, the distribution centers aim to maximize the overall profit, while at the lower level, the manufacturer aims to minimize the costs.

Let y_{ij} be the upper-level decision variables, which denote the amount of products that will be available at distribution center i. Also, let x_{ij} be the lower-level decision variables and represent the demand of customer j supplied from the distribution center i. Therefore, the upper-level objective function is given by the benefit obtained from selling x_{ij} products at price p_{ij} minus the production costs u_i. The lower-level objective function is given by the manufacturing costs o_{ij} and the acquisition cost from other manufacturers. The imposed constraints are typical on a problem of this type; that is, there is a capacity c_i at the distribution centers, the demand of the customers cannot be supplied from distribution centers that are closed, and at least the minimum demand of customers must be fulfilled.

Considering the particular case study, the obtained data are assumed and the resulting bi-level model is as follows:

Table 1: Results in linear semi-infinite programming.

Example	Optimal solution	Math Prog solved to find the optimal solution	Math Prog solved to guarantee optimality	Required time (sec)
P_1: $\min\limits_{\mathbf{x}} x_1 + 2x_2$ s.t. $x_1 + x_2 \geq y$, $y \in Y = [0,1]$	$v_1 = 1$ $v_1 = 1$ $x_1 = 1$ $x_2 = 0$	3	3	0.031
P_2: $\min\limits_{\mathbf{x}} 2x_1 + x_2$ s.t. $x_1 + x_2 \geq y_1 + y_2$, $\mathbf{y} \in Y = [0,1] \times [0,1]$	$v_2 = 2$ $x_1 = 0$ $x_2 = 2$	5	5	0.047
P_3: $\min\limits_{\mathbf{x}} 2x_1 + x_2$ s.t. $x_1 + x_2 \geq y_1 + y_2$, $\mathbf{y} \in Y$, where $Y = \{\mathbf{y} \mid 0 \leq y_1 + y_2 \leq 1\}$	$v_3 = 1$ $x_1 = 0$ $x_2 = 1$	3	3	0.031
P_4: $\max\limits_{x} x$ s.t. $2y + x - 1 \leq 0$, $y \in Y(x)$, where $Y(x) = \{y \mid 0 \leq y \leq (x/2)\}$	$v_4 = (1/2)$ $x = (1/2)$	3	3	0.031
P_5: $\min\limits_{x} x$ s.t. $x - y \leq 0$, $y \in Y(x)$, where $Y(x) = \left\{ y \,\middle|\, \begin{array}{r} -x - y \leq -3 \\ x - y \leq 3 \\ (x/2) + y \leq 3 \end{array} \right\}$	$v_5 = 0$ $x = 0$	2	3	0.031
P_6: $\min\limits_{x} x$ s.t. $x + y \leq 0$, $y \in Y(x)$, where $Y(x) = \left\{ y \,\middle|\, \begin{array}{r} -x - y \leq -3 \\ x - y \leq 3 \\ (x/2) + y \leq 3 \end{array} \right\}$	infeasible	1	1	0.016
P_7: $\min\limits_{x} x$ s.t. $-x + y \leq 0$, $y \in Y(x)$ where $Y(x) = \{y \mid x + y \leq 3\}$	$v_7 = 1.5$ $x = 1.5$	3	3	0.031
P_8: $\min\limits_{\mathbf{x}} -x_1 + x_2$ $s.t.$ $y \geq 0$, $y \in Y(\mathbf{x})$ where $Y(\mathbf{x}) = \left\{ y \begin{array}{l} y \geq x_1 \\ y \geq x_2 \\ y \leq 1 \end{array} \right\}$	$v_8 = -1$ $x_1 = 1$ $x_2 = 0$	2	3	0.031
P_9: $\min\limits_{\mathbf{x}} x_1 - x_2$ s.t. $y \geq 0$, $y \in Y(\mathbf{x})$, where $Y(\mathbf{x}) = \left\{ y \,\middle|\, \begin{array}{r} x_1 - y \leq -1 \\ x_2 - y \leq 1 \\ y \leq 2 \end{array} \right\}$	$v_9 = -3$ $x_1 = 0$ $x_2 = 3$	3	5	0.047
P_{10}: $\min\limits_{x_2 \leq 3} -x_1 - x_2$ s.t. $y \geq 0$, $y \in Y(\mathbf{x})$, where $Y(\mathbf{x}) = \{y \mid y \geq x_1,\ y \leq x_2\}$	$v_{10} = -6$ $x_1 = 3$ $x_2 = 3$	2	3	0.031

Table 2: Results in linear bi-level programming.

Example	Optimal solution	Math Prog solved to find the optimal solution	Math Prog solved to guarantee optimality	Required time (sec)
BL_1: $\min_{\mathbf{x}\geq 0} -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$ s.t. $\min_{\mathbf{y}\geq 0} x_1 + 2x_2 + y_1 + y_2 + 2y_3$ s.t. $-y_1 + y_2 + y_3 \leq 1$ $2x_1 - y_1 + 2y_2 - 0.5y_3 \leq 1$ $2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1$	$v_1 = -29.2$ $x_1 = 0$ $x_2 = 0.9$ $y_1 = 0$ $y_2 = 0.6$ $y_3 = 0.4$	5	15	0.14
BL_2: $\min_{x} -x - 10y$ s.t. $\min_{y} y$ s.t. $-25x + 20y \leq 30$ $x + 2y \leq 10$ $2x - y \leq 15$ $2x + 10y \geq 15$	$v_2 = -18$ $x = 8$ $y = 1$	5	7	0.078
BL_3: $\min_{x\geq 0} x + y$ s.t. $\min_{y\geq 0} -5x - y$ s.t. $-x - \frac{y}{2} \leq -2$ $-\frac{x}{4} + y \leq 2$ $x + y/2 \leq 8$	$v_3 = 28/9$ $x = 8/9$ $y = 20/9$	5	5	0.047
$x - 2y \leq 4$ BL_4: $\min_{x} x + 2y$ s.t. $\min_{y} -y$ s.t. $-x + 2.5y \leq 3.75$ $x + 2.5y \geq 3.75$ $2.5x + y \leq 8.75$	$v_4 = 3$ $x = 0$ $y = 1.5$	1	1	0.016
BL_5: $\min 3x + y$ $s.t.\ 1 \leq y \leq 6$ $\min_{x} -x$ $s.t.\ x + y \leq 8$ $4x + y \geq 8$	$v_5 = 12$ $x = 2$ $y = 6$	5	5	0.047
$2x + y \leq 13$ BL_6: $\min_{y} 2x_2 - y,$ s.t. $0 \leq y \leq 3$ $\min_{x} -x_1$ s.t. $100x_1 - x_2 \leq 1$ $x_2 \leq y$ $x \geq 0$	$v_6 = 0$ $x_1 = 0.01$ $x_2 = 0$ $y = 0$	3	3	0.031

Table 2: Continued.

Example	Optimal solution	Math Prog solved to find the optimal solution	Math Prog solved to guarantee optimality	Required time (sec)
BL_7: $\max x + y$ s.t. y solves $Q(x)$: $\max\limits_{y} 2y$ s.t. $2y + x - 1 \leq 0$ $0 \leq y \leq \frac{1}{2}x$	$v_7 = 1$ $x = 1$ $y = 0$	2	3	0.031
BL_8: $\max x + y$ s.t. $x + 2y \leq 8$ and y is a solution of $Q(x)$: $\max x + y$ s.t. $0 \leq y \leq 4$	$v_8 = 4$ $x = 0$ $y = 4$	3	3	0.031
BL_9: $\min\limits_{x_1} 2x_1 - 11x_2$ s.t. $\min\limits_{x_2} x_1 + 3x_2$ s.t. $x_1 - 2x_2 \leq 4$ $2x_1 - x_2 \leq 24$ $3x_1 + 4x_2 \leq 96$ $x_1 + 7x_2 \leq 126$ $-4x_1 + 5x_2 \leq 65$	$v_9 = -85.09$ $x_1 = 17.45$ $x_2 = 10.90$	6	9	0.08
$-x_1 - 4x_2 \leq -8$ $x_1, x_2 \geq 0$ BL_{10}: $\min\limits_{x_1} -x_2$ s.t. $\min\limits_{x_2} x_2$ s.t. $-x_1 - 2x_2 \leq 10$ $x_1 - 2x_2 \leq 6$ $2x_1 - x_2 \leq 21$ $x_1 + 2x_2 \leq 38$	$v_{10} = -11$ $x_1 = 16$ $x_2 = 11$	6	9	0.064
$-x_1 + 2x_2 \leq 18$ $x_1, x_2 \geq 0$ BL_{11}: $\min\limits_{x_1} -x_1 - 3x_2$ s.t. $\min\limits_{x_2} x_2$ s.t. $-x_1 + x_2 \leq 3$ $x_1 + 2x_2 \leq 12$ $4x_1 - x_2 \leq 12$ $x_1, x_2 \geq 0$	$v_{11} = -16$ $x_1 = 4$ $x_2 = 4$	4	5	0.049

$$\begin{aligned}
&\max_{y_1,y_2} 110x_{11} + 120x_{21} - 40y_1 - 50y_2,\\
&\min_{x_{11},x_{21}} 130x_{11} + 145x_{21},\\
&x_{11} \leq y_1,\\
&x_{21} \leq y_2,\\
&y_1 \leq 1000,\\
&y_2 \leq 500,\\
&y_1 + y_2 \geq 750,\\
&x_{11}, x_{21}, y_1, y_2 \geq 0.
\end{aligned} \tag{48}$$

In order to apply the proposed algorithm, its single-level reformulation obtained by applying its Kuhn–Tucker conditions is as follows:

$$\begin{aligned}
&\max_{y_1,y_2,x_{11},x_{21}.z_k} \quad 110x_{11} + 120x_{21} - 40y_1 - 50y_2,\\
&z_1 - z_6 = -130,\\
&z_2 - z_7 = -145,\\
&x_{11} \leq y_1,\\
&x_{21} \leq y_2,\\
&y_1 \leq 1000,\\
&y_2 \leq 500,\\
&y_1 + y_2 \geq 750,\\
&(y_1 - x_{11})z_1 = 0,\\
&(y_2 - x_{21})z_2 = 0,\\
&(1000 - y_1)z_3 = 0,\\
&(500 - y_2)z_4 = 0,\\
&(-750 + y_1 + y_2)z_4 = 0,\\
&x_{11}z_6 = 0\,,\\
&x_{21}z_7 = 0,\\
&x_{11}, x_{21}, y_1, y_2 \geq 0,\\
&z_k \geq 0, k = 1, \ldots, 7.
\end{aligned} \tag{49}$$

The obtained solution is the same as in [27, 28], that is, $(x_{11}, x_{21}) = (y_1, y_2) = (1000, 500)$, with a corresponding objective function value of 105,000. The algorithm solves 5 mathematical problems, and the optimal solution was assured in the 5th iteration. It required 0.031 seconds to reach the optimal solution. It is important to mention that, in [27], hybrid genetic and particle swarm optimization algorithms solved the problem in 8 iterations, while 6 iterations of a population-based metaheuristic are required in [28]. Both references do not report the computational time, but for the type of algorithms, it is probably that our methodology is faster for solving this real-life application.

6. Conclusions

In this study, the bi-level formulation of linear generalized semi-infinite programming problems is exploited to propose an easy-to-implement exact solution algorithm. The core of the method relies in the use of Kuhn–Tucker optimality conditions to reduce the bi-level formulation into a nonlinear programming problem with complementarity constraints. The main contribution of this study is the approach of satisfying the complementarity constraints via a branch-and-bound scheme. That strategy is utilized in bi-level programming, but the adaptation to solve linear generalized semi-infinite programming problems is a novel contribution in this research area.

Additionally, computational experimentation is conducted to validate the proposed algorithm and its effectiveness. Diverse linear generalized semi-infinite programming problems with different characteristics are considered to validate the robustness of the implemented algorithm. An interesting real-life application is considered during the numerical experience, which demonstrates the convenience and applicability of the proposed approach.

Therefore, the proposed method results in an alternative for solving this particular type of problems that avoid the inclusion of additional functions to regulate the complementarity constraints, as in [8]. Nevertheless, in both methods, similar assumptions are made before applying them.

A generalization of the proposed method for the case when many semi-infinite constraints are considered is presented in [3]. Their proposal is described in a theoretical manner, but its application to specific semi-infinite problems and computational experiments are not reported. The latter may be a challenging further research direction. Another opportunity is to extend the presented algorithm for solving other variants of SIPs. Also, the generalization of the algorithm to the case with many semi-infinite constraints can also be considered as future work.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] M. Diehl, B. Houska, O. Stein, and P. Steuermann, "A lifting method for generalized semi-infinite programs based on lower level Wolfe duality," *Computational Optimization and Applications*, vol. 54, no. 1, pp. 189–210, 2013.

[2] O. Stein, *Bi-level Strategies in Semi-infinite Programming*, KIuwer, Alphen aan den Rijn, Netherlands, 2003.

[3] O. Stein and G. Still, "On generalized semi-infinite optimization and bilevel optimization," *European Journal of Operational Research*, vol. 142, no. 3, pp. 444–462, 2002.

[4] A. Ghate, "Inverse optimization in semi-infinite linear programs," *Operations Research Letters*, vol. 48, no. 3, pp. 278–285, 2020.

[5] H. Djelassi, A. Mitsos, and S. Oliver, "Recent advances in nonconvex semi-infinite programming: applications and

algorithms," *EURO Journal on Computational Optimization*, vol. 9, pp. 1–10, 2021.

[6] F Guerra-Vázquez, J. J. Rückmann, O. Stein, and G. Still, "Generalized semi-infinite programming: a tutorial," *Journal of Computational and Applied Mathematics*, vol. 217, no. 2, pp. 394–419, 2008.

[7] R. Hettich and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications," *SIAM Review*, vol. 35, no. 3, pp. 380–429, 1993.

[8] O. Stein, "How to solve a semi-infinite optimization problem," *European Journal of Operational Research*, vol. 223, no. 2, pp. 312–320, 2012.

[9] R. Reemtsen and S. Görner, "Numerical methods for semi-infinite programming: a survey," *Nonconvex Optimization and Its Applications*, vol. 76, pp. 195–275, 1998.

[10] M. A. Goberna and M. López, *Linear Semi-infinite Optimization*, John Wiley & Sons, England, UK, 1998.

[11] E. Polak, *Optimization, Algorithms and Consistent Approximations*, Springer, Berlin, Germany, 1997.

[12] A. B. Barragán, L. A. Hernández, and M. I. Todorov, "New primal-dual partition of the space of linear semi-infinite continuous optimization problems," *Comptes rendus, Bulgare*, vol. 69, pp. 1263–1274, 2016.

[13] A. B. Barragán, L. A. Hernandez, and M. I. Todorov, "Primal-dual and general primal-dual partitions in linear semi-infinite programming with bounded coefficients," in *MOL2NET, International Conference Series on Multidisciplinary Sciences, AWP-04*Academic Week of Physics, UNIFAL-MG Alfenas, Brazil, 2018.

[14] A. B. Barragán, L. A. Hernández, and M. I. Todorov, "Primal-dual partitions in linear semi-infinite optimization," *XIII Balkan Conference on Operational Research*, vol. 27, pp. 204–210, 2018.

[15] A B Barragán, L. A. Hernández, and M. I. Todorov, "Solvability and primal-dual partitions of the space of continuous linear semi-infinite optimization problems," *Computación Y Sistemas*, vol. 22, no. 2, pp. 315–329, 2018.

[16] R. Hettich and G. Still, "Semi-infinite programming models in Robotics," in *Parametric Optimization and Related Topics II*, G. etal, Ed., Akademie Verlag, Berlin, Germany, 1991.

[17] A. Hoffmann and R. Reinhardt, *On Reverse Chebyshev Approximation Problems*, Technical University of Illmenau, Ilmenau, Germany, 1994.

[18] G. Still, "Generalized semi-infinite programming: theory and methods," *European Journal of Operational Research*, vol. 119, no. 2, pp. 301–313, 1999.

[19] V. Kalashnikov, T. I. Matis, J. F. Camacho-Vallejo, and S. V. Kavun, "Bilevel programming, equilibrium, and combinatorial problems with applications to engineering," *Mathematical Problems in Engineering*, vol. 2015, Article ID 4360909, 3 pages, 2015.

[20] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: from classical to evolutionary approaches and applications," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2018.

[21] J. F. Bard, *Practical Bilevel Optimization*, Kluwer, Dordrecht, The Netherlands, 1998.

[22] S. Dempe, *Foundations of Bilevel Programming*, Kluwer, Alphen aan den Rijn, Netherlands, 2002.

[23] A. Marendet, A. Goldsztejn, G. Chabert, and C. Jermann, "A standard branch-and-bound approach for nonlinear semi-infinite problems," *European Journal of Operational Research*, vol. 282, no. 2, pp. 438–452, 2020.

[24] O. Stein and A. Winterfeld, "Feasible method for generalized semi-infinite programming," *Journal of Optimization Theory and Applications*, vol. 146, no. 2, pp. 419–443, 2010.

[25] A. Winterfeld, "Application of general semi-infinite programming to lapidary cutting problems," *European Journal of Operational Research*, vol. 191, no. 3, pp. 838–854, 2008.

[26] M. A. Leal-Coronado, A. García-Martínez, and V. Kalashnikov, "Effects of globalization under consistent conjectures," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 7, no. 3, pp. 62–74, 2016.

[27] Z. Huang, C. Yang, X. Zhou, and W. Gui, "A novel cognitively inspired state transition algorithm for solving the linear bi-level programming problem," *Cognitive Computation*, vol. 10, no. 5, pp. 816–826, 2018.

[28] R. J. Kuo and Y. S. Han, "A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem-a case study on supply chain model," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 3905–3917, 2011.