Ann Oper Res (2015) 229:771–790
DOI 10.1007/s10479-015-1842-4

Systematic evolutionary algorithm for general multilevel Stackelberg problems with bounded decision variables (SEAMSP)

Ashenafi Teklay Woldemariam[1] · Semu Mitiku Kassa[1]

Published online: 27 March 2015

Abstract Multilevel Stackelberg problems are nested optimization problems which reply optimally to hierarchical decisions of subproblems. These kind of problems are common in hierarchical decision making systems and are known to be NP-hard. In this paper, a systematic evolutionary algorithm has been proposed for such types of problems. A unique feature of the algorithm is that it is not affected by the nature of the objective and constraint functions involved in the problem as long as the problem has a solution. The convergence proof of the proposed algorithm is given for special problems containing non-convex and non-differentiable functions. Moreover, a new concept of (ε, δ)-approximation for Stackelberg solutions is defined. Using this definition comparison of approximate Stackelberg solutions has been studied in this work. The numerical results on various problems demonstrated that the proposed algorithm is very much promising to multilevel Stackelberg problems with bounded constraints, and it can be used as a benchmark for a comparison of approximate results by other algorithms.

Keywords Multilevel Stackelberg problems · Hierarchical decision · Evolutionary algorithm · Systematic sampling

1 Introduction

Many resource allocation or planning problems require compromises among the objectives of several interacting individuals or agencies, most of the time, arranged in hierarchical administrative structure and can have independent even sometimes conflicting objectives. A decision maker (DM) at one level of the hierarchy may have its objective function determined partly

✉ Semu Mitiku Kassa
semu.mitiku@aau.edu.et

Ashenafi Teklay Woldemariam
ashemaths@yahoo.com

[1] Department of Mathematics, Addis Ababa University, P.O. Box 1176, Addis Ababa, Ethiopia

by variables controlled at other levels. Assuming that the decision process has a preemptive nature and having k levels of hierarchy, we consider the decision maker at the kth level to be the *leader* and those at lower levels to be *followers*.

Such type of decision making problems are often modeled as Stackelberg games, but with some finite number of hierarchical levels. We refer to these problems as *decentralized multilevel systems or multilevel Stackelberg problems (MSPs)*. Such problems are also referred to as multilevel programming problems in some literature and have got the attention of researchers in the last four decades.

MSP was first proposed by Bracken and McGill (1973) to model a decentralized non-cooperative decision making problems. However, the formulation of the problem in the form of Bilevel and Multilevel programs was first used in Candler and Townsley (1982) (See also Colson et al. (2007) for the details of the historic development of the problem). MSPs involving random variables was also proposed first in Patriksson and Wynter (1999). These kind of problems are known to be common in various areas of application.

The mathematical form of a general k-level Stackelberg problems can be expressed as:

$$\begin{aligned}
&\min_{x_1\in X_1} f_1(x_1,x_2,\ldots,x_k)\\
&\qquad\qquad \text{s.t. } (x_1,\ldots,x_k)\in S_1, \qquad \text{where } [x_2,x_3,\ldots,x_k] \text{ solves}\\
&\quad \min_{x_2\in X_2} f_2(x_1,x_2,\ldots,x_k)\\
&\qquad\qquad\quad \text{s.t. } (x_1,\ldots,x_k)\in S_2,\ \text{where } [x_3,x_4,\ldots,x_k] \text{ solves}\\
&\qquad \min_{x_3\in X_3} f_3(x_1,x_2,\ldots,x_k)\\
&\qquad\qquad\qquad \text{s.t. } (x_1,\ldots,x_k)\in S_3,\ \text{where } [x_4,\ldots,x_k] \text{ solves}\\
&\qquad\qquad\quad \ddots\\
&\qquad\qquad\quad \min_{x_k\in X_k} f_k(x_1,x_2,\ldots,x_k)\\
&\qquad\qquad\qquad\qquad \text{s.t. } (x_1,\ldots,x_k)\in S_k
\end{aligned} \tag{1}$$

For a mathematical formulation of MSPs, consider the problem in (1) composed of k-levels each characterized by individual objective functions f_i for $i=1,2,\ldots,k$, which are to be minimized by the respective DMs. Let the decision variable space $\mathbb{R}^n$ be partitioned among k-levels, such that $(x_1,\ldots,x_k)\in X_1\times\cdots\times X_k = S\subseteq\mathbb{R}^n$, where $X_i\subseteq\mathbb{R}^{n_i}$, $\sum_{i=1}^k n_i = n$ and S be a nonempty set. Assume that decisions are made sequentially beginning with the leader DM who has control over a vector $x_1\in X_1$, followed by 2^{nd} DM who has control over a vector $x_2\in X_2$ down through k^{th} DM who has control over a vector $x_k\in X_k$. If there are two DMs, in which the leader DM first makes a decision and the follower DM who knows the decision of the opponent makes a decision next, then the problem can be formulated as a Bilevel Stackelberg problem.

Some of the basic components in MSPs are defined below.

1. Constraint region of the MSP is

$$\Omega = \{(x_1,\ldots,x_k)\in S : (x_1,\ldots,x_k)\in S_1\cap S_2\cap\cdots\cap S_k\}$$

2. For each given vector $(x_1^*,\ldots,x_p^*)\in X_1\times\cdots\times X_p$, $1\le p<k$, the feasible region of the $(p+1)$, and lower order levels is given by:

$$\Omega(x_1^*, \ldots, x_p^*) = \{(x_{p+1}, \ldots, x_k) \in X_{p+1} \times \cdots \times X_k : \\ (x_1^*, \ldots, x_p^*, x_{p+1}, \ldots, x_k) \in S_{p+1} \cap \cdots \cap S_k\}$$

3. Projection of Ω onto the $1^{st}, \ldots, p^{th}$, $1 \leq p < k$, levels decision space is given by:

$$\Omega(X_1, \ldots, X_p) = \{(x_1, \ldots, x_p) : \exists (x_{p+1}, \ldots, x_k), (x_1, \ldots, x_k) \in \Omega\}$$

4. For each $(x_1^*, \ldots, x_p^*) \in \Omega(X_1, \ldots, X_p)$, $1 \leq q < k$, $p = k - q$, the rational reaction set for the $(p+1)$ and lower order levels is given by:

$$\begin{aligned} M(x_1^*, \ldots, x_p^*) = \{(x_{p+1}^*, \ldots, x_k^*) : (x_{p+1}^*, \ldots, x_k^*) \\ \in \operatorname{argmin}\{f_{p+1}(x_1^*, \ldots, x_p^*, x_{p+1}, \ldots, x_k) : \\ (x_{p+1}, \ldots, x_k) \in \Omega(x_1^*, \ldots, x_p^*), \\ (x_{p+2}, \ldots, x_k) \in M(x_1^*, \ldots, x_p^*, x_{p+1})\}\} \end{aligned}$$

5. The Induced Region ($\mathcal{IR}$) at level one is:

$$\mathcal{IR} = \{(x_1, \ldots, x_k) \in S_1 : x_1 \in X_1, (x_2, \ldots, x_k) \in M(x_1)\}$$

6. Using the set $\mathcal{IR}$, one can describe the MSP at level one as:

$$\min_{(x_1,\ldots,x_k) \in \mathcal{IR}} f_1(x_1, \ldots, x_k) \tag{2}$$

We refer to any optimal solution of problem (2) as a *Stackelberg solution* of problem (1).

Because of the hierarchical structure of the rational reaction sets, MSPs are difficult to solve. Algorithms for function optimization of such type of problems are generally limited to convex regular functions and the solution methods proposed for MSPs are mainly for Bilevel Stackelberg problems with linear or convex property. Moreover, the linear Bilevel Stackelberg problem has been shown to be NP-hard in Ben-Ayed and Blair (1990) and to be strongly NP-hard in Hansen et al. (1999). However, many functions that can possibly model real life problems are multi-modal, discontinuous, and non-differentiable.

To solve MSPs, many numerical algorithms for linear bilevel programming problems have been designed in the last two decades, such as implicit enumeration scheme in Candler and Townsley (1982), k-th best algorithm in Bialas and Karwan (1984), branch and bound algorithm in Hansen et al. (1999), grid search algorithm in Bard (1983), descent algorithm in Savard and Gauvin (1994), penalty functions method in White and Anandalingam (1993), fuzzy approach in Shih et al. (1996) and genetic algorithms in Liu (1998), Hejazi et al. (2002) and Wang et al. (2011).

Calvete et al. (2009) have proposed a genetic algorithm to solve linear fractional bilevel problems on a polyhedral constraint region. Since the population generated for the algorithm are extreme points of the polyhedron, it cannot be extended to a general bilevel problem with non polyhedral constraint region. On the other hand, the algorithm for nonlinear bilevel Stackelberg problems proposed in Li and Wang (2011) assumes a unique reaction y for each selected x by the leader DM. With this assumption, the follower's problem is considered to be equivalent to its stationary system based on Karush–Kuhn–Tucker (KKT) conditions. Then, y is replaced by $y(x)$ the solution of the KKT system, and as a result, the bilevel Stackelberg problem is converted to a single optimization problem which involves only x and evolutionary algorithm was applied to find an optimal value of the leader's objective function. If the inner problem is non-convex or convex but non-differentiable, however, the above procedure fails to work.

In Wang et al. (2005), another algorithm for nonlinear bilevel Stackelberg problems is proposed by constructing an optimization problem with two objective functions, such that, a Pareto optimal solution for the two objective optimization problem is an optimal solution for a single optimization problem which is transformed from the bilevel Stackelberg problem using the KKT conditions. To find the Pareto optimal solution of the equivalent two objective optimization problem, an evolutionary algorithm was applied. However, this method works for only special classes of bilevel Stackelberg problems as Pareto optimal points need not be optimal for the bilevel problems and *vice-versa* as described in Dempe (2002).

In Hejazi et al. (2002), a genetic algorithm for linear bilevel Stackelberg problems is proposed. However, the proposed genetic algorithm is applied only for a single level problem which is translated from the bilevel Stackelberg problems by deriving the Kuhn–Tucker condition for the second level problem.

The algorithms proposed in Liu (1998) and Tilahun et al. (2012) seem quite more general for bilevel and multilevel programming problems respectively.

In Liu (1998), a genetic algorithm is proposed for solving Stackelberg-Nash equilibrium of general bilevel programming models with multiple followers in which there might be information exchange among the followers. The algorithm in the paper does not assume any convexity, continuity, and differentiability in the models. However, the extension of the algorithm to general bilevel Stackelberg problems and the convergence of the algorithm to the desired solution is not studied for any formulation of the problem.

The algorithm proposed in Tilahun et al. (2012) tries to solve a multilevel problem by calling a $(1+1)$-evolutionary strategy at each level while fixing the decision variables on the other levels. However, running a $(1+1)$-evolutionary strategy at every step of movement decreases the efficiency of the algorithm. Moreover, the convergence of the algorithm to the required solution is not studied and the algorithm may converge to solutions which are too far from a global optimal solution.

Recently, a solution strategy for MSPs with a very special non-convexity formulation in the objectives of inner level problems has been proposed in Kassa and Kassa (2013). The algorithm tries to solve the problem by convexifying the inner level problem and use a multi-parametric programming approach to propose a branch-and-bound algorithm to find a global approximate solution for MSPs with non-convexity at their inner levels.

In this paper, we propose a new evolutionary type algorithm for a general MSP with bounded decision variables. A unique feature of the algorithm is that it is not affected by the behavior of the functions included in a problem and convergence of the algorithm has been studied for special problems containing non-convex and non-differentiable functions at any level.

Most existing researches on MSP describe the quality of approximate solutions by just looking at the objective function values at each level, which in-general may not be correct. Therefore, new definitions concerning (ε, δ)-approximate Stackelberg solutions has been presented in this paper. This definition has been used to select a best solution from a given set of approximate Stackelberg solutions.

The paper is organized as follows. In Sect. 2, the proposed algorithm has been presented. Section 3 presents new studies on (ε, δ)-approximate Stackelberg solutions and convergence proof of the proposed algorithm. Numerical results of illustrative examples are given in Sect. 4. Section 5 presents conclusive remarks of the paper.

2 Systematic evolutionary algorithm for a general MSPs with bounded decision variables (SEAMSP)

2.1 Description of the algorithm

SEAMSP searches Stackelberg solutions of a given MSP in a hierarchical way. Within an iteration, the leader DM observes the reaction of lower level DMs by passing different values of the variables under its control. We call those values, *"virtual control parameters (VCPs)"* of the leader DM. Similarly for each hierarchical VCPs passed by the DMs above their level, the other DMs observe the reactions of those DMs below their level by passing their own VCPs separately. The reaction is made sequentially beginning from the lowest kth DM up through the $2nd$ DM. The leader DM records the feedbacks corresponding to all of its VCPs and selects the best of all based on its objective function and constraint region. The best VCP of the leader DM with all the hierarchical reactions of the lower level DMs is placed temporarily in a stock, being taken as the best hierarchical VCPs of the iteration. In the next iterations, the above steps are repeated again and again until the requirements of a termination is fulfilled. Best of the best hierarchical VCPs of the iterations, with regard to the leader's objective function are proposed as an approximate Stackelberg solution of the MSP.

As the number of VCPs for each level increases, the probability of getting better results will increase, but accordingly the number of function evaluations will increase so that the computational time of the algorithm will not be acceptable. Instead, in SEAMSP each DM sends few but *"intelligent VCPs"* having a nice cooperation with the VCPs of the other DMs, so that it can represent the whole decision spaces in a random, unique, diverse and systematic way. The decision space of a variable is divided into equal regions of the number of VCPs, where each VCP represents a unique region. Once representing the decision spaces using[1] systematic sampling, the DMs select candidates for a decision/reaction from the representatives of their spaces in a hierarchical way. Among other statistical sampling techniques we use systematic sampling to: (1) assure representation of whole decision spaces in a random, unique and diverse way; (2) protect a clustered selection of VCPs; (3) manage the selection of VCPs at each iteration, so that each iteration will come with new hierarchical VCPs from a region which didn't have representatives before.

If a VCP is selected as a candidate for a reaction by a DM of any level below the leader, a mutation operator is applied to search for a better local representative on the region where the VCP is selected. The best among a candidate (parent) and its offsprings is/are selected by the DM according to its own objective values. Every iteration comes with new VCPs of the leader DM from a region that does not have a representative in the first decision space and tour the same hierarchical paths for a decision.

The main idea of the proposed algorithm, more or less, depends on:

1. Systematic and Fair distribution of VCPs on S,
2. Definition of Mutation operator,
3. Selection of initial solution set and adaptation to the environment,
4. Termination criteria of the algorithm

The flowchart in Fig. 1 shows the procedures followed by the proposed algorithm to solve a k-level Stackelberg problem (1).

[1] Systematic sampling is a statistical method of selecting sample members from a larger population according to a random starting point and a fixed, periodic interval Wikipedia (2013).

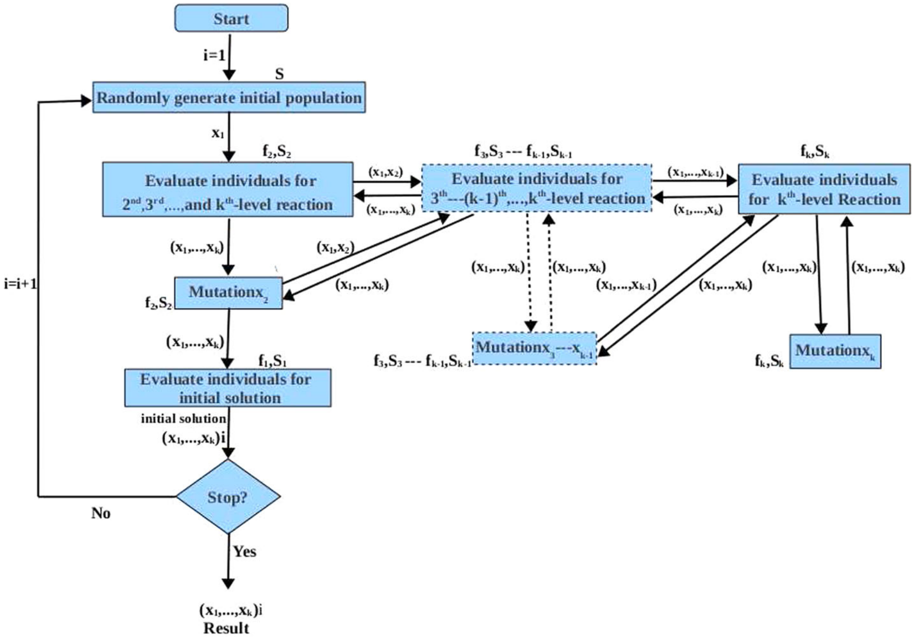

Fig. 1 SEAMSP flowchart for MSPs

2.2 Detailed procedures of the proposed algorithm for the k-level Stackelberg problem, SEAMSP

For the algorithm we require the following **Input** values: n_j, m_j, M_j (where $X_j = [m_j, M_j]^{n_j}$), S_j, and f_j, $j = 1, 2, \ldots, k$.

Notations:

- $d_j := M_j - m_j$, for $j = 1, 2, \ldots, k$
- $X_{ji} := \{x^* : x^*$ is a VCP on X_j, at iteration $i\} \subseteq X_j$, $j = 1, 2, \ldots, k$
- $2b_j :=$ number of VCPs on a single dimension space of X_j, $j = 1, 2, \ldots, k$
- $maxiter :=$ maximum number of iterations before termination
- $\pi_j := 1 + |S_j|$, where $|S_j|$ denotes the number of constraint functions in S_j, $j = 1, 2, \ldots, k$
- $a \lesssim b :=$ greatest natural number a which is less than or equal to b
- $a - b \approx 0 :=$ possible numbers a and b with minimum value of $|a - b|$
- $d := \prod_{j=1}^{k} d_j^{n_j}$, $d' := \prod_{j=2}^{k} d_j^{n_j}$, $n := \sum_{j=1}^{k} n_j$

Parameters of the algorithm:

- $\alpha :=$ a maximum number of hierarchical VCPs on $S = X_1 \times \cdots \times X_k$, in one iteration
- $\gamma :=$ a minimum number of iterations, if not terminated by $maxiter$
- $t(< \gamma) :=$ number of nonempty initial solutions required before a start of computing tolerance of initial solutions of an iteration based on the leader objective function
- $\epsilon_1 :=$ tolerance of a reaction, from VCPs on X_j, $j = 2, 3, \ldots k$
- $\epsilon :=$ maximum tolerance, with respect to the leaders problem, of consecutive $(\gamma - t)$ initial solutions of iterations
- $\beta_1 :=$ minimum number of VCPs on X_1 for $d_1 = 1$, if not terminated by tolerance

- β_2 := minimum number of iterations, if not terminated by tolerance

In the algorithm initial populations are systematically selected and processed using the above given notations and parameter values. The procedures are described in the coming points.

1) The maximum number of VCPs on a single dimensional space of X_j, a_j, are selected as follows:
   To be fair on the distributions of VCPs we consider the following k equations:

$$\prod_{j=1}^{k} a_j^{n_j} \lesssim \alpha \tag{3}$$

$$\gamma \left(\frac{a_1}{d_1}\right)^{n_1} - \left(\frac{a_j}{d_j}\right)^{n_j} \approx 0\,, \text{ where } j = 2, 3, \ldots, k \tag{4}$$

   Equation (3) indicates that the total number of hierarchical VCPs is less than or equal to α and we recommend α to be directly proportional to d. Equation (4) indicates that,[2]$|X_{ji}|$ is directly proportional to d_j, for $j = 1, \ldots, k$.
   From Eqs. (3) and (4), we reach to the following approximations:

$$a_1 \lesssim \left[\frac{\alpha d_1^{n_1(k-1)}}{\gamma^{k-1} d'}\right]^{\frac{1}{n_1 k}}, \; a_j \lesssim d_j \left[\gamma \left(\frac{a_1}{d_1}\right)^{n_1}\right]^{\frac{1}{n_j}}, \tag{5}$$

   where $j = 2, 3, \ldots, k$. Natural numbers $a_j > 1$, $j = 1, \ldots, k$ are chosen from (5), for sufficiently large α.
   Set $b_j = floor\left(\frac{a_j}{2}\right)$,[3] $j = 1, 2, \ldots k$ and step-size $\delta_j = \frac{d_j}{b_j}$, where the range $[m_j, M_j]$ is divided into b_j intervals of step-size δ_j and each interval contains two subintervals of step-size $\frac{\delta_j}{2}$. For $j > 1$, from the two subintervals of the first interval two VCPs are selected randomly and the rest are distributed with a step-size of δ_j from the two randomly selected VCPs, by using systematic sampling.
   To avoid repetition of VCPs on X_1, the two subintervals of the first interval are divided into distinct $maxiter$ sub-subintervals of step size δ_{11}, where $\delta_{11} = \frac{\delta_1}{2maxiter}$. At each iteration new VCPs are randomly selected from two unique sub-subintervals in the two subintervals of the first interval, and distributed with the step size of δ_1 by using systematic sampling technique (Figs. 2, 3, 4 and 5).
2) After the representatives of each decision space are chosen, every VCP on X_1 is sent to search good reactions from the VCPs on X_j, where $j > 1$. While fixing $(x_1^*, \ldots, x_{j-1}^*)$, if $f_j(x_1^*, \ldots, x_k^*) = v_j$ denotes the minimum f_j values at the hierarchical reaction from the[4] $VCPs^*$ on $X_j \times \cdots \times X_k$, then any point $(x_j', \ldots, x_k')$ from the $VCPs^*$ on $X_j \times \cdots \times X_k$ such that $(x_1^*, \ldots, x_{i-1}^*, x_i', \ldots, x_k') \in S_i$ and $f_i(x_1^*, \ldots, x_{i-1}^*, x_i', x_{i+1}^*, \ldots, x_k^*) \leq v_i + \epsilon_1(1 + |f_i(x_1^*, \ldots, x_{i-1}^*, x_i', x_{i+1}^* \ldots, x_k^*)|)$, $\forall i$ such that $j \leq i \leq k$ is considered to be an approximate reaction to $(x_1^*, x_2^*, \ldots, x_{j-1}^*)$.
   The values of v_j are determined by evaluating the objective function f_j from the constraint region $S_j \cap \cdots \cap S_k$ at the $VCPs^*$ on X_j paired with the hierarchical reaction from the $VCPs^*$ on $X_j \times \cdots \times X_k$.

[2] $|X_{ji}|$ stands for a number of VCPs on X_j.

[3] In MATLAB $floor\left(\frac{a_j}{2}\right)$ displays the greatest integer which is less than $\frac{a_j}{2}$.

[4] $VCPs^*$ of any decision space contains all representatives/ VCPs of the decision space and offsprings of the best representatives of the decision space.

The best, among all the representatives of the decision space X_j is assumed to be a parent and mutated to find better representatives from a region bounded by a movement scale of the mutation operator. The best of the parent and offsprings is selected by the respective DM.

Mutation operator is applied to all parents by considering:

(a) random movements of parents within restricted region
(b) against any repetition of offsprings

For instance, if we consider a parent $\lambda_j = [x_1, \ldots, x_{n_j}] \in \mathbb{R}^{n_j}, 2 \leq j \leq k$, then the mutation operator is defined by:

$$\lambda_j^{mut_m} = \begin{pmatrix} x_1 + (A(m,1)\delta_j rand) \\ \vdots \\ x_{n_j} + (A(m,n_j)\delta_j rand) \end{pmatrix},$$

where $m \in \{1, 2, \ldots, 2^{n_j}\}$, A is a $2^{n_j} \times n_j$ matrix having a permutation of rows of entries 1 and -1, (in MATLAB, $A = rowexch(n_j, 2^{n_j})$ can give us the required matrix) and $rand$ denotes a random number between 0 and 1.

Generally offsprings of a parent are reproduced from distinct and equal rectangular neighbors (line-segments, rectangles, rectangular-boxes, …etc) of their parents.

3) For any fixed leader's VCP x_1^*, if $(x_2^*, \ldots, x_k^*)$ denotes the reaction from the $VCPs^*$ on $X_2 \times \cdots \times X_k$ and $(x_1^*, \ldots, x_k^*) \in S_1$, then the point $(x_1^*, \ldots, x_k^*)$ is selected to be approximate point of the induced region, $\mathcal{IR}$. An initial solution set is selected from all approximate points of $\mathcal{IR}$, with the minimum value of the leader's objective function f_1. For a better adaptation of the environment, the initial solution is updated at any iteration having a better f_1 value (Table 1).

In a situation, where every representatives $(x_1^*, \ldots, x_{j-1}^*)$, $1 < j \leq k$ have a unique reaction, $(x_j^*, \ldots, x_k^*)$, from the $VCPs^*$ in $X_j \times \cdots \times X_k$ and $(x_1^*, \ldots, x_k^*) \in S_1$, the total function evaluations in one iteration is:

$$\sum_{j=1}^{k} \pi_j \left(\prod_{i=1}^{j} (2b_i)^{n_i} \right) + \sum_{j=2}^{k} \left(\pi_j 2^{n_j} \prod_{i=1}^{j-1} (2b_i)^{n_i} \right) \tag{6}$$

4) The algorithm terminates whenever the number of iterations reach $maxiter$ or $tol^{(p)} < \epsilon$ for any $p \in \{1, 2, 3, \ldots\}$, where

$$maxiter = \max\left(\beta_1 \left(\frac{d_1}{2b_1} \right)^{n_1}, \beta_2 \right),$$

$$tol^{(p)} = \sum_{i=t+(\gamma-t)(p-1)+1}^{t+(\gamma-t)p} \frac{f_1(x_1^i, \ldots, x_k^i) - f_1(x_1^{i+1}, \ldots, x_k^{i+1})}{1 + |f_1(x_1^i, \ldots, x_k^i)|},$$

$(x_1^i, \ldots, x_k^i)$ is an initial solution at iteration i.

Since $(x_1^i, \ldots, x_k^i)$ is taken as a seed (initial) point in the next iteration, the result of the $(i+1)$ iteration, $(x_1^{i+1}, \ldots, x_k^{i+1})$, is a better solution in terms of the leader's objective function value.

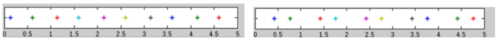

Fig. 2 Possible VCPs on $X_j = [0, 5]$, $b_j = 5$, $j > 1$, in iterations 1 and 2 resp

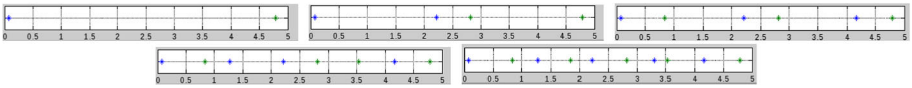

Fig. 3 Aggregate VCPs on $X_1 = [0, 5]$, $b_1 = 1$, $maxiter = 5$, in iterations 1–5 resp

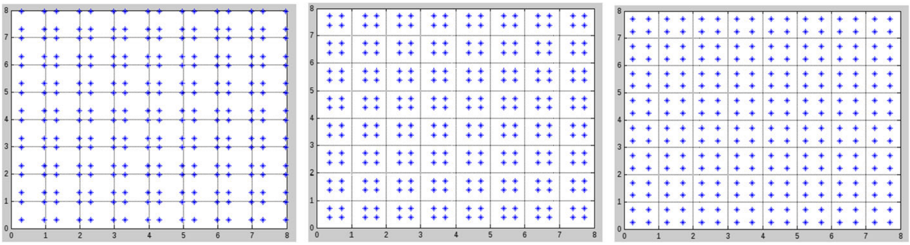

Fig. 4 Possible VCPs on $X_j = [0, 8]^2$, $b_j = 8$, $j > 1$, in iterations 1–3 resp

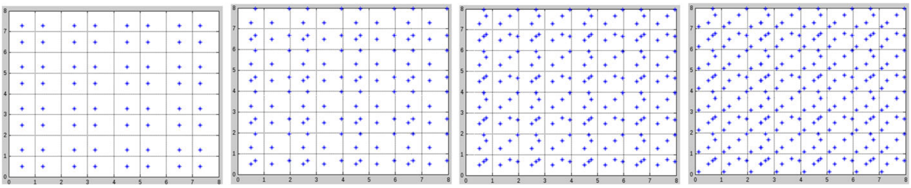

Fig. 5 Aggregate VCPs on $X_1 = [0, 8]^2$, $b_1 = 4$, $maxiter = 4$, in iterations 1–4 resp

Table 1 The proposed algorithm is summarized in the following table

Step	Description
1	Consider the MSP in (1) with bounded decision variables, Input: $n_j, m_j, M_j, S_j, f_j, \forall j \in \{1, \dots, k\}$ Algorithm Parameters: $\alpha, \gamma, t, \beta_1, \beta_2, \epsilon, \epsilon_1$
2	Set iteration $i = 1$ and $\forall j \in \{1, \dots, k\}$, determine δ_j and b_j
3	$\forall j \in \{1, \dots, k\}$ generate VCPs, X_{ji}, of the j^{th}−DM
4	for $j = k : 1 : 2$ $\forall (x_1^*, \dots, x_{j-1}^*) \in X_{1i} \times \cdots \times X_{(j-1)i}$, select parents λ_j from all $x_j \in X_{ji}$ find offsprings, λ_j^*, from a rectangular region around λ_j select the best among the parents and offsprings, λ_j^{*mut}, end for
5	Select the best VCP of the leader's DM, λ_1, from all $x_1 \in X_{1i}$
6	Set an initial solution $(x_1^i, \dots, x_k^i) = (\lambda_1, \lambda_2^{*mut}, \dots, \lambda_k^{*mut})$
7	If $i > 1$, update initial solution $(x_1^i, \dots, x_k^i) =$ $\text{argmin}\{\min(f_1(x_1^i, \dots, x_k^i), f_1(x_1^{i-1}, \dots, x_k^{i-1}))\}$
8	terminate, if the termination criteria is fulfilled; else set $i = i + 1$ and go to step-3
9	**Output:** $(x_1^i, \dots, x_k^i)$

3 Approximation of Stackelberg solutions

While using probabilistic method to solve optimization problems, we don't expect to get exact solutions, instead the algorithms try to approach a solution in an acceptable time. To describe the quality of an approximate Stackelberg solution of a MSP and to select the best result from a set of approximate Stackelberg solutions, one needs to consider the following three basic properties of approximate results:

1. The feasibility of the results to the given MSP
2. The approximation of the results to the reaction sets
3. The fitness of the results, on evaluation of the leader's objective function

Most existing algorithms compare their results by just looking the numerical values of the objective functions and the feasibility of their results at each level, which is generally not correct for approximate Stackelberg solutions. All the objective function values at a non-Stackelberg solution from the constraint region of a given MSP may have a better value than the same function values at a known Stackelberg solution. In other words, a Stackelberg solution doesn't obey Pareto optimality.

Even though there is no common agreement on what an approximate solution to MSPs mean, according to the definition of a Stackelberg solution, in the next subsection we have tried to define different terms in a logical manner.

3.1 $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$:Approximation and comparison methods

Definition 1 A given point $(x_1^*, \ldots, x_k^*) \in \Omega$ is said to be an (ε, δ)-*approximate Stackelberg solution* to the MSP in (1), where $\delta, \varepsilon \geq 0$, if:

1. $f_1(x_1^*, \ldots, x_k^*) \leq f_1(x_1', \ldots, x_k') + \varepsilon(1 + |f_1(x_1^*, \ldots, x_k^*)|)\ ,\ \forall (x_1', \ldots, x_k') \in \mathcal{IR}$
2. $f_j(x_1^*, \ldots, x_k^*) \leq f_j(x_1^*, \ldots, x_{j-1}^*, x_j, \ldots, x_k) + \delta(1 + |f_j(x_1^*, \ldots, x_k^*)|)\ ,$
   $\forall j \in \{2, 3, \ldots, k\},\ \ (x_j, \ldots, x_k) \in M(x_1^*, \ldots, x_{j-1}^*)$

A point $(x_1^*, \ldots, x_k^*)$ satisfying the second condition is called *in a δ-reaction* to x_1^*.

Definition 2 A point $(x_1^*, \ldots, x_k^*) \in \Omega$ is said to be *first-rank better* (or simply better) than $(x_1', \ldots, x_k')$ if

1. $(x_1', \ldots, x_k') \notin \Omega$, or of the following two statements holds true,
2. $\forall (\varepsilon_1, \delta_1)$ such that $(x_1', \ldots, x_k')$ is an $(\varepsilon_1, \delta_1)$-approximate Stackelberg solution, $\exists (\varepsilon, \delta)$ such that $\varepsilon \leq \varepsilon_1, \delta < \delta_1$ or $\varepsilon < \varepsilon_1, \delta \leq \delta_1$ and $(x_1^*, \ldots, x_k^*)$ is an (ε, δ)-approximate Stackelberg solution,

Definition 3 A point $(x_1^*, \ldots, x_k^*) \in \Omega$ is said to be *second-rank better* than $(x_1', \ldots, x_k')$ if

1. $(x_1', \ldots, x_k') \notin \Omega$, or
2. $\forall (\varepsilon_1, \delta_1)$ such that $(x_1', \ldots, x_k')$ is an $(\varepsilon_1, \delta_1)$-approximate Stackelberg solution, $\exists (\varepsilon, \delta)$ such that $\varepsilon + \delta < \varepsilon_1 + \delta_1$ and $(x_1^*, \ldots, x_k^*)$ is an (ε, δ)-approximate Stackelberg solution.

Proposition 1 (a) *Any Stackelberg solution is first-rank better than a non-Stackelberg point and any point can never be second-rank better than a Stackelberg solution.*

(b) *(*Transitivity*) If a point $(x_1^*, \ldots, x_k^*)$ is first-rank (or second-rank) better than $(x_1', \ldots, x_k')$ and $(x_1', \ldots, x_k')$ is first-rank (or second-rank) better than $(x_1'', \ldots, x_k'')$, then $(x_1^*, \ldots, x_k^*)$ is first-rank (or second-rank) better than $(x_1'', \ldots, x_k'')$.*

(c) *A first-rank better solution is second-rank better, but not the converse.*

Proof (a) Let $(x_1^*, \ldots, x_k^*)$ be a Stackelberg solution and $(x_{1*}, \ldots, x_{k*})$ be non-Stackelberg point.
Then $(x_1^*, \ldots, x_k^*)$ is $(0, 0)$-approximate Stackelberg solution and $\forall \varepsilon, \delta \geq 0$, if $(x_{1*}, \ldots, x_{k*})$ is an (ε, δ)-approximate Stackelberg solution then $\varepsilon > 0$ or $\delta > 0$ (or both $\varepsilon > 0$ and $\delta > 0$), hence the result.

(b) If $(x_1'', \ldots, x_k'') \notin \Omega$, the statement holds clearly. Now let $(x_1'', \ldots, x_k'') \in \Omega$, and let $(\varepsilon_1, \delta_1), (\varepsilon_2, \delta_2)$ and $(\varepsilon_3, \delta_3)$ denote the minimum possible values such that $(x_1^*, \ldots, x_k^*)$, $(x_1', \ldots, x_k')$, and respectively $(x_1'', \ldots, x_k'')$ are (ε, δ)-approximate Stackelberg solution. Then $\varepsilon_1 < \varepsilon_2 \leq \varepsilon_3, \delta_1 \leq \delta_2 < \delta_3$ or $\varepsilon_1 \leq \varepsilon_2 < \varepsilon_3, \delta_1 < \delta_2 \leq \delta_3$ (or $\varepsilon_1 + \delta_1 < \varepsilon_2 + \delta_2 < \varepsilon_3 + \delta_3$). Which implies that $\varepsilon_1 < \varepsilon_3, \delta_1 < \delta_3$ (or $\varepsilon_1 + \delta_1 < \varepsilon_3 + \delta_3$), hence the result.

(c) Let $(x_1^*, \ldots, x_k^*)$ be first-rank better than $(x_1', \ldots, x_k')$, where $(\varepsilon_1, \delta_1)$ and $(\varepsilon_2, \delta_2)$ denote the minimum possible values such that $(x_1^*, \ldots, x_k^*)$, and respectively $(x_1', \ldots, x_k')$ are $(\varepsilon_i, \delta_i)$-approximate Stackelberg solutions.
Then either $\varepsilon_1 < \varepsilon_2, \delta_1 \leq \delta_2$ or $\varepsilon_1 \leq \varepsilon_2, \delta_1 < \delta_2$. In both cases $\varepsilon_1 + \delta_1 < \varepsilon_2 + \delta_2$. Which implies that $(x_1^*, \ldots, x_k^*)$ is second-rank better than $(x_1', \ldots, x_k')$.
To show that the converse is not necessarily true, consider the case where, $\varepsilon_1 = 0.01, \varepsilon_2 = 0.02, \delta_1 = 0.001, \delta_2 = 0.0005$, and $(\varepsilon_1, \delta_1), (\varepsilon_2, \delta_2)$ denote the minimum possible values such that $(x_1^*, \ldots, x_k^*)$, and respectively $(x_1', \ldots, x_k')$ are $(\varepsilon_i, \delta_i)$-approximate Stackelberg solutions.
Then $(x_1^*, \ldots, x_k^*)$ is second-rank (but not first-rank) better than $(x_1', \ldots, x_k')$. □

3.2 $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$-convergence of SEAMSP

Convergence of evolutionary algorithms for single-level optimization problems has been studied by different authors, such as Greenwood and Zhu (1999), and Rudolph (1994, 1996, 1998). However, the conflicting multiparamertic behavior of the lower-level problems in the MSPs make it hard to apply the results of these papers. In this subsection, according to Definition 5, the convergence of SEAMSP to a Stackelberg solution of (1) has been studied for problems satisfying the assumptions in Theorem 1 below. The result in the theorem shows the performance of SEAMSP in searching the entire feasible region of a given MSP.

Definition 4 An evolutionary algorithm for a MSP is said to be an (ε, δ)-convergent algorithm if, there exist natural numbers N and τ, such that for any iteration $i \geq N$ and number of initial population τ, the algorithm results with an (ε, δ)-approximate Stackelberg solutions.

Definition 5 An evolutionary algorithm for a MSP is said to be approximately convergent if, $\forall \varepsilon, \delta > 0$ the algorithm is (ε, δ)-convergent.

Theorem 1 *Consider the k-level Stackelberg problem* (1)*, defined in* Sect. 1*, where* $X_i = [m_i, M_i]^{n_i}$ *and* $S^j = X_j \times \cdots \times X_k$*. Let,* $\forall j \in \{1, \ldots, k\}$*,* f_j *be a continuous function and* S_j *be a convex set. Let* $\forall j \in \{2, \ldots, k\}$*,* f_j *be a strictly convex function with respect to* x_j *and* $M(x_1^*, \ldots, x_{j-1}^*) \neq \emptyset$*, where* $(x_1^*, \ldots, x_{j-1}^*) \in X_1 \times \cdots \times X_{j-1}$*. Let* $\forall x \in \mathbb{R}$*,* $\exists \mu_1 > 0$ *such that*[5] $\beta(x, \mu_1) \cap S^1 \subseteq \Omega$*. In addition,* $\forall j \in \{2, \ldots, k\}$*,* $\exists \mu_j > 0$ *such that* $(x_1, \ldots, x_{j-1}, x_{j_*}, \ldots, x_{k_*}) \in \Omega$*, where* $(x_{j_*}, \ldots, x_{k_*}) \in \beta((x_j', \ldots, x_k'),$[6] $\mu_j) \cap S^j$ *and* $(x_j', \ldots, x_k') \in M(x_1, \ldots, x_{j-1})$.

[5] $\beta(x, \mu_1) :=$ an open ball of radius μ_1 about the point x.

[6] $\beta((x_j', \ldots, x_k'), \mu_j) :=$ an open ball of radius μ_j about the point $(x_j', \ldots, x_k')$.

Then $\forall\varepsilon, \delta > 0$, $\exists\alpha < \infty$ *such that each iteration of the SEAMSP results with an* (ε, δ)*-approximate Stackelberg solutions (or equivalently, SEAMSP is convergent to a Stackelberg solution of the given MSP).*

Proof Step-1 (existence of solution)
By definition, the induced region, $\mathcal{IR} = \{(x_1, x_2, \ldots, x_k) \in S_1 : x_1 \in X_1, (x_2, \ldots, x_k) \in M(x_1)\}$. Then using $\mathcal{IR}$ the MSP can be equivalently written as:

$$\min_{(x_1,\ldots,x_k)\in\mathcal{IR}} f_1(x_1, \ldots, x_k) \tag{7}$$

But from the assumption, $\forall x_1 \in X_1$, $\exists\mu_2 > 0$, such that $(x_1, x'_2, \ldots, x'_k) \in \Omega$, for any $(x'_2, \ldots, x'_k) \in \beta((x_2, \ldots, x_k), \mu_2) \cap S^i$, and $(x_2, \ldots, x_k) \in M(x_1) \neq \emptyset$.

Then $(x_1, x_2, \ldots, x_k) \in \Omega$, where $x_1 \in X_1$ and $(x_2, \ldots, x_k) \in M(x_1)$. Which implies that $(x_1, x_2, \ldots, x_k) \in S_1$, where $x_1 \in X_1$ and $(x_2, \ldots, x_k) \in M(x_1)$.

So, problem (7) is equivalent to:

$$\begin{aligned} &\min f_1(x_1, \ldots, x_k) \\ &s.t \quad x_1 \in X_1 \\ &\qquad (x_2, \ldots, x_k) \in M(x_1) \end{aligned} \tag{8}$$

From the assumption, for all $j > 1$, $f_j(x^*_1, \ldots, x^*_{j-1}, x_j, x'_{j+1}, \ldots, x'_k)$ is a strictly convex function, where $(x'_j, x'_{j+1}, \ldots, x'_k) \in M(x^*_1, \ldots, x^*_{j-1}) \neq \emptyset$, $(x^*_1, \ldots, x^*_{j-1}, x_j, x'_{j+1}, \ldots, x'_k) \in S \cap S_j \cap \cdots \cap S_k$, and $(x_j, x'_{j+1}, \ldots, x'_k) \in \beta((x'_j, \ldots, x'_k), \mu_j) \cap S$.

This implies that, the j^{th}-level problem, $j \in \{2, \ldots, k\}$, has at least one reaction, x'_j, for any other reactions below the j-level. But since for all $j > 1$, f_j is strictly convex w.r.t x_j, regardless of the other reactions, the j^{th}-level problem, $j \in \{2, \ldots, k\}$, reacts uniquely. Which implies that, $\forall x_1 \in X_1$, $M(x_1)$ contains a unique point.

Therefore, problem (8) is equivalent to:

$$\begin{aligned} &\min f_1(x_1, M(x_1)) \\ &\quad s.t \quad x_1 \in X_1, \end{aligned} \tag{9}$$

which is an optimization problem of a continuous function over a compact set. Hence, the existence of a solution follows from a well known Weierstrass Theorem.

Step-2 (selection of step-sizes and α)

Let $M(x_1, \ldots, x_{j-1}) = (x'_j, \ldots, x'_k)$, for arbitrary $j > 1$ and $(x_1, \ldots, x_{j-1}) \in X_1 \times \cdots \times X_{j-1}$. From the assumption that, f_j is continuous on $(x_1, \ldots, x_{j-1}, x'_j, \ldots, x'_k)$ we have, $\forall\delta > 0$, $\exists\delta'_j > 0$ such that $f_j(x_1, \ldots, x_k) \in \beta(f_j(x_1, \ldots, x_{j-1}, x'_j, \ldots, x'_k), \delta)$ whenever, $(x_1, \ldots, x_k) \in \beta((x_1, \ldots, x_{j-1}, x'_j, \ldots, x'_k), \delta'_j)$.

Let $\delta^* = min\{\delta'_i : i \in \{j, \ldots, k\}\}$ then, $(x_1, \ldots, x_k) \in \beta((x_1, \ldots, x_{j-1}, x'_j, \ldots, x'_k), \delta^*)$ implies that,

$$|f_j(x_1, \ldots, x_{j-1}, x'_j, \ldots, x'_k) - f_j(x_1, \ldots, x_k)| < \delta.$$

But, $(x_1, \ldots, x_k) \in \beta((x_1, \ldots, x_{j-1}, x'_j, \ldots, x'_k), \delta^*)$ if and only if,

$$\sum_{i=j}^{k} \|x_i - x'_i\|^2 < \delta^{*2}.$$

Choose $x_{i'}$ and $x'_{i'}$ such that[7] $\|x_i - x'_i\| = \|x_{i'} - x'_{i'}\|, \forall i, i' \in \{j, \dots, k\}$. Then

$$(k - j + 1)\|x_j - x'_j\|^2 < \delta^{*2}.$$

This implies that

$$\sum_{i=1}^{n_j} |x_{ji} - x'_{ji}|^2 < \frac{\delta^*}{(k - j + 1)^{\frac{1}{2}}},$$

where $x_j = (x_{j1}, \dots, x_{jn_j})$ and $x'_j = (x'_{j1}, \dots, x'_{jn_j})$.

Now choose $|x_{ji} - x'_{ji}| = |x_{ji'} - x'_{ji'}|, \forall i, i' \in \{1, \dots, n_j\}$. Then

$$|x_{ji} - x'_{ji}| < \left(\frac{\delta^*}{n_j(k - j + 1)^{\frac{1}{2}}}\right)^{\frac{1}{2}} \tag{10}$$

Hence for all $j' \in \{j, \dots, k\}$ any feasible movement of the sample points $(x_{j'}, \dots, x_k)$, from their optimal reaction $(x'_{j'}, \dots, x'_k)$, with a step size of $\left(\frac{\delta^*}{n_j(k-j+1)^{\frac{1}{2}}}\right)^{\frac{1}{2}}$ results in a δ-reaction.

But from the assumption, $\forall j \in \{2, \dots, k\}, \exists \mu_j > 0$ such that, $(x_1, \dots, x_{j-1}, x'_j, \dots, x'_k) \in \Omega$ whenever $(x_j, \dots, x_k) \in \beta((x'_j, \dots, x'_k), \mu_j) \cap S^j$.

Choose $\mu^* = \min\{\mu_i : i \in \{j, \dots, k\}\}$ then, $(x_1, \dots, x_{j-1}, x_j, \dots, x_k) \in \Omega$ whenever $(x_j, \dots, x_k) \in \beta((x'_j, \dots, x'_k), \mu^*) \cap S^j$.

Define r as follows,

$$r = \min\left\{\left(\frac{\delta^*}{n_j(k - j + 1)^{\frac{1}{2}}}\right)^{\frac{1}{2}}, \left(\frac{\mu^*}{n_j(k - j + 1)^{\frac{1}{2}}}\right)^{\frac{1}{2}}, j \in \{2, \dots, k\}\right\}.$$

Then any choice of step sizes, $\delta_j : 0 < \delta_j \leq r, j \in \{2, \dots, k\}$, results with in a δ-reaction.

Let $(x^*_1, \dots, x^*_k)$ be a Stackelberg solution. But f_1 is continuous implies $\forall \varepsilon > 0, \exists \varepsilon^* > 0$ such that, $f_1(x_1, \dots, x_k) \in \beta(f_1(x^*_1, \dots, x^*_k), \varepsilon)$ whenever $(x_1, \dots, x_k) \in \beta((x^*_1, \dots, x^*_k), \varepsilon^*)$.

Using similar procedure as in (10) we can choose x_{ji} with $|x_{ji} - x^*_{ji}| < \left(\frac{\varepsilon^*}{n_1 k^{\frac{1}{2}}}\right)^{\frac{1}{2}}$, where $j \in \{1, \dots, k\}$.

Thus, any feasible movement of a VCP $(x_1, \dots, x_k)$, from the Stackelberg solution $(x^*_1, \dots, x^*_k)$, with a step size of $\left(\frac{\varepsilon^*}{n_1 k^{\frac{1}{2}}}\right)^{\frac{1}{2}}$ results in less than ε difference from $f_1(x^*_1, \dots, x^*_k)$.

But from the assumption $\exists \mu_1 > 0$ such that, $(x_1, \dots, x_k) \in \Omega$ whenever $(x_1, \dots, x_k) \in \beta((x^*_1, \dots, x^*_k), \mu_1) \cap S^1$.

Set $r^* = \min\left\{\left(\frac{\mu_1}{n_1 k^{\frac{1}{2}}}\right)^{\frac{1}{2}}, \left(\frac{\varepsilon^*}{n_1 k^{\frac{1}{2}}}\right)^{\frac{1}{2}}\right\}$ and $p = \min\{r, r^*\}$. Then, any movement of VCPs $(x_1, \dots, x_k)$, from the Stackelberg solution $(x^*_1, \dots, x^*_k)$, with a step size of r^* results in less than ε difference from $f_1(x^*_1, \dots, x^*_k)$.

[7] $\|x\|$:= Euclidean Norm of $x = (x_1, \dots, x_k) \in R^n$, i.e $\|x\| = \left(\sum_{i=1}^{i=n} x_i^2\right)^{\frac{1}{2}}$.

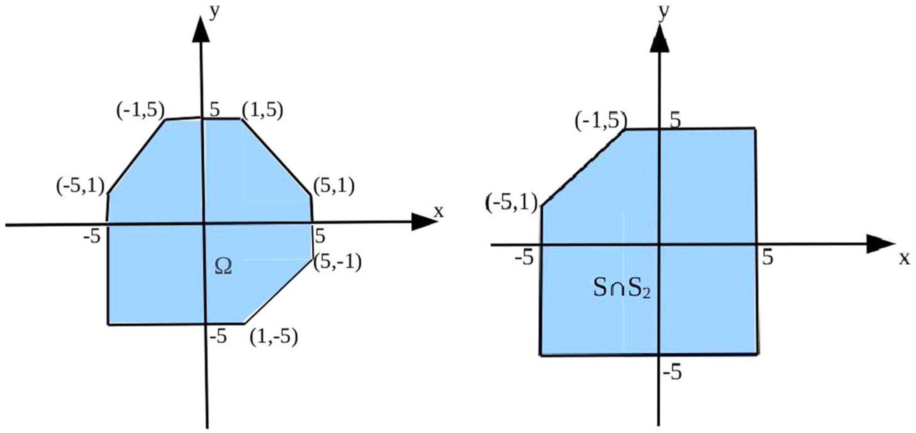

Fig. 6 Feasible sets for problem 11

Hence, any choice of step-sizes δ_j where $0 < \delta_j \leq p$ and $j \in \{1, \ldots, k\}$, results in an (ε, δ)-approximate Stackelberg solution. So the number of VCPs on a single dimensional space of x_j is $b_j = \frac{d_j}{\delta_j}$, for $j \in \{1, \ldots, k\}$.

Now one can easily select α, sufficiently large number which results in $a_j \geq 2b_j, \forall j \in \{1, \ldots, k\}$. □

Note that the above proof has shown that the result can be achieved within a single iteration of the algorithm. When the iteration continuous, the points chosen will be better in terms of ε. Therefore, each iteration of SEAMSP may result in non-Stackelberg but approximate Stackelberg solution of the problem. Initial solution of an iteration is updated by the best VCP of the leader DM together with its hierarchical reaction from the VCPs* of the lower level DMs. This kind of selection guarantees that any non-Stackelberg but first-rank better approximate Stackelberg solution in an iteration, if it exist, will always be set as an initial solution of the iteration.

Remark 1 The assumptions in Theorem 1, could be satisfied by different MSPs, containing non-convex and non-differentiable functions in all levels.

For instance, consider a bilevel Stackelberg problem of the form MSP-(1) where,

$$\begin{aligned} &f_1(x, y) = 5\sin(x^2 + 1) - |\cos(y - 1)| + |x - y|, \\ &f_2(x, y) = |-x^3 + 2x^2| + y^2, \\ &x \in X_1 = [-5, 5], y \in X_2 = [-5, 5], \end{aligned} \tag{11}$$

$S_1 = \{(x, y) : x + y \leq 6, x - y \leq 6\}$ and $S_2 = \{(x, y) : -x + y \leq 6\}$,

The feasible regions of the leader and follower problems are shown in Fig. 6:

Then f_1 and f_2 are continuous functions, S_1 and S_2 are convex sets. $\forall x^* \in X_1$, $M(x^*) = \{0\}$ (i.e $\mathcal{IR} = \{(x^*, 0) : x^* \in X_1\}$) and $f_2(x^*, y) = |-x^{*3} + 2x^{*2}| + y^2$ is a strictly convex function.

We can easily observe from Fig. 6 that, $\forall (x^*, y^*) \in \mathcal{IR} = \{(x^*, 0) : x^* \in X_1\}$, $\exists \mu_1 > 0$ (any μ_1, where $0 < \mu_1 < 1$) such that $(x, y) \in \Omega, \forall (x, y) \in \beta((x^*, 0), \mu_1) \cap S$ and $\exists \mu_2 > 0$ (any μ_2, where $0 < \mu_2 < 1$) such that $(x^*, y) \in \Omega, \forall y \in \beta(0, \mu_2) \cap X_2, x^* \in X_1$.

Hence, it satisfies the assumptions in Theorem 1

Remark 2 The convergence proof of SEAMSP is done only for MSPs satisfying the assumptions in Theorem 1, however the algorithm works for a general MSPs having bounded decision variables. We can observe this from the numerical examples in the next section, in which the problems may contain any kind of nonconvex, non-differentiable and even discontinuous function forms.

4 Numerical results

To show the efficiency and effectiveness of the proposed algorithm for solving MSPs with various types of objective functions, we conduct some numerical experiments. For each test problem, we execute the MATLAB code of the proposed algorithm for 10 independent runs and recorded the following data:

1) The best solution by the algorithm;
2) The objective function values at each solution;
3) The CPU time (in seconds) required.

In-addition to this we recorded (ε, δ)-approximation for those problems which already have a known exact solution and easily determined reactions. An (ε, δ)-approximation are also recorded for those problems having known boundary values of the objective functions.

In all the test problems the algorithm parameters are selected as follows:

$\gamma = 10$, $t = 5$, $\beta_1 = 5$, $\beta_2 = 50$, $\epsilon_1 = 0.005$, and $\alpha = 50,000$ for Examples 2 and 5. But for Examples 1, 3, 4, we used $\alpha = 10,000$, and for Example 6, $\alpha = 70,000$ we used. Moreover, for the Trilevel Problems in Examples 2 and 5, we used the parameter values $\gamma = 6$, $t = 3$, $\beta_1 = 10$, $\beta_2 = 50$, $\epsilon_1 = 0.01$. In Examples 7 and 8 we considered 4-level problems with non convex constraint sets in some of the levels. For these two problems we used the parameter values $\gamma = 4$, $t = 2$, $\beta_1 = 4$, $\beta_2 = 10$, $\epsilon_1 = 0.05$, and $\alpha = 170,000$, while the rest remain the same in the above cases.

The first two examples below are relatively simple and we presented them here only for comparison purpose to what has been reported in literature. Tables 2 and 3

Example 1 Wang et al. (2005)

$$\begin{aligned}
&\min_x f_1(x, y) = x^2 + (y-10)^2\\
&\qquad \text{s.t } (x, y) \in S_1 = \{(x, y) : -x + y \le 0\} \quad \text{where } y \text{ solves}\\
&\quad \min_y f_2(x, y) = (x + 2y - 30)^2\\
&\qquad\quad \text{s.t } (x, y) \in S_2 = \{(x, y) : x + y \le 20\}\\
&\quad x \in X_1 = [0, 15], \text{ and } y \in X_2 = [0, 20]
\end{aligned} \tag{12}$$

Example 2 Tilahun et al. (2012)

$$\begin{aligned}
&\min_x f_1(x, y, z) = -x + 4y\\
&\qquad \text{s.t } (x, y, z) \in S_1 = \{(x, y, z) : x + y \le 1\}\\
&\quad \min_y f_2(x, y, z) = 2y + z\\
&\qquad\quad \text{s.t } (x, y, z) \in S_2 = \{(x, y, z) : -2x + y \le -z\}\\
&\qquad \min_z f_3(x, y, z) = -z^2 + y\\
&\qquad\qquad \text{s.t } (x, y, z) \in S_3 = \{(x, y, z) : z \le x\}\\
&\qquad x \in X_1 = [0, 0.5], \ y \in X_2 = [0, 1] \text{ and } z \in X_3 = [0, 1]
\end{aligned} \tag{13}$$

Table 2 Best solutions by SEAMSP

Ex.Nr	Best Solution	Best f_1	Best f_2	Best f_3	Best f_4
1	(10.0002,9.9981)	100.0035	1.3216e−005		
2	(0.4994, 0.0016, 0.4988)	−0.4929	0.5020	−0.2472	
3	(0.2910, −0.4952, 0.3695, −1.1020, −1.9020, −1.0635)	5.0074	−0.7051		
4	(−1.2076, −1.2893, 0.1794, 0.2028, −1.8235, 0.1345)	0.0268	0.0013		
5	(−0.0290, 0.0062, 0.0041)	−5.9989	8.4293e−004	5.4521e−005	
6	(−0.1074, −0.0055, 0.0703,−0.1074, 0.0686, 3.6320, −0.0671, 0.3630)	0.0016	−4.9946		
7	(1.6503, 1.5488, −0.9988,−1.0022, −0.9899, −0.9976, 1.6486, 1.5998 −1.5487, 1.7001)	−1.1081	0.0100	−4.9961	−0.9943
8	(0.0013, 4.9871, 0.0121,0.0219, 0.0023, 0.0104, 3.1391, 3.2017 3.2196, 3.1985)	−247.8865	0.0634	0.0046	0.0838

Table 3 (ε, δ)-approximate Stackelberg solution of the best solutions by SEAMSP

Ex. Nr.	Mean CPU (sec.)	ε	δ
1	2.7509	1.3216e−5	3.466e−5
2	5.0708	4.756e−3	1.732e−3
3	20.664	1.232e−3	8.211e−4
4	83.5820	0.0262	1.299e−3
5	16.9521	1.572e−4	8.423e−4
6	13.2544	1.598e−3	9.009e−4
7	109.1522	0.1068	0.0100
8	125.2658	0.0117	0.0838

Example 3 (Nonconvex bilevel problem)

$$\min_x f_1(x,y) = \begin{cases} |x_1^2 - x_2^2| + |x_2^2 - x_3^2| + |y_1^2 - y_2^2| + 5, & \text{if } x_1 < x_2 \text{ and } y_1 < y_2 \\ \sin^2(x_2^2 - 2y_1) + 5\cos^2(2y_2 - 3x_1) + 5, & \text{if } x_1 < x_3 \text{ and } x_2 \leq x_1 \\ |y_3^2 - y_2^2 + 9|, & \text{otherwise} \end{cases}$$

$$\text{s.t } (x,y) \in S_1 = \{(x,y) : x_1 - 2x_2^2 y_1 + y_2 \leq 0, x_1 - 4x_2 + 4y_3 \leq 0, \\ -x_1 + 4x_2 - 2y_3 \leq 0\}$$

$$\min_y f_2(x,y) = (y_3 - y_1)^2 - \cos(x_2 - x_1) \tag{14}$$

$$\begin{aligned}&\text{s.t}\ \ (x,y)\in S_2=\{(x,y):-x_1+x_2^2-y_3^2\le 0\}\\ &x=(x_1,x_2,x_3)\in X_1=[-2,2]^3 \text{ and } y=(y_1,y_2,y_3)\in X_2=[-2,2]^3\end{aligned}$$

Example 4 (Nonconvex bilevel problem)

$$\begin{aligned}&\min_x f_1(x,y)=2|\sin(x_1-y_2-x_2+y_4)|\\ &\quad \text{s.t}\ (x,y)\in S_1=\{(x,y):x_1-2x_2^2y_4+y_2\le 0,\\ &\qquad\qquad x_1-4x_2+y_3\le 0,-x_1+4x_2-2y_3\}\\ &\quad\min_y f_2(x,y)=x_1^2y_1^2-2x_1^2y_1y_2+x_1^2y_2^2+5|(y_4-y_1)^3|\\ &\qquad \text{s.t}\ (x,y)\in S_2=\{(x,y):x_1+x_2^2-y_3^2\le 0\}\\ &x=(x_1,x_2)\in X_1=[-2,2]^2 \text{ and } y=(y_1,y_2,y_3,y_4)\in X_2=[-2,2]^4\end{aligned}\tag{15}$$

Example 5 (Trilevel problem with nonconvex inner levels)

$$\begin{aligned}&\min_x f_1(x,y,z)=x^2+4y^2+\sin^2(y+z)-6\\ &\quad \text{s.t}\ (x,y,z)\in S_1=\{(x,y,z):3x-2y-z\le 0,2x-y^2+z^3,2|x|-3y\le 2\}\\ &\quad\min_y f_2(x,y,z)=\frac{1}{5}\sin^2 y+x^2\\ &\qquad \text{s.t}\ \ (x,y,z)\in S_2=\{(x,y,z):x-y^2\le -z\}\\ &\qquad\min_z f_3(x,y,z)=z^2+y^2\\ &\qquad\quad \text{s.t}\ (x,y,z)\in S_3=\{(x,y,z):x+y\le z\}\\ &\qquad x\in X_1=[-2,2],\ \ y\in X_2=[-2,2] \text{ and } z\in X_3=[-2,2]\end{aligned}\tag{16}$$

Example 6 (Nonconvex bilevel problem)

$$\begin{aligned}&\min_x f_1(x,y)=x_1^2(y_1^2+2y_3^2)+3x_2^2(y_2+y_4)^2+y_1^2(x_2^2-2x_1x_2)^2\\ &\qquad\qquad +2y_3^2(x_4^2-2x_1x_4)^2+x_3^2(x_3y_4-2x_2y_4)^2\\ &\quad \text{s.t}\ (x,y)\in S_1=\{(x,y):2|x_3-4|-3y_2\le 2,2x_4-y_2^2+y_3^2\le 1,\\ &\qquad\qquad 3x_1-2y_2-y_4\le 0\}\\ &\quad\min_y f_2(x,y)=2x_1^2y_3^2|y_3-y_4|+x_3^2(x_1y_2-y_1x_2)^2+3x_1^2y_4^2-5\\ &\qquad \text{s.t}\ (x,y)\in S_2=\{(x,y):x_1-y_2^2\le -y_3\}\\ &x=(x_1,x_2,x_3,x_4)\in X_1=[-2,2]^4 \text{ and } y=(y_1,y_2,y_3,y_4)\in X_2=[-4,4]^4\end{aligned}\tag{17}$$

Example 7 (4-level problem with nonconvex constraint sets)[8]

$$\begin{aligned}&\min_w (w_1^2-w_2z_4)^2-\frac{z_1+z_4}{3}\\ &\quad s.t\quad -5z_1+w_2^3\sin w_1\le 0\\ &\qquad \min_x \|y-x\|\\ &\qquad\quad s.t\quad floor(5x_1+z_2+y_1^3)\le 0\\ &\qquad\qquad ceil(2x_1+y_2-x_2+z_3^3)\le 0\end{aligned}$$

[8] In this and the next examples, **log** stands for the natural logarithm; and the **floor** and ceiling (**ceil**) functions are functions that map a real number to the largest previous or the smallest following integer, respectively.

$$\begin{aligned}
&\tan(y_2^3 - x_1 + 3z_1) \leq 0\\
&\min_y 3|\sin^3(y_1 - z_2 - y_2 + z_4)| + 5\cos^3(z_2 - z_3)\\
&\quad s.t \quad \log(y_1 + 2y_2^2 z_4 + y_2) - 2z_1 \leq 0\\
&\qquad 2^{\sin(y_1 - z_1)} + \cos(y_1 - 4z_2 + x_2) - 2z_4 \leq 0\\
&\quad \|z\| - x_1 + 4y_2 + 3x_2 \leq 0\\
&\quad \min_z y_1^2 z_1^2 - 2y_1^2 z_1 z_2 + y_1^2 z_2^2 + 5|(z_4 - z_1)^3| + sin\left(\frac{y_1 + x_2}{2}\right)\\
&\qquad s.t \quad - y_1 + y_2^3 - z_3^2 \leq 0\\
&\qquad\qquad \sin(x_1 + y_1 + z_4) \leq 0\\
&\quad 0 \leq w \leq 5 - 5 \leq x, y \leq 0 \;\; and \; -2 \leq z \leq 2
\end{aligned} \tag{18}$$

Example 8 (Nonconvex 4-level problem with nonconvex constraint sets)

$$\begin{aligned}
&\min_w \|y - z\| + \cos(\frac{w_1 - w_2}{3}) + 2(w_1 - w_2)^3\\
&\quad s.t \quad \cot(x_1 + 2z_1 - y_2) + \csc(z_1 - x_4) \leq 0\\
&\qquad \log(w_2 - w_1^3) - 3\sec(3y_1 - z_2) \leq 0\\
&\qquad \min_x x_1^2(y_1^2 + 2z_1^2) + 3x_2^2(y_2 + z_2)^2 + y_1^2(x_2^2 - 2x_1 x_2)^2 +\\
&\qquad\qquad + 2z_1^2(x_4^2 - 2x_1 x_4)^2 + x_3^2(x_3 z_2 - 2x_2 z_2)^2 + 2x_4^3\\
&\qquad s.t \quad |x_3 - 4| - 3y_2 \leq 0\\
&\qquad\quad 2x_4 - y_2^2 + z_1^2 - 1 \leq 0\\
&\qquad\quad 3x_1 - 2y_2 + z_2 \leq 0\\
&\qquad\quad \min_y 2x_1^2 z_1^2 |z_1 - z_2| + x_3^2(x_1 y_2 + y_1 x_2)^3 + 3x_1^2 z_2^2 + 2x_3^3\\
&\qquad\quad s.t \quad x_1 - y_2^2 + z_1 \leq 0\\
&\qquad\qquad floor(z_1 - 2y_2 + x_3^3) \leq 0\\
&\qquad\qquad \min_z (x_2 - x_4)^2 + |y_1 - y_2| + |z_1 - z_2|\\
&\qquad\quad s.t \quad x_2 + y_2 - z_1 \leq 0\\
&\qquad\qquad ceil(3x_2^3 - 2y_1^3 + z_2) \leq 0\\
&\qquad\qquad 0 \leq w, z \leq 5, 0 \leq x, y \leq 4
\end{aligned} \tag{19}$$

5 Conclusion

In this paper a natural way of passing decision values consecutively into lower level decision makers from the upper levels has been used to develop a heuristic algorithm that can give an approximate global solution for multilevel nonlinear problems. The proposed algorithm can possibly give approximate solutions for multilevel Stackelberg problems with non-differentiable objective functions and also problems with non-convex constraints in their inner levels. It is implemented on a personal computer using MATLAB software and tested on 8 problems that are selected partly from existing literature (one bilevel problem and one trilevel problem) and others constructed for testing purpose.

In all the test problems, SEAMSP results in an (ε, δ)-approximate Stackelberg solutions, with sufficiently small values of ε and δ. SEAMSP has been tested with MSPs including non familiar functions, like discontinuous, non-convex, non-differentiable, etc. functions, in which no existing algorithm can deal with such kind of problems. The proposed algorithm performs well in all the test problems including the difficult once and all the results show that, the proposed algorithm can find near global optimal solutions in a relatively short period of time. This indicates that is efficient and effective. The definitions of (ε, δ)-approximation and the new comparison methods might be used in the future as a reference to select a best approximate Stackelberg solutions of a MSP from a given set of approximate Stackelberg solutions. The newly constructed problems could be used as a benchmark to a comparison of similar algorithms for MSPs.

We believe that, the theoretical and numerical parts of our proposed algorithm constitutes a new and useful addition to the topic. Strengths of the algorithm include that it can be extended to any complex optimization problems. Our further research will extend topics on $(\varepsilon, \delta)-$approximation and (ε, δ)-convergence of algorithms for the general multilevel optimization problems.

Acknowledgments This work is in part supported by the International Science Program (ISP) of Sweden, a research project at the Department of Mathematics, Addis Ababa University.

References

Bard, J. F. (1983). An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, *31*, 670–684.

Ben-Ayed, O., & Blair, C. (1990). Computational difficulties of bilevel linear programming. *Operations Research*, *38*, 556–560.

Bialas, W. F., & Karwan, M. H. (1984). Two-level linear programming. *Management Science*, *30*, 1004–1020.

Bracken, J., & McGill, J. (1973). Mathematical programs with optimization problems in the constraints. *Operational Research*, *22*, 1086–1096.

Calvete, H. I., Galé, C., & Mateo, P. M. (2009). A genetic algorithm for solving linear fractional bilevel problems. *Annals of Operation Research*, *166*, 39–56.

Candler, W., & Townsley, R. (1982). A linear two-level programming problem. *Computers and Operation Research*, *9*, 59–76.

Colson, B., Marcotte, P., & Savard, G. (2007). An overview of bilevel optimization. *Annals of Operation Research*, *153*, 235–256.

Dempe, S. (2002). *Foundation of bileve programming*. Boston: Kluwer Academic Publisher.

Greenwood, G., & Zhu, Q. (1999). Convergence in evolutionary programs with self-adaptation. *Evolutionary Computation*, *7*, 275–312.

Hansen, P., Jaumard, B., & Savard, G. (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, *13*, 1194–1217.

Hejazi, S., Memariani, A., Jahanshahloo, G., & Sepehri, M. (2002). Linear bilevel programming solution by genetic algorithm. *Computers and Operations Research*, *29*, 1913–1925.

Kassa, A., & Kassa, S. (2013). A multi-parametric programming algorithm for special classes of non-convex multilevel optimization problems. *An International Journal of Optimization and Control Theories and Application (IJOCTA)*, *3*(2), 133–144.

Li, H., & Wang, Y. (2011). An evolutionary algorithm with local search for convex quadratic bilevel programming problems. *Applied Mathematics and Information Sciences*, *5*, 139–146.

Liu, B. (1998). Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Computer Mathematics Application*, *36*, 79–89.

Patriksson, M., & Wynter, L. (1999). Stochastic mathematical programs with equilibrium constraints. *Journal Operations Research Letters*, *25*, 159–167.

Rudolph, G. (1994). Convergence properties of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, *5*, 96–101.

Rudolph, G. (1996). *Convergence of evolutionary algorithms in general search spaces: Proceedings of the Third IEEE Conference on Evolutionary Computation*, (pp. 50–54).

Rudolph, G. (1998). Finite Markov chain results in evolutionary computation: A tour d'Horizon. *Fundamenta Informaticae*, *34*, 1–22.

Savard, G., & Gauvin, J. (1994). The steepest descent direction for nonlinear bilevel programming problem. *Operations Research letters*, *15*, 265–272.

Shih, H., Lai, Y., & Lee, E. (1996). Fuzzy approach for multi-level programming problems. *Computers and Operations Research*, *23*(1), 73–91.

Tilahun, S., Kassa, S.M., & Ong, H. (2012). A New Algorithm for Multilevel Optimization Problems Using Evolutionary Strategy Inspired by Natural Adaptation. In P. Anthony, M. Ishizuka, and D. Lkose (Eds.), *PRICAI 2012, LNAI 7458*, (pp. 577–588). Berlin: Springer.

Wang, Y., Li, H., & Dang, C. (2011). A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence. *INFORMS Journal on Computing*, *23*, 618–629.

Wang, Y., Jiao, Y., & Li, H. (2005). An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. *IEEE Transactions on Systems, Man, and Cybernetics Applications and Reviews*, *35*, 221–232.

White, D., & Anandalingam, A. (1993). A penalty function approach for solving bilevel linear programs. *Journal of Global Optimization*, *3*, 397–419.

Wikipedia., (2013). Wikipedia—the free encyclopedia. Accessed January 26, 2013 from http://en.wikipedia.org/wiki/Systematic_Sampling