Swarm and Evolutionary Computation 44 (2019) 496–510

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Using Karush-Kuhn-Tucker proximity measure for solving bilevel optimization problems

Ankur Sinha [a,*], Tharo Soun [b], Kalyanmoy Deb [b]

[a] Production and Quantitative Methods, Indian Institute of Management Ahmedabad, Ahmedabad, 380015, India

[b] Department of Electrical and Computer Engineering, Michigan State University East Lansing, 48823, MI, USA

ARTICLE INFO

Keywords:
Bilevel optimization
Stackelberg games
Evolutionary algorithms
Mathematical programming

ABSTRACT

A common technique to solve bilevel optimization problems is by reducing the problem to a single level and then solving it as a standard optimization problem. A number of single level reduction formulations exist, but one of the most common ways is to replace the lower level optimization problem with its Karush-Kuhn-Tucker (KKT) conditions. Such a reduction strategy has been widely used in the classical optimization as well as the evolutionary computation literature. However, KKT conditions contain a set of non-linear equality constraints that are often found hard to satisfy. In this paper, we discuss a single level reduction of a bilevel problem using recently proposed relaxed KKT conditions. The conditions are relaxed; therefore, approximate, but the error in terms of distance from the true lower level KKT point is bounded. There is a proximity measure associated to the new KKT conditions, which gives an idea of the KKT error and distance from the optimum. We utilize this reduction method within an evolutionary algorithm to solve bilevel optimization problems. The proposed algorithm is compared against a number of recently proposed approaches. The idea is found to lead to significant computational savings, especially, in the lower level function evaluations. The idea is promising and might be useful for further developments on bilevel optimization both in the domain of classical as well as evolutionary optimization research.

1. Introduction

A large number of practical optimization problems are hierarchical in nature involving multiple levels of decision making. For instance, consider two companies, a leader and a follower, operating in a duopoly. Any actions regarding a product launch by the leader company should consider the possible reactions of the follower company. The leader cannot optimize its strategies without considering the actions of the follower, which in turn is optimizing its own strategies after observing the actions of the leader. Such a problem is commonly referred to as a Stackelberg game [56] and the solution that optimizes the strategy of the leader and the follower is referred to as the Stackelberg equilibrium. When this scenario is formulated as a mathematical optimization problem, it represents a challenging class of optimization problem referred to as bilevel optimization problem. There has been an ever growing interest in bilevel optimization problems since these problems were first realized by the mathematical optimization community in 1973 [13]. Initially, the research in bilevel optimization problems was motivated by application problems in the area of game theory; however, the mathematical properties of such problems intrigued the mathematical optimization community that led to a huge body of literature in the past few decades [9,20,51]. Over the years many new application areas for bilevel optimization problem have also been identified, for instance, problems in the areas of defense [15,61], transportation [14,36], investigation of strategic behavior in deregulated markets [26], model production processes [37], chemical engineering [17,55], and optimal tax policies [30,47,53].

Bilevel problems are known to be strongly NP-hard [24], and it has been proven that merely identifying whether a solution is optimal or not itself is NP-hard [58]. It is also know that a linear bilevel problem is hard enough that it is unlikely to develop a polynomial algorithm to solve the problem to global optimality. Given these difficulties, researchers have still attempted to develop methods that are capable of solving special classes of bilevel optimization problems by exploiting its mathematical properties. For instance, linear bilevel problems have received quite a significant attention [11,62]. Some of the strategies

* Corresponding author.
E-mail addresses: asinha@iima.ac.in (A. Sinha), sounthar@msu.edu (T. Soun), kdeb@egr.msu.edu (K. Deb).

https://doi.org/10.1016/j.swevo.2018.06.004
Received 28 October 2017; Received in revised form 16 April 2018; Accepted 3 June 2018
Available online 6 June 2018
2210-6502/© 2018 Elsevier B.V. All rights reserved.

that have been developed for solving linear bilevel problems rely on single level reduction of the bilevel problem by replacing the lower level optimization task with its KKT conditions. The single level formulation is then solved using branch-and-bound (B&B) [7,23] or vertex enumeration [12,16,57] methods. Similar ideas have also been used on other classes of bilevel problems like linear-quadratic [8] and quadratic-quadratic [3,22]. Other variations of KKT conditions have also been considered in some of the studies [42].

In addition to the KKT based techniques, gradient based methods have also been investigated. For example, researchers have attempted to compute approximations of the gradient of the upper level objective [29] or have formulated auxiliary programs [41,58] to determine the direction of descent. Other ideas include penalty based techniques and trust region methods for bilevel optimization. In penalty based techniques, attempts have been made to add penalties to the bilevel problems in different ways, i.e. to lower and/or upper level objectives. Some studies in this direction are [1,2,27,63]. Trust-region based ideas were also used, particularly, for non-linear bilevel optimization problems. In such methods, the original functions in the problem are approximated with model functions in a certain region of the search space. The approximation region is then either expanded or contracted based on the approximation region being good or bad. Some of the studies that utilize trust-regions are [18,33,34]. Interestingly, in Ref. [18] the authors attempt to approximate the original bilevel program at a point with another bilevel program. The model bilevel program is a linear-quadratic bilevel model that is reduced to single level using lower level KKT conditions and solved.

Given that most of the above techniques were limited to handling problems with various simplifying assumptions like linearity, convexity, differentiability etc., during the 1990s researchers started looking into computational approaches like evolutionary algorithms to solve bilevel optimization problems. This of course expanded the class of bilevel problems that can be addressed, however, bilevel problems beyond a few variables were difficult for evolutionary techniques because of the significant rise in computational expense with even a small increase in problem dimensionality. Most of the early attempts that utilized evolutionary algorithms for bilevel optimization were nested in nature, i.e. solving the lower level optimization problem completely for any given upper level point. Researchers used various combinations like, evolutionary algorithm at the upper level and classical optimization algorithm at the lower level [35,64,65], or two different evolutionary algorithms to tackle the two levels [5,32,54]. Researchers from the evolutionary computation community also relied on utilizing lower level KKT conditions to reduce the bilevel problem to single level [25,28,31,59,60]. Another interesting direction of research using evolutionary algorithms creates approximation of various mappings in bilevel optimization to exploit the structure and properties of the bilevel problem to reduce the computational expense [6,43–45,48,50]. Extensions of some of these studies to multi-objective bilevel optimization problems can be found in Refs. [19,49,52].

Despite the existence of above techniques, there is still a strong gap requiring further research and development of bilevel optimization algorithms. While classical approaches are able to address bilevel optimization problems with certain regularities, the evolutionary algorithms are able to address bilevel optimization problems that are not well behaved but small in size (typically 10–20 variables). However, when it comes to solving problems that are not well-behaved and relatively larger in size most of the existing methods suffer. Particularly on the evolutionary optimization front there is a need to improve the existing techniques when it comes to their computational requirements for solving bilevel optimization problems. In this paper, we address this gap by proposing an evolutionary algorithm that relies on approximate KKT conditions to solve bilevel optimization problems. A number of approaches utilize strict KKT conditions to solve bilevel optimization problems. However, it is well-known that KKT conditions are difficult to satisfy [28] because of a number of strict equality constraints. The violations in the KKT conditions need to be small for a point to be close to the optimum; however, the extent of violation of KKT conditions at points in the proximity of a KKT point is not smooth [21]. To counter this problem a proximity measure was proposed in Ref. [21] that reduces sequentially to zero as one approaches a KKT point. We utilize this proximity measure in this paper for solving bilevel problems. A number of bilevel problems from the literature have been successfully solved and the advantage of using the method has been highlighted through comparison studies with other approaches.

In the next section, we provide a formulation of the bilevel problem that has been studied in this paper. Thereafter, we discuss the approximate KKT conditions and apply it to the bilevel formulation. This is followed by a step-by-step procedure for the proposed evolutionary algorithm that utilizes the approximate KKT conditions. Finally, we provide computational results on a number of test problems along with a comparative study with other approaches and conclude the paper.

2. Bilevel formulation and KKT-based single level reduction

From a practical point of view, bilevel optimization assumes that for any given decision by the leader the follower always reacts optimally. The leader's decisions are represented by the upper level variables, and the follower's decisions are represented by the lower level variables. Similarly, the upper level (leader) and lower level (follower) players have their own objectives and constraints. Interestingly, the follower's problem acts as a constraint to the leader's problem, i.e. the leader is optimizing its strategies by taking the optimal responses of the follower into account. The leader's decisions act as parameters to the follower's decisions, i.e. the follower assumes the upper level variables to be fixed while solving the lower level optimization task. An upper level decision and the corresponding lower level optimal decision when feasible with respect to the problem constraints form a feasible solution to the bilevel problem. A general mathematical formulation of the bilevel optimization problem is provided below. A summary of central notations that are commonly used in the context of bilevel optimization can be found in Table 1.

Definition 1. *For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the bilevel optimization problem is given by*

$$\min_{x,y} \; F(x,y)$$

$$\text{subject to}$$

$$y \in \underset{y}{\operatorname{argmin}} \{f(x,y) : g_j(x,y) \le 0, j = 1,\dots,J\}$$

$$G_k(x,y) \le 0, k = 1,\dots,K$$

where $G_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $k = 1,\dots,K$ denotes the upper level constraints, and $g_j : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $j = 1,\dots,J$ represents the lower level constraints, respectively.

The above problem is clearly defined, if for any given decision of the leader, there is only a single optimal strategy for the follower. However, the problem gets ill-defined, if there are multiple optimal solutions at the lower level for any given decision of the leader. It is not clear that for which decision of the follower is the leader optimizing its objective. In such a situation, it is common to assume two extreme positions of bilevel formulation; the optimistic and pessimistic positions. In the optimistic position it is assumed that the follower is cooperating with the leader and chooses that decision from the optimal set that is best for the leader. Of course the follower is indifferent to the many optimal solutions and tries to provide benefit to the leader. However, a leader might not have such a cooperation with the follower. At times, either the leader might have no control over follower's decisions or the fol-

Table 1
Terminologies in bilevel optimization.

Category	Notation(s)	Description
Decision vectors	x	Upper level decision variable.
	y	Lower level decision variable.
Objectives	F	Upper level objective function.
	f	Lower level objective function.
Constraints	$G_k,\ k=1,\dots,K$	Upper level constraint functions.
	$g_j,\ j=1,\dots,J$	Lower level constraint functions.
Lower level feasible region	Ω	$\Omega(x)=\{y : g_j(x,y)\le 0\,\forall j\}$, represents the lower level feasible region for any given upper level decision vector
Relaxed feasible set	$\Phi=\operatorname{gph}\Omega$	$\Phi=\{(x,y) : G_k(x,y)\le 0\,\forall k, g_j(x,y)\le 0\,\forall j\}$, represents the region that satisfies both upper and lower level constraints
Lower level reaction set	Ψ	$\Psi(x)=\{y : y\in \operatorname{argmin}_y\{f(x,y) : y\in\Omega(x)\}\}$, represents the lower level optimal solution(s) for an upper level decision vector
Inducible region (Feasible set)	$I=\operatorname{gph}\Psi$	$I=\{(x,y) : (x,y)\in\Phi, y\in\Psi(x)\}$, represents the set of upper level decision vectors and corresponding lower level optimal solution(s) belonging to feasible constraint region

lower would like to conflict the leader maximally. In such situations, it is ideal for the leader to consider the pessimistic formulation, wherein the problem is solved considering the worst response from the follower out of its optimal set. Pessimistic position is known to be harder to handle as compared to optimistic position. In this paper, we will focus on the optimistic solutions to the bilevel optimization problems.

As discussed earlier, it is possible to reduce a bilevel problem to a single level optimization task with the help of the KKT conditions of the lower level optimization problem. The lower level problem in Definition 1 can be replaced with its KKT conditions to obtain the reduced formulation which has been provided below. Of course the lower level problem needs to adhere to certain convexity and regularity conditions [10] for the KKT conditions to be written.

Definition 2. *The KKT conditions of the lower level lead to the Lagrangian and complementarity constraints, which replace the lower level optimization problem and we obtain a single level formulation:*

$$\min_{x,y,\lambda} \quad F(x,y)$$

subject to

$$G_k(x,y)\le 0, k=1,\dots,K,$$

$$g_j(x,y)\le 0, j=1,\dots,J,$$

$$\nabla_y L(x,y,\lambda)=0,$$

$$\lambda_j g_j(x,y)=0, j=1,\dots,J,$$

$$\lambda_j\ge 0, j=1,\dots,J,$$

$$\textit{where} \quad L(x,y,\lambda)=f(x,y)+\sum_{j=1}^{J}\lambda_j g_j(x,y).$$

Though the above formulation appears more tractable than the bilevel formulation as it is a single level optimization problem, it might not necessarily be any simpler to handle the above problem. There are two difficulties associated with a KKT-based reduction. Firstly, a KKT-based reduction can be done only for very special problems where the lower level satisfies certain regularity conditions [10], like Mangasarian-Fromovitz constraint qualification, and the functions are continuously differentiable. Secondly, even when the formulation can be written, it can be difficult to handle; the Lagrangian constraint ($\nabla_y L(x,y,\lambda)=0$) and the complementary slackness conditions ($\lambda_j g_j(x,y)=0$) are both equality constraints and may lead to non-convexities that are not straightforward to handle. Also the complementary slackness conditions is a product of two functions and any attempt to separate the product terms may lead to inclusion of combinatorial variables in the system. Having said that, the utility of such a reduction also cannot be ignored, as it provides a straightforward way to solve linear bilevel problems (also linear-quadratic), where the Lagrangian constraint is linear and linearizing the complementary slackness conditions makes the overall problem a mixed integer linear program that can be handled by existing solvers.

3. Approximate KKT conditions

The KKT conditions are the first order optimality conditions that necessarily need to be satisfied in order for a solution to be optimal. If the optimization problem under consideration is convex then these conditions guarantee optimality. However, when it comes to solving an optimization problem, KKT conditions are seldom used in the process. Most of the algorithms act directly on the mathematical program rather the KKT conditions. The KKT conditions are usually used post-optimization, to check if a solution obtained by an algorithm is actually a candidate for the optimal solution. Few methods attempt to directly find the KKT point completely ignoring the mathematical program itself. The reason being that identifying a KKT solution directly from the set of KKT constraints itself might not be an easier task than solving the corresponding mathematical program. In a recent study [21], the authors studied the regularity of KKT conditions in the neighborhood of the KKT point. The study explored the behavior of KKT conditions when a sequence of points approached a KKT point through a series of iterates. In their study, they identify the extent of violations in the KKT conditions when one is away from the actual KKT point. The authors were able to derive a simple KKT proximity metric that provides an idea about how far the true KKT point is from the current iterate, if the KKT conditions are violated. The definition for the approximate KKT measure is valid for smooth as well as non-smooth problems. The intention of the authors for proposing an approximate KKT measure was to provide a convergence metric that can act as a termination criteria for smooth as well as non-smooth optimization problems. However, the utility of the idea is not limited only to termination criteria, but as we will see, it can be a useful tool for reducing bilevel problems to single level problems.

Before discussing the utility of the KKT proximity measure for bilevel problems, we introduce the idea on a standard convex optimization problem.

$$\min_{y} \quad f(y)$$

subject to

$$g_j(y)\le 0, j=1,\dots,J,$$

With certain assumptions of smoothness of the functions in the above formulation, it is possible to rewrite the above problem using its modified first-order conditions and the proximity measure ϵ.

$$\min_{y} \quad \epsilon$$
$$\text{subject to}$$
$$g_j(y) \leq 0, j = 1, \ldots, J,$$
$$\|\nabla_y L(y, \lambda)\|^2 \leq \epsilon,$$
$$\sum_{j=1}^{J} \lambda_j g_j(y) \geq -\epsilon$$
$$\lambda_j \geq 0, j = 1, \ldots, J,$$
$$\text{where} \quad L(y, \lambda) = f(y) + \sum_{j=1}^{J} \lambda_j g_j(y).$$

Solving the above optimization problem provides us the proximity measure along with the Lagrange multipliers. The closer the proximity measure is to zero, it gives us an idea about the closeness of the solution to the optimum. For the extension of the above measure to non-smooth problems, the readers may refer to [21]. From here on, we refer to the above formulation as the ϵ-KKT formulation. There can be a number of ways in which ϵ-KKT idea can be beneficial for bilevel optimization problems; below we describe a few ideas that will be later used to develop a bilevel optimization algorithm.

3.1. Using KKT proximity measure as a termination condition for the lower level optimization calls

While applying evolutionary algorithms to bilevel problems, one often requires to solve the lower level optimization problem multiple times. One of the important classes of solution procedures for bilevel optimization are nested methods that solve the lower level optimization problem for every upper level vector generated during the upper level search process. Even approaches that are not nested require a substantial number of calls to the lower level optimization problem. Termination of the lower level optimization task for any given upper level decision is often based on expected improvement over generations of the evolutionary algorithm, i.e. variance of the population members becoming small, no improvement in a pre-specified number of generations etc. Such termination methods [40] that are commonly used in evolutionary algorithms do not provide sufficient confidence about the proximity of the best obtained solution to the optimum or at least a KKT point. Moreover, having a very strict termination criteria based on above methods often leads to wastage of computational resources. Errors made in identifying the lower level optimal solution may lead to issues in converging toward the bilevel optimum. Allowing the lower level to run until a pre-specified ϵ value of KKT proximity measure would provide sufficient confidence for termination.

3.2. Using KKT proximity measure as a constraint parameter

In the previous section, we provided a brief discussion about the reduction of bilevel optimization problem to a single-level optimization problem using the KKT conditions. However, there are a number of difficulties associated with solving such a reduced problem, for instance, handling of the stationarity and the complementary slackness conditions is not straightforward. The ϵ-KKT approach provides us a single parameter ϵ to control the violations in these constraints that limits the errors emanating from the approximate lower level optimum. Given a fixed parameter ϵ_0, the bilevel optimization problem in this case can be reduced to a single level as follows:

$$\min_{x,y,\lambda,\epsilon} \quad F(x,y)$$
$$\text{subject to}$$
$$G_k(x,y) \leq 0, k = 1, \ldots, K,$$
$$g_j(x,y) \leq 0, j = 1, \ldots, J,$$
$$\|\nabla_y L(x, y, \lambda)\|^2 \leq \epsilon,$$
$$\sum_{j=1}^{J} \lambda_j g_j(x,y) \geq -\epsilon$$
$$\lambda_j \geq 0, j = 1, \ldots, J,$$
$$\epsilon \leq \epsilon_0$$
$$\text{where} \quad L(x, y, \lambda) = f(x,y) + \sum_{j=1}^{J} \lambda_j g_j(x,y).$$

When the above problem is solved, the KKT violations at the lower level are always bounded by ϵ_0, which also places a fixed bound on the distance of the lower level optimal solution from the ϵ-KKT point for any given upper level decision. The equality constraints from the KKT conditions are also relaxed, which provide a search window to the algorithm. However, the above formulation requires a parameter ϵ_0, which should be close to zero for higher accuracy. A tight value may still lead to a constrained search space; therefore, we write a penalty-based formulation in the next subsection to solve the above problem.

3.3. Using KKT proximity measure as a penalty term

It is possible to incorporate the KKT proximity measure, ϵ, as a penalty term in the upper level problem, and solve the reduced single-level problem using various penalty handling techniques. The penalty-based formulation for the bilevel optimization problem is provided below:

$$\min_{x,y,\lambda,\epsilon} \quad F(x,y) + r\epsilon$$
$$\text{subject to}$$
$$G_k(x,y) \leq 0, k = 1, \ldots, K,$$
$$g_j(x,y) \leq 0, j = 1, \ldots, J,$$
$$\epsilon = \max(\|\nabla_y L(x, y, \lambda)\|^2, -\sum_{j=1}^{J} \lambda_j g_j(x,y))$$
$$\lambda_j \geq 0, j = 1, \ldots, J,$$
$$\text{where} \quad L(x, y, \lambda) = f(x,y) + \sum_{j=1}^{J} \lambda_j g_j(x,y), \text{and } r \text{ is a penalty parameter.}$$

Note that in the penalty based approach, we need not fix a parameter ϵ_0 a priori; however, we end up incorporating another parameter r. Next, we show the working of the ϵ-KKT idea through an example; then we utilize the above strategies within an evolutionary algorithm to handle bilevel optimization problems.

3.4. An example

Before we proceed to implementing the idea in an evolutionary bilevel algorithm, we provide a graphical insight for the ϵ-KKT measure in the context of a bilevel optimization example. Consider the following simple unconstrained quadratic bilevel problem with one variable at each level.

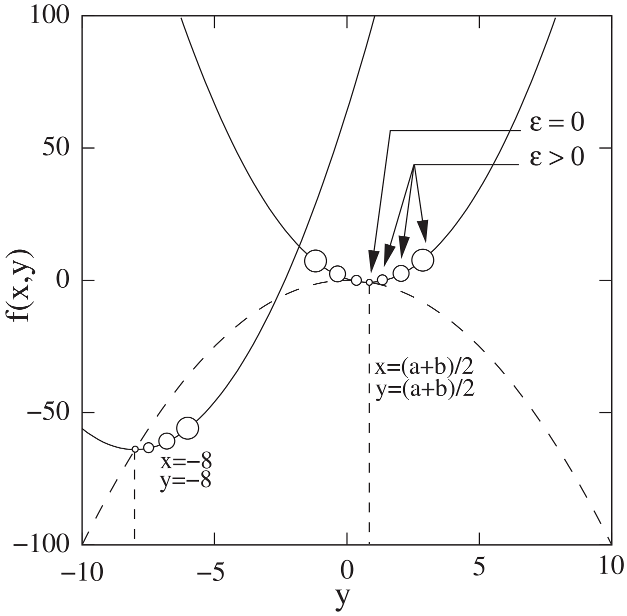

Fig. 1. Lower level optimization problems for two different values of x and the ϵ-KKT points shown by circles. A larger circle represents a larger ϵ-error.

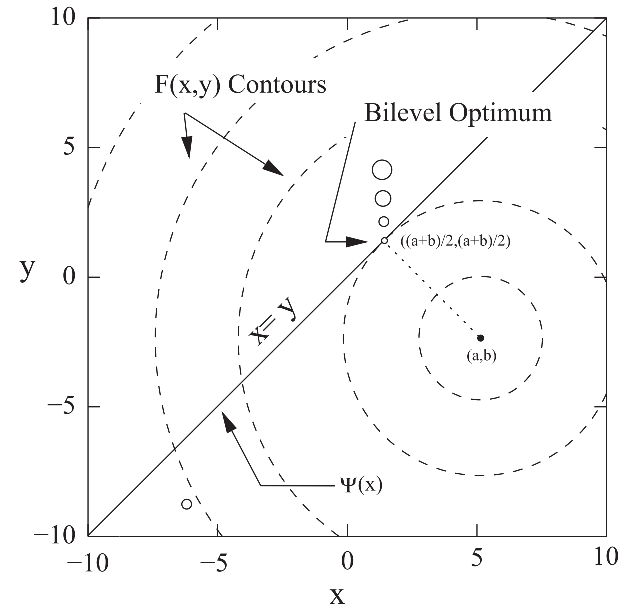

Fig. 2. Upper level objective contours with respect to x and y and the ϵ-KKT points shown by circles. A larger circle represents a larger ϵ-error.

$$
\begin{aligned}
&\underset{(x,y)}{\text{Min}}\ F(x,y)=(x-a)^2+(y-b)^2,\\
&\quad \text{s.t. } y\in \underset{x_l}{\text{argmin}}\left\{f(x,y)=(x-y)^2-x^2\right\},\\
&\quad -10(a+b)\le x,y\le 10(a+b).
\end{aligned}
\tag{1}
$$

The ϵ-KKT based single level formulation for the above problem is given as follows:

$$
\begin{aligned}
&\underset{(x,y)}{\text{Min}}\ F(x,y)=(x-a)^2+(y-b)^2,\\
&\quad \text{s.t.}\\
&\quad 4(x-y)^2\le \epsilon\\
&\quad -10(a+b)\le x,y\le 10(a+b).
\end{aligned}
\tag{2}
$$

Fig. 1 shows the lower level optimization problem corresponding to a few instances of the upper level decisions, i.e., $x=-8$ and $x=(a+b)/2$. The smallest circles corresponding to the two upper level decisions represent the actual lower level optimal solution for the upper level decisions. The broken quadratic curve traces all the possible lower level optimal solution corresponding to different upper level decisions. A solution on this broken curve has a property, such that, the value of optimal y is equal to x, which is obvious from the lower level optimization problem. For the optimal bilevel upper level decision $x=(a+b)/2$ and $x=-8$, we provide the ϵ-KKT points for different values of ϵ through different sizes of the circles. A larger circle represents a larger ϵ or error. Fig. 2, provides the y versus x plot. In Fig. 2, the straight line corresponds to the broken curve from Fig. 1 in terms of x and y, which is the feasible region I for the bilevel problem. The circular contours represent the upper level objective. The ϵ-KKT points from Fig. 1 for $x=(a+b)/2$ are also shown in Fig. 2.

4. Bilevel evolutionary algorithm based on KKT proximity measure as a constraint parameter

In this section, we provide the bilevel evolutionary algorithm that utilizes ϵ-KKT formulation to solve bilevel optimization problems. A stepwise procedure for the implementation of the algorithm is provided in Table 2. Below we provide a detailed explanation for some of the important steps in the algorithm. For a quick overview of the algorithm, the readers may refer to the flowchart shown in Fig. 3.

4.1. Initialization

During the initialization stage, some of the population members at the upper level are created in the relaxed feasible region Φ by solving the following optimization problem.

$$
\begin{aligned}
&\min_{x,y}\quad 0\\
&\quad \text{subject to}\\
&\quad g_j(x,y)\le 0, j=1,\dots,J,\\
&\quad G_k(x,y)\le 0, k=1,\dots,K.
\end{aligned}
$$

The optimization is performed using sequential quadratic programming (SQP) that terminates as soon as a feasible solution is found. The above problem is solved using random starting points leading to different points in the relaxed feasible space.

4.2. Local approximation of functions and constraints

Note that the ϵ-KKT method requires gradients for the functions and constraints. In our algorithm we do not compute the gradients, rather create a sample of points around a point (x^0, y^0) and find a quadratic approximation for the objective functions and linear approximation of the constraints, i.e. we create a bilevel optimization problem that locally approximates the original bilevel problem. Let the following bilevel problem be an approximate representation of the original bilevel problem around a point (x^0, y^0).

$$
\begin{aligned}
&\min_{x,y}\quad \widehat{F}(x,y)\\
&\quad \text{subject to}\\
&\quad y\in \underset{y\in Y}{\text{argmin}}\{\widehat{f}(x,y):\widehat{g}_j(x,y)\le 0, j=1,\dots,J\}\\
&\quad \widehat{G}_k(x,y)\le 0, k=1,\dots,K
\end{aligned}
$$

Table 2
Step-by-step procedure for ϵ-KKT algorithm.

Step	Description
1	**Initialization**: Randomly generate upper level (x) and lower level members (y) in the population $\mathcal{P}$. At least half of the members are created in the relaxed feasible region. Initialize the generation counter $g \leftarrow 0$ (see Section 4.1). **(a)** For each $x^{(i)} \in \mathcal{P}$, find the optimal lower level member. The optimal lower level member replaces the random $y^{(i)}$ created in the previous step. **(b)** Create a random sample of points $(x^{(is)}, y^{(is)}) : s \in \{1, \ldots, S\}$ around each point $(x^{(i)}, y^{(i)})$. Compute their upper and lower level function values and constraints, and add all the points to A_0. **(c)** For each member for which lower level is solved and optimization is successful, tag the vector $(x^{(i)}, y^{(i)})$ as 1; otherwise, tag it as 0. Add the tag 0 members to tag 0 archive $\mathcal{A}_0$ and the tag 1 members to tag 1 archive $\mathcal{A}_1$. **(d)** Assign fitness to all the population members based on upper level function and constraints (see Section 4.5). **(e)** For each vector $(x^{(i)}, y^{(i)}) \in \mathcal{P}$, create local quadratic-linear approximations of objectives and constraints. (see Section 4.2)
2	**Reproduction**: Increment the generation counter by one: $g \leftarrow g + 1$. **(a) Parent selection**: Randomly pick 2μ members from $\mathcal{P}$. Perform tournament selection from the picked members based on their tags and fitness. This results in μ parents for crossover. Tag 1 member is given priority over tag 0 member during comparison. If the tags of the members being compared are the same then their fitness values are compared. **(b) Offspring generation**: Create λ offspring from the μ parents using genetic operations (see Section 4.5). Note that genetic operations at the upper level are performed only with upper level variables.
3	**Update Offsprings**: For each offspring, two methods are used to find its lower level counterpart and update the population $\mathcal{P}$. **(a) Lower Level Optimization**: There is a probability γ that the lower level optimization is performed for the offspring to find the lower level counterpart. If the lower level optimization is successful, the solution is compared using fitness with the worst tag 1 member of the population $\mathcal{P}$, and it replaces that member if it is better. If the lower level optimization is unsuccessful follow a similar process with the worst Tag 0 member in $\mathcal{P}$. The offspring is also added to the archive A_0 or A_1 based on its success flag. **(b) Approximations**: This step is performed if Step 3a is not executed. For a given offspring, using the closest members in A_0 compute local approximations for the lower level objective (quadratic approximation) and constraints (linear approximation). The locally approximated quadratic-linear lower level problem is solved using SQP to get the lower level counterpart for the offspring. Fitness is assigned and the offspring is added to $\mathcal{A}_0$. For each offspring, randomly pick a tag 0 member from $\mathcal{P}$ and compare the fitness. If the offspring is better, then it replaces the chosen tag 0 member in the population.
4	**ϵ-KKT based Local Search** The following search is performed every α generations: $g \bmod \alpha = 0$. **(a)** Randomly choose 2 members from the population $\mathcal{P}$. Perform tournament selection between the two members. Tag 1 member is given priority when compared. Let the winning member be $(x^{(w)}, y^{(w)})$ **(b)** Get new local quadratic-linear approximations of objectives and constraints for the winning member $(x^{(w)}, y^{(w)})$. (see Section 4.2) **(c)** Perform ϵ-KKT based local search with $(x^{(w)}, y^{(w)})$ as the initial point with its approximated objectives and constraints. If the search succeeds, divide ϵ by 10. Otherwise, multiply ϵ by 10 (see Section 4.2). Let the member produced from the search be $(x^{(v)}, y^{(v)})$ **(d)** Find the optimal lower level solution for $x^{(v)}$ by solving the lower level optimization problem with $y^{(v)}$ as the starting point. If lower level optimization is successful then the member is tagged as 1 otherwise 0. The best lower level member obtained from the lower level optimization task replaces $y^{(v)}$. **(e)** Assign fitness to $(x^{(v)}, y^{(v)})$ based on its upper level objective and constraints values. If both $(x^{(w)}, y^{(w)})$ and $(x^{(v)}, y^{(v)})$ have same tags then use their fitness values to choose the better member; otherwise the member with tag 1 is chosen as the better member. If $(x^{(v)}, y^{(v)})$ is better it replaces $(x^{(w)}, y^{(w)})$ in the population. **(f)** Add the solution $(x^{(v)}, y^{(v)})$ to its corresponding archive.
5	**Update best member**: Identify the Tag 1 member in the current generation in $\mathcal{P}$ with the best fitness, denoted as (x_{best}, y_{best}).
6	**Termination check:** Perform a termination check (see Section 4.6). If false, proceed to the next generation (Step 2). If true, report the best member as the bilevel optimum after performing ϵ-KKT based local search on the best member, as discussed in Step 4.

Note that if the objective functions are quadratic and the constraints are linear then the above approximate bilevel problem can be written as follows:

$$\min_{x,y} \widehat{F}(x,y) = (x,y)R(x,y)' + (x,y)S + T$$

subject to

$$y \in \underset{y}{\operatorname{argmin}} \left\{ \widehat{f}(x,y) = (x,y)A(x,y)' + (x,y)B + C : \right.$$

$$\left. \widehat{g}(x,y) = (x,y)D + E \leq 0, \right\}$$

$$\widehat{G}(x,y) = (x,y)U + V \leq 0,$$

where R and A are square matrices, S, B, D and $U$are column vectors, andT, C, E and$V$are constants.

Applying the ϵ-KKT reduction to the above quadratic bilevel problem we obtain a single-level reduction with non-linear constraints. It is possible to solve such a reduced single-level problem using SQP; however, without guaranteeing the optimum if the problem is non-convex. The presence of non-linear complementary slackness conditions may make the problem non-convex. There is a possibility to linearize the complementary slackness condition using the big-M method, but it would lead to inclusion of combinatorial variables in the problem. In this paper, we do not linearize the complementary slackness constraints rather solve the reduced bilevel problem directly using SQP.

4.3. Parameters

The following are the parameters used in the algorithm that have been kept fixed throughout the computations in this study:

$N = 20$ (Population size at upper level)

$n = 20$ (Population size at lower level)

$\epsilon_0 = 10^{-3}$ (Parameter)

$m = 0.1$ (Probability of mutation)

$c = 0.9$ (Probability of crossover)

$\gamma = 0.2$ (Probability of lower level optimization)

$\alpha = 5$ (Generations between local search)

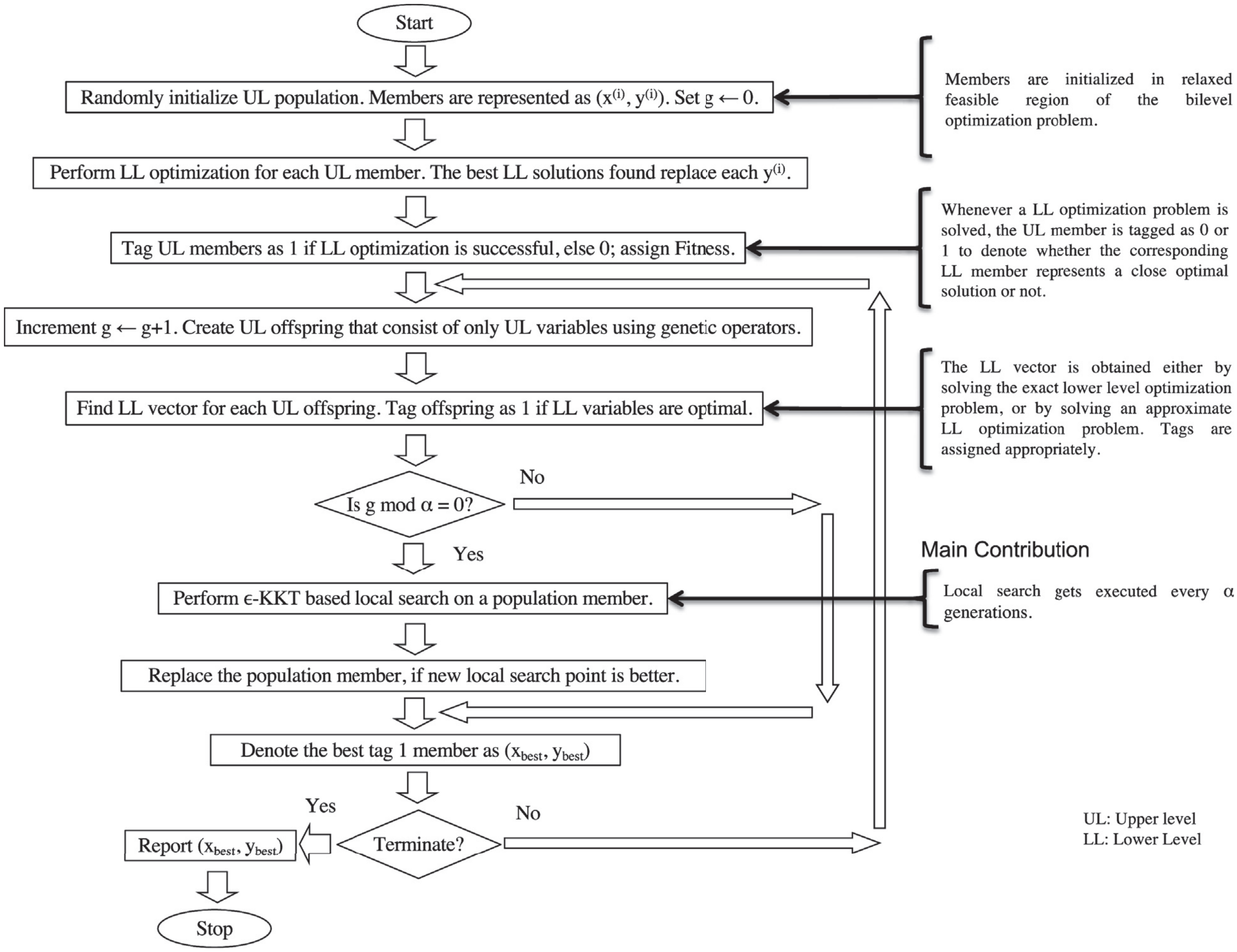

Fig. 3. Flowchart showing the important steps of the ϵ-KKT algorithm.

$\mu = 3$ (Number of parents for crossover)

$\lambda = 2$ (Number of offspring)

4.4. Lower level search

The lower level optimization, whenever required, is done using SQP or an evolutionary algorithm. For details about switching between SQP and evolutionary algorithms for lower level optimization we refer the readers to [50].

4.5. Genetic operators and fitness assignment

The genetic operations at the upper level are performed only with upper level variables, and at the lower level only with respect to the lower level variables. The genetic operations and fitness assignment used in this algorithm are same as that in Ref. [50].

4.6. Termination

The lower level optimization task, whenever performed, is terminated based on the ϵ-KKT idea as discussed in Section 3.1. The termination at lower level occurs if $\epsilon \leq \epsilon_0$ is encountered during the optimization process. At the upper level, the termination is based on improvement at the upper level. If the function value does not improve for 20 consecutive generations, the upper level search is terminated and the algorithm stops. For all the results reported in the paper, we consider the run successful only if it achieves an accuracy of 10^{-2} in the function value at the upper level and lower level individually.

5. Results

To begin with, we study the performance of the ϵ-KKT based algorithm where KKT proximity measure is used as a constraint parameter (Section 3.2). The algorithm is tested on two test suites and the results are compared against a recently proposed bilevel evolutionary algorithm based on quadratic approximations (BLEAQ) [44,45,50] and its extension BLEAQ-II [43]. Thereafter, we evaluate a slightly different approach where ϵ-KKT proximity measure is used as a penalty term (Section 3.3). Finally, we compare the proposed methods against a couple of other methods available in the literature. For all the results reported in this section, we have performed 31 runs with different initial populations on each of the test problems considered in this study.

5.1. TP-suite

The first test suite, that we refer as TP, is a set of 8 bilevel optimization problems taken from the literature [2,4,9,33,38,39,60] for which the best solutions are available. The problem definitions are provided in Tables 3 and 4. The results from multiple runs showing minimum, median and maximum function evaluations required at upper and lower levels to solve the bilevel optimization problems are provided in Fig. 4. In Tables 5 and 6, we perform a comparative study of the ϵ-KKT algorithm against BLEAQ and BLEAQ-II, respectively, where

Table 3
Standard test problems TP1-TP4. (Note that $x = x_u$ and $y = x_l$).

Problem	Formulation	Best Known Sol.
TP1 $n = 2$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = (x_1-30)^2 + (x_2-20)^2 - 20y_1 + 20y_2$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = (x_1-y_1)^2 + (x_2-y_2)^2 \\ 0 \le y_i \le 10, \quad i=1,2 \end{array} \right\}$, $x_1 + 2x_2 \ge 30, x_1 + x_2 \le 25, x_2 \le 15$	$F = 225.0$, $f = 100.0$
TP2 $n = 2$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2 \\ x_1 - 2y_1 \ge 10, x_2 - 2y_2 \ge 10 \\ -10 \ge y_i \ge 20, \quad i=1,2 \end{array} \right\}$, $x_1 + x_2 + y_1 - 2y_2 \le 40$, $0 \le x_i \le 50, \quad i=1,2.$	$F = 0.0$, $f = 100.0$
TP3 $n = 2$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = -(x_1)^2 - 3(x_2)^2 - 4y_1 + (y_2)^2$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = 2(x_1)^2 + (y_1)^2 - 5y_2 \\ (x_1)^2 - 2x_1 + (x_2)^2 - 2y_1 + y_2 \ge -3 \\ x_2 + 3y_1 - 4y_2 \ge 4 \\ 0 \le y_i, \quad i=1,2 \end{array} \right\}$, $(x_1)^2 + 2x_2 \le 4$, $0 \le x_i, \quad i=1,2$	$F = -18.6787$, $f = -1.0156$
TP4 $n = 2$, $m = 3$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = x_1 + 2x_2 + y_1 + y_2 + 2y_3 \\ y_2 + y_3 - y_1 \le 1 \\ 2x_1 - y_1 + 2y_2 - 0.5y_3 \le 1 \\ 2x_2 + 2y_1 - y_2 - 0.5y_3 \le 1 \\ 0 \le y_i, \quad i=1,2,3 \end{array} \right\}$, $0 \le x_i, \quad i=1,2$	$F = -29.2$, $f = 3.2$

the median function evaluations required by the two methods at the upper and lower level have been compared separately. We provide three comparison metrics in Table 5. The ratio corresponding to the ‘Upper’ column denotes the ratio of upper level function evaluations required by ϵ-KKT algorithm against BLEAQ. The ratio corresponding to the ‘Lower’ column denotes the ratio of lower level function evaluations required by ϵ-KKT algorithm against BLEAQ. The ratio corresponding to the ‘Upper + Lower’ column denotes the ratio of sum of upper and lower level function evaluations required by ϵ-KKT algorithm against BLEAQ. Whenever the ratio in any of the three columns is less than 1, it means that ϵ-KKT fares better than BLEAQ, otherwise BLEAQ is better than ϵ-KKT. It is interesting to observe that the proposed approach requires more upper level function evaluations than BLEAQ on many of the test problems. However, the number of lower level function evaluations required by ϵ-KKT are significantly lower than BLEAQ. The proposed approach provides a better overall performance on TP1-TP3 and TP5-TP8. It fares slightly worse on TP7. However, a comparison with BLEAQ-II in Table 6 shows that the proposed approach performs much worse on most of the test problem, except TP5. As mentioned in the earlier section, ϵ-KKT algorithm requires sample points around population members, which increases the overheads at the upper level leading to higher function evaluations and it makes it difficult for ϵ-KKT method to beat BLEAQ-II on these simple problems.

TP-suite consists of problems that are linear or quadratic in structure; therefore, a local approximation represents the exact problem and does not offer much benefit. However, the local search strategy is significantly useful. As soon as the first local search is performed after α generations of evolutionary search, most of these problems are solved to optimality. It is noteworthy that though the evolutionary search looks unnecessary for these problems, it might not necessarily be the case.

Table 4
Standard test problems TP5-TP8. (Note that $x = x_u$ and $y = x_l$).

Problem	Formulation	Best Known Sol.
TP5 $n = 2$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = rt(x)x - 3y_1 - 4y_2 + 0.5t(y)y$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = 0.5t(y)hy - t(b(x))y \\ -0.333y_1 + y_2 - 2 \leq 0 \\ y_1 - 0.333y_2 - 2 \leq 0 \\ 0 \leq y_i, \quad i = 1,2 \end{array} \right\}$, where $h = \begin{pmatrix} 1 & 3 \\ 3 & 10 \end{pmatrix}, b(x) = \begin{pmatrix} -1 & 2 \\ 3 & -3 \end{pmatrix} x, r = 0.1$; $t(\cdot)$ denotes transpose of a vector	$F = -3.6$ $f = -2.0$
TP6 $n = 1$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = (x_1 - 1)^2 + 2y_1 - 2x_1$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = (2y_1 - 4)^2 + \\ (2y_2 - 1)^2 + x_1y_1 \\ 4x_1 + 5y_1 + 4y_2 \leq 12 \\ 4y_2 - 4x_1 - 5y_1 \leq -4 \\ 4x_1 - 4y_1 + 5y_2 \leq 4 \\ 4y_1 - 4x_1 + 5y_2 \leq 4 \\ 0 \leq y_i, \quad i = 1,2 \end{array} \right\}$, $0 \leq x_1$	$F = -1.2091$ $f = 7.6145$
TP7 $n = 2$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = -\frac{(x_1 + y_1)(x_2 + y_2)}{1 + x_1y_1 + x_2y_2}$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = \frac{(x_1 + y_1)(x_2 + y_2)}{1 + x_1y_1 + x_2y_2} \\ 0 \leq y_i \leq x_i, \quad i = 1,2 \end{array} \right\}$, $(x_1)^2 + (x_2)^2 \leq 100$; $x_1 - x_2 \leq 0$; $0 \leq x_i, \quad i = 1,2$	$F = -1.96$ $f = 1.96$
TP8 $n = 2$, $m = 2$	$\underset{(x,y)}{\text{Minimize}}\ F(x,y) = \lvert 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 \rvert$, s.t. $y \in \underset{(y)}{\text{argmin}} \left\{ \begin{array}{c} f(x,y) = (y_1 - x_1 + 20)^2 + \\ (y_2 - x_2 + 20)^2 \\ 2y_1 - x_1 + 10 \leq 0 \\ 2y_2 - x_2 + 10 \leq 0 \\ -10 \leq y_i \leq 20, \quad i = 1,2 \end{array} \right\}$, $x_1 + x_2 + y_1 - 2y_2 \leq 40$; $0 \leq x_i \leq 50, \quad i = 1,2$	$F = 0.0$ $f = 100.0$

Bilevel problems with simple structures can also be non-convex, and therefore, a local search alone will not be able to find optimum. For instance, consider TP2, where we applied local search alone without utilizing the evolutionary search. A different starting point was used for 100 different runs. The 100 different searches terminated on three different points in the search space, out of which 1 was the right bilevel optimum. The bilevel optimum and the local points for this problem are provided in Table 7. Such an approach was able to find the correct bilevel optimum in only 2 of the 100 different searches. However, when the local search is embedded in an evolutionary algorithm,

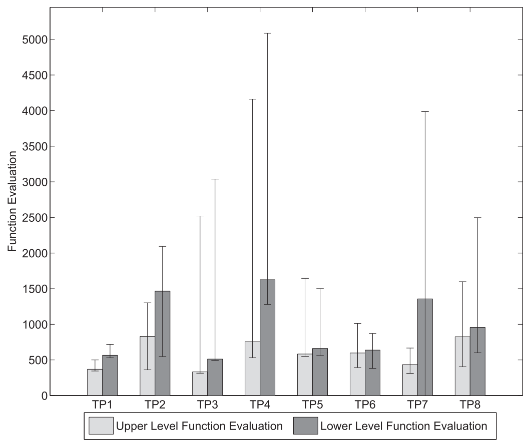

Fig. 4. Results of TP test problems showing the minimum, median and maximum function evaluations.

it could find the bilevel optimum in 70% of the runs, while for all the other test cases, the proposed algorithm had a 100% rate of success.

5.2. SMD-suite

SMD test problems [46] are a set of scalable test problems, where the bilevel optimum for each test problem is known exactly. A lot of information about the structure of the test problems is available to a developer that allows one to test the performance of the algorithm

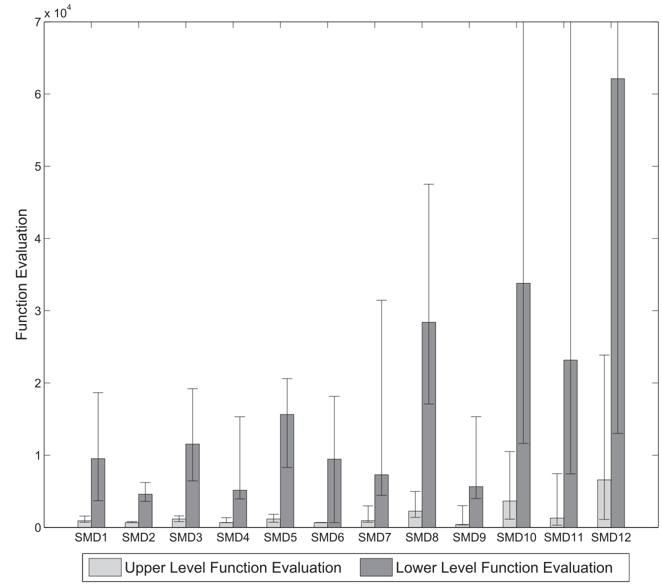

Fig. 5. Results of SMD test problems with 2 upper level variables and 3 lower level variables showing the minimum, median and maximum function evaluations.

Table 5
TP test suite results for ϵ-KKT and comparison with BLEAQ.

Problem	UL Func. Eval. (Med.)		LL Func. Eval. (Med.)		Ratio		
	ϵ-KKT	BLEAQ	ϵ-KKT	BLEAQ	Upper	Lower	Overall
TP 1	369	265	565	1276	1.39	0.44	**0.61**
TP 2	830	870	1465	2295	0.95	0.64	**0.73**
TP 3	333	256	512	1239	1.30	0.41	**0.57**
TP 4	756	358	1626	1856	2.11	0.88	1.08
TP 5	583	867	659	2361	0.67	0.28	**0.38**
TP 6	597	289	638	3270	2.07	0.20	**0.35**
TP 7	434	368	1357	2878	1.18	0.47	**0.55**
TP 8	826	692	956	2176	1.19	0.44	**0.62**

Table 6
TP test suite results for ϵ-KKT and comparison with BLEAQ-II.

Problem	UL Func. Eval. (Med.)		LL Func. Eval. (Med.)		Ratio		
	ϵ-KKT	BLEAQ-II	ϵ-KKT	BLEAQ-II	Upper	Lower	Overall
TP 1	369	244	565	340	1.51	1.66	1.60
TP 2	830	350	1465	610	2.37	2.40	2.39
TP 3	333	255	512	360	1.31	1.42	1.37
TP 4	756	260	1626	940	2.91	1.73	1.99
TP 5	583	392	659	1080	1.49	0.61	**0.84**
TP 6	597	257	638	440	2.32	1.45	1.77
TP 7	434	170	1357	690	2.55	1.97	2.08
TP 8	826	320	956	590	2.58	1.62	1.96

Table 7
Results on TP2 (100 starting points) when ϵ-KKT local search is performed alone using SQP without embedding in the evolutionary approach.

	UL Vector		LL Vector		UL Func. Val.	LL Func. Val.	Count (out of 100)
Local	25	30	5	10	5	0	97
Local	0	0	−10	−10	0	200	1
Global	0	30	−10	10	0	100	2

on different difficulty landscapes. In case the algorithm fails on any of the test problems, one can readily ascertain the weaknesses in the algorithm and attempt to rectify it. In this section, we evaluate the ϵ-KKT algorithm on SMD test problems. The study is performed in three parts; firstly, on small SMD instances consisting of five variables, two at upper level and three at lower level; secondly, on larger SMD instances consisting of ten variables, five at upper level and five at lower level, and thirdly, on even larger SMD instances consisting of 20 variables, ten at upper level and 10 at lower level. Fig. 5 provides the details of function evaluations required from multiple runs for the five variable instances. Tables 8 and 9 provide a comparison of ϵ-KKT against BLEAQ and BLEAQ-II, respectively, on the five variable instances. We observe that ϵ-KKT method is able to beat BLEAQ on all SMD test problems, with quite significant margins on problems SMD2, SMD4, SMD7, SMD9 and SMD11. When compared agains BLEAQ-II, we observe that ϵ-KKT is able to beat the algorithm only on problems SMD2, SMD3, SMD4, SMD5, SMD7, SMD9 and SMD11. It fares worse on other SMD test problems.

For ten variables SMD problems, the results are presented through Fig. 6 and Table 10. Table 10 provides a comparison of ϵ-KKT against both BLEAQ and BLEAQ-II. Interestingly, none of the algorithms could solve all the test problems to the desired accuracy of 10^{-2} with respect to upper and lower level functions. ϵ-KKT could solve SMD1-SMD9, BLEAQ could solve SMD1-SMD6 and BLEAQ-II could solve SMD1-SMD8. Interestingly, ϵ-KKT method beats both the algorithms on all the test problems. It is quite clear that ϵ-KKT performs much better than BLEAQ on SMD1-SDM6, while BLEAQ could not even find the optimal solution to the desired accuracy for SMD7-SMD9. While BLEAQ-II perfoms slightly better than BLEAQ, the comparison results are still favourable for ϵ-KKT on all the test problems that could be solved.

Next, we solve 20-variables instances of SMD test problems, for which the results are presented in Table 11. None of the algorithms could solve SMD8-SMD12. Moreover, even for SMD1-SMD7, the success rate was not 100% on some of the test problems for all the algorithms. This shows that there is a significant increase in difficulty with an increase in the dimensionality. The results once again are favourable for ϵ-KKT method as it outperforms both BLEAQ and BLEAQ-II on SMD1-SMD7 test problems significantly. The overall ratio shows that ϵ-KKT offers enormous savings in function evaluations particularly at the lower level for these problems. The overall ratio in the last columns of Tables 8–11 essentially represents the complement of savings. However, the success rate of ϵ-KKT is poorer as compared to BLEAQ and BLEAQ-II for some of the test problems, as shown in Table 12.

5.3. Importance of local search

For problems SMD1 and SMD2 with 10-dimensions we provide the convergence plots with random initialization for two cases; when run

Table 8
Comparison of ϵ-KKT against BLEAQ on SMD test problems with two upper level variables and three lower level variables. The (*) in the table denotes that the SMD 6 lower level objective function is rewritten as $f^{\mathrm{NE}}(x,y) = f(x,y) + \nu F(x,y)$, where $\nu = 1e-6$ to take care of the multiple optimal solutions arising from the lower level.

Problem	UL Func. Eval. (Med.)		LL Func. Eval. (Med.)		Ratio ($\frac{\epsilon-\text{KKT}}{\text{BLEAQ}}$)		
	ϵ-KKT	BLEAQ	ϵ-KKT	BLEAQ	Upper	Lower	Overall
SMD 1	912	156	9518	15456	5.85	0.62	**0.67**
SMD 2	668	168	4593	14175	3.98	0.32	**0.37**
SMD 3	1158	256	11532	18366	4.52	0.63	**0.68**
SMD 4	668	215	5156	16362	3.11	0.32	**0.35**
SMD 5	1159	156	15644	24368	7.43	0.64	**0.69**
SMD 6*	668	155	9452	14688	4.31	0.64	**0.68**
SMD 7	931	177	7292	16468	5.26	0.44	**0.49**
SMD 8	2248	385	28406	39246	5.84	0.72	**0.77**
SMD 9	410	229	5653	18346	1.79	0.31	**0.33**
SMD 10	3648	549	33803	42312	6.64	0.80	**0.87**
SMD 11	1294	389	23165	143246	3.33	0.16	**0.17**
SMD 12	6572	663	62126	132134	9.91	0.47	**0.52**

Table 9
Comparison of ϵ-KKT against BLEAQ-II on SMD test problems with two upper level variables and three lower level variables.

Problem	UL Func. Eval. (Med.)		LL Func. Eval. (Med.)		Ratio ($\frac{\epsilon-\text{KKT}}{\text{BLEAQ-II}}$)		
	ϵ-KKT	BLEAQ-II	ϵ-KKT	BLEAQ-II	Upper	Lower	Overall
SMD 1	912	191	9518	9511	4.77	1.00	1.07
SMD 2	668	210	4593	9020	3.18	0.51	**0.57**
SMD 3	1158	731	11532	14693	1.58	0.78	**0.82**
SMD 4	668	489	5156	11590	1.37	0.44	**0.48**
SMD 5	1159	208	15644	19238	5.57	0.81	**0.86**
SMD 6*	668	213	9452	9792	3.13	0.97	1.01
SMD 7	931	325	7292	10803	2.86	0.67	**0.74**
SMD 8	2248	1155	28406	24669	1.95	1.15	1.19
SMD 9	410	382	5653	12231	1.07	0.46	**0.48**
SMD 10	3648	1071	33803	24361	3.41	1.39	1.47
SMD 11	1294	2645	23165	121487	0.49	0.19	**0.20**
SMD 12	6572	265	62126	25410	24.78	2.44	2.68

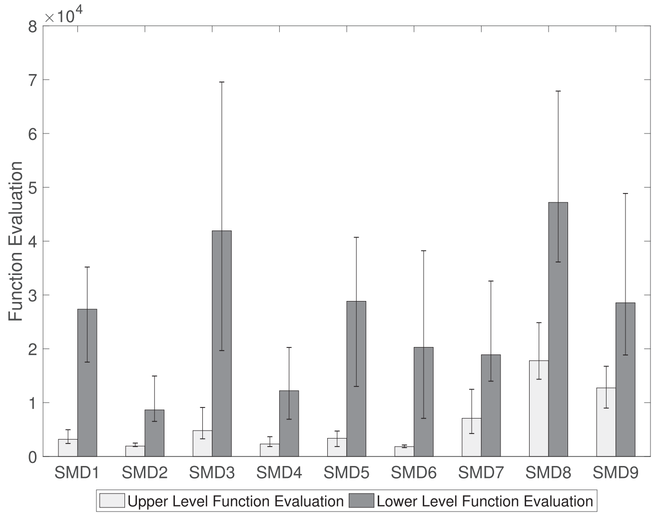

Fig. 6. Results of SMD test problems with five upper level variables and five lower level variables showing the minimum, median and maximum function evaluations.

with local search and when run without local search. In the convergence plots, the x-axis represents number of generations and the y-axis represents the distances of the best feasible point from the bilevel optimum. Figs. 7 and 9 show the convergence plots when ϵ-KKT based local search is implemented and Figs. 8 and 10 show the convergence plots when ϵ-KKT based local search is not implemented. The importance of local search is very clear through these plots, as the cases with no local search do not converge to the bilevel optimum even after a large number of generations.

Table 12
Success rate of ϵ-KKT, BLEAQ and BLEAQ-II approaches when executed on 20 variable SMD test problems.

Prob.	ϵ-KKT	BLEAQ	BLEAQ-II
SMD 1	100%	100%	100%
SMD 2	100%	100%	100%
SMD 3	35%	58%	71%
SMD 4	26%	71%	84%
SMD 5	10%	65%	77%
SMD 6*	100%	100%	100%
SMD 7	29%	0%	10%

Success rate is 0% for all other SMD problems.

5.4. Results with KKT proximity measure as a penalty term

Next, we provide the results when the KKT proximity measure is used as a penalty term. We refer to this method as ϵp-KKT in our discussions. All the steps of the algorithm remain the same, apart from the implementation of local search. Whenever local search is performed we do it multiple times with different parameter values of the penalty parameter r. The penalty parameter r is changed in an ordered manner from the set $\{1, 10, 100, 1000\}$. We continue to use SQP for local search, however, we rewrite the constraint $\epsilon(x, y) = \max(\|\nabla_y L(x, y, \lambda)\|^2, -\sum_{j=1}^{J} \lambda_j g_j(x, y))$ as

$$\|\nabla_y L(x, y, \lambda)\|^2 \le \epsilon,$$

$$-\sum_{j=1}^{J} \lambda_j g_j(x, y) \le \epsilon$$

Table 10
Comparison of ϵ-KKT against BLEAQ and BLEAQ-II on SMD test problems with five upper level variables and five lower level variables.

Prob.	UL Func. Eval. (Med.)			LL Func. Eval. (Med.)			Ratio (Overall)	
	ϵ-KKT	BLEAQ	BLEAQ-II	ϵ-KKT	BLEAQ	BLEAQ-II	$\frac{\epsilon-\text{KKT}}{\text{BLEAQ}}$	$\frac{\epsilon-\text{KKT}}{\text{BLEAQ-II}}$
SMD 1	3187	466	865	27367	74268	62289	**0.41**	**0.48**
SMD 2	1932	362	527	8649	71876	57501	**0.15**	**0.18**
SMD 3	4810	720	1399	41925	99269	77209	**0.47**	**0.59**
SMD 4	2317	499	616	12221	64746	37769	**0.22**	**0.38**
SMD 5	3374	467	527	28835	93540	62360	**0.34**	**0.51**
SMD 6*	1838	2165	700	22680	71928	61139	**0.33**	**0.40**
SMD 7	7090	–	1589	18899	–	85387	–	**0.30**
SMD 8	17798	–	7398	47196	–	240764	–	**0.26**
SMD 9	12735	–	–	28546	–	–	–	–

All algorithms failed to achieve the required threshold for SMD10-SMD12 in all the 31 runs.

Table 11
Comparison of ϵ-KKT against BLEAQ and BLEAQ-II on SMD test problems with ten upper level variables and ten lower level variables.

Prob.	UL Func. Eval. (Med.)			LL Func. Eval. (Med.)			Ratio (Overall)	
	ϵ-KKT	BLEAQ	BLEAQ-II	ϵ-KKT	BLEAQ	BLEAQ-II	$\frac{\epsilon-\text{KKT}}{\text{BLEAQ}}$	$\frac{\epsilon-\text{KKT}}{\text{BLEAQ-II}}$
SMD 1	10,897	3866	4515	1,65,874	5,27,126	5,05,507	**0.33**	**0.35**
SMD 2	11,284	4646	4802	1,01,234	5,57,719	4,91,248	**0.20**	**0.23**
SMD 4	15,290	2419	2559	89,845	3,23,456	2,23,273	**0.32**	**0.47**
SMD 5	34,870	6361	6763	1,68,492	5,91,633	4,74,161	**0.34**	**0.42**
SMD 6*	10,102	2426	2647	1,32,456	5,55,897	3,34,876	**0.26**	**0.42**
SMD 7	23,209	–	5678	1,42,948	–	5,64,784	–	**0.29**

All algorithms failed to achieve the required threshold for SMD8-SMD12 in all the 31 runs.

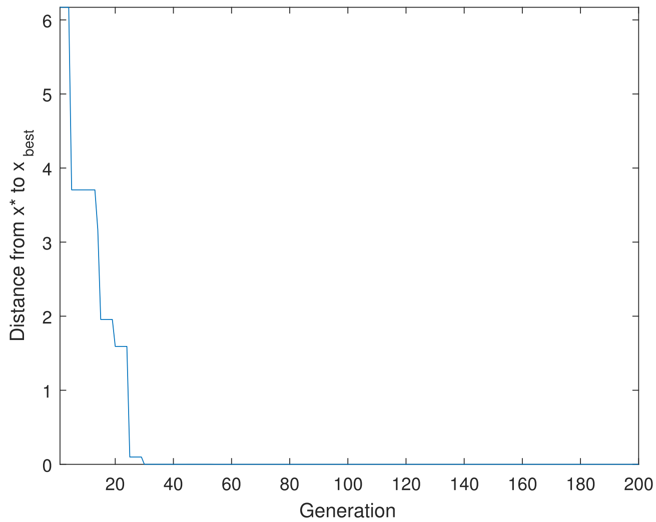

Fig. 7. Euclidean distance of the best solution at every generation from bilevel optimum for 10-dimensional SMD1 test problem when run with local search.

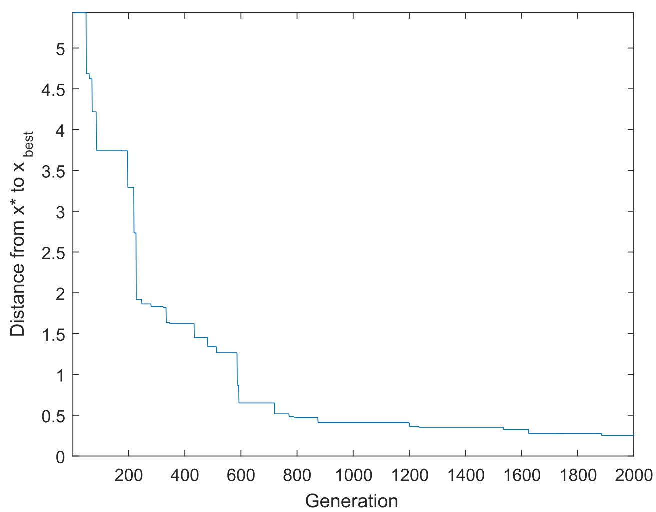

Fig. 8. Euclidean distance of the best solution at every generation from bilevel optimum for 10-dimensional SMD1 test problem when run without local search.

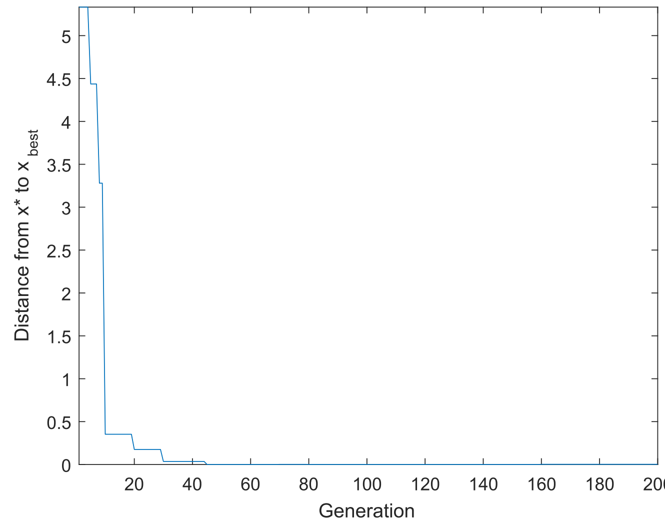

Fig. 9. Euclidean distance of the best solution at every generation from bilevel optimum for 10-dimensional SMD2 test problem when run with local search.

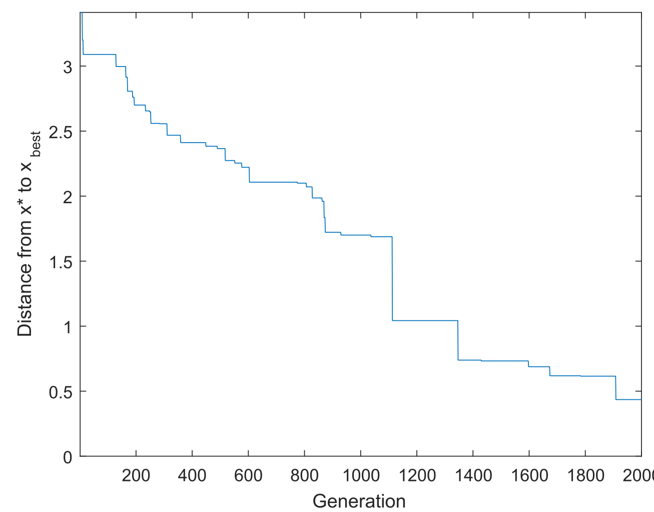

Fig. 10. Euclidean distance of the best solution at every generation from bilevel optimum for 10-dimensional SMD2 test problem when run without local search.

to remove the non-differentiability introduced by "max" within the constraint. The ϵp-KKT method was tested on the SMD test suite, for which we provide the results in Table 13. A comparison of ϵ-KKT results against ϵp-KKT reveals a slightly better performance for ϵp-

Table 13
SMD test suite results for two-variable upper level dimension and three-variable lower level dimension for ϵp-KKT.

Problem	UL Func. Eval.			LL Func. Eval.			Ratio (Overall)
	Min	Median	Max	Min	Median	Max	$\frac{\epsilon p\text{-KKT}}{\epsilon\text{-KKT}}$
SMD 1	661	826	1936	5161	8196	11177	**0.87**
SMD 2	681	972	2980	4982	8315	13439	1.77
SMD 3	743	1200	2732	7377	11421	15032	**0.99**
SMD 4	661	661	907	4786	5808	8113	1.11
SMD 5	702	1465	2471	8043	12213	17624	**0.81**
SMD 6	639	784	2126	751	1067	1613	**0.18**
SMD 7	685	1222	14480	5323	9083	38226	1.25
SMD 8	1223	2807	5676	11174	21916	33832	**0.81**
SMD 9	406	1055	4537	4909	9578	18016	1.75
SMD 10	496	1616	9463	8239	11944	36679	**0.36**
SMD 11	309	522	3657	6354	11072	32596	**0.47**
SMD 12	518	3016	66585	9394	21991	166718	**0.36**

Table 14
Mean of total function evaluations (UL evaluations + LL evaluations) required by different approaches.

	Mean Func. Evals. (UL + LL)					
	ϵ-KKT	ϵp-KKT	BLEAQ	BLEAQ-II	WJL [59]	WLD [60]
TP 1	974	**883**	2421	923	85,499	86,067
TP 2	2843	3192	3262	**1615**	256,227	171,346
TP 3	1184	**934**	1482	987	92,526	95,851
TP 4	4476	3290	6721	**2664**	291,817	211,937
TP 5	**1383**	1553	3388	1618	77,302	69,471
TP 6	1362	**902**	1034	1104	163,701	65,942
TP 7	1908	2354	1456	**1254**	1,074,742	944,105
TP 8	**1932**	2583	3434	2187	213,522	182,121

KKT approach. The approach performs better on SMD1, SMD3, SMD5, SMD6, SMD8, SMD10, SMD11 and SMD12. The reason for the better performance is that the local search is able to generate better lower level optimal solutions in case of ϵp-KKT as compared to ϵ-KKT. Note that in case of ϵ-KKT the local search is likely to terminate with an error of ϵ_0. Next, we compare all the methods together against other benchmarks from the literature.

5.5. Comparison results against other benchmarks

In Table 14, we have provided a comparison of the two proposed techniques against other techniques from the literature. For a straightforward comparison, we have compared the mean of the sum of upper and lower level function evaluations from multiple runs. We have shown in our previous study [50] that BLEAQ is able to significantly outperform the methods WJL [59] and WLD [60]. Interestingly, in tems of mean of overall function evaluations, we observe that the methods based on KKT proximity measure are competitive against both BLEAQ and BLEAQ-II. The ϵ-KKT approach perfoms the best on TP5 and TP8 test problems, while ϵp-KKT performs the best on TP1, TP3 and TP6 test problems. Therefore, better performances are dominated by KKT based procedures. BLEAQ-II dominates other algorithms on TP2, TP4, and TP7.

6. Conclusions

KKT based reduction is a very common approach for reducing bilevel optimization problems to single level before applying the standard optimization techniques on the single level problem. There exist a large number of studies in the classical as well as evolutionary literature that have aimed at such a reduction. Despite such a reduction, a constrained search space makes finding the bilevel optimal solution still a tedious task in many cases. In this paper, we use a relaxed KKT based reduction method for bilevel optimization. The reduction method has been integrated in an evolutionary algorithm and the results show that the approach can be quite competitive to some of the existing bilevel techniques. In particular, it leads to a significant amount of savings in computations for the lower level optimization problem. This study provides a number of directions for future work on bilevel optimization. It should encourage theoretical developments toward solving bilevel optimization problems using approximate KKT method. Moreover, integration of the ideas in this paper with different bilevel mappings, like the lower level reaction set and lower level optimal value function, may lead to new ways of solving bilevel optimization problems. The method has a potential of handling non-linear bilevel problems with a large number of variables; however, this paper has only tested it on non-linear bilevel problems with 20-variables. Future research may also focus on utilizing such techniques to solve non-linear bilevel application problems with large number of variables. We believe that this study on using relaxed KKT conditions for bilevel optimization problems will encourage research along these lines in both the classical and the evolutionary computation literature.

Acknowledgement

Ankur Sinha acknowledges the support provided by SERB (Grant YSS/2015/001370), Department of Science and Technology, India and Research and Publication Office, Indian Institute of Management Ahmedabad for executing this study.

References

[1] E. Aiyoshi, K. Shimizu, Hierarchical decentralized systems and its new solution by a barrier method, IEEE Trans. Syst. Man Cybern. 6 (1981) 444–449.
[2] E. Aiyoshi, K. Shimizu, A solution method for the static constrained Stackelberg problem via penalty method, IEEE Trans. Automat. Contr. 29 (1984) 1111–1114.
[3] F. Al-Khayyal, R. Horst, P. Pardalos, Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming, Ann. Oper. Res. 34 (1992) 125–147.
[4] Mahyar A. Amouzegar, Khosrow Moshirvaziri, Determining optimal pollution control policies: an application of bilevel programming, Eur. J. Oper. Res. 119 (1) (1999) 100–120.
[5] J. Angelo, E. Krempser, H. Barbosa, Differential evolution for bilevel programming, in: Proceedings of the 2013 Congress on Evolutionary Computation (CEC-2013), IEEE Press, 2013.
[6] Jaqueline S. Angelo, Eduardo Krempser, Helio JC. Barbosa, Differential evolution assisted by a surrogate model for bilevel programming problems, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 1784–1791.
[7] J. Bard, J. Falk, An explicit solution to the multi-level programming problem, Comput. Oper. Res. 9 (1982) 77–100.
[8] J. Bard, J. Moore, A branch and bound algorithm for the bilevel programming problem, SIAM J. Sci. Stat. Comput. 11 (1990) 281–292.
[9] J.F. Bard, Practical Bilevel Optimization: Algorithms and Applications, Kluwer, The Netherlands, 1998.
[10] Mokhtar S. Bazaraa, Hanif D. Sherali, Chitharanjan M. Shetty, Nonlinear Programming: Theory and Algorithms, John Wiley & Sons, 2013.
[11] O. Ben-Ayed, Bilevel linear programming, Comput. Oper. Res. 20 (1993) 485–501.
[12] W. Bialas, M. Karwan, Two-level linear programming, Manag. Sci. 30 (1984) 1004–1020.
[13] J. Bracken, J. McGill, Mathematical programs with optimization problems in the constraints, Oper. Res. 21 (1973) 37–44.
[14] Luce Brotcorne, Martine Labbe, Patrice Marcotte, Gilles Savard, A bilevel model for toll optimization on a multicommodity transportation network, Transport. Sci. 35 (4) (2001) 345–358.
[15] G. Brown, M. Carlyle, D. Diehl, J. Kline, K. Wood, A two-sided optimization for theater ballistic missile defense, Oper. Res. 53 (5) (2005) 745–763.
[16] Y. Chen, M. Florian, On the Geometry Structure of Linear Bilevel Programs: a Dual Approach, Technical Report CRT-867, Centre de Recherche sur les Transports, 1992.
[17] Peter A. Clark, Arthur W. Westerberg, Bilevel programming for steady-state chemical process design-i. fundamentals and algorithms, Comput. Chem. Eng. 14 (1) (1990) 87–97.
[18] Benoît Colson, Patrice Marcotte, Gilles Savard, A trust-region method for nonlinear bilevel programming: algorithm and computational experience, Comput. Optim. Appl. 30 (3) (2005) 211–227.
[19] K. Deb, A. Sinha, An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm, Evol. Comput. J. 18 (3) (2010) 403–449.
[20] Stephan Dempe, Foundations of Bilevel Programming, Kluwer Academic Publishers, Secaucus, NJ, USA, 2002.
[21] Joydeep Dutta, Kalyanmoy Deb, Rupesh Tulshyan, Ramnik Arora, Approximate kkt points and a proximity measure for termination, J. Global Optim. 56 (4) (2013) 1463–1499.

[22] T. Edmunds, J. Bard, Algorithms for nonlinear bilevel mathematical programming, IEEE Trans. Syst. Man Cybern. 21 (1991) 83–89.

[23] J. Fortuny-Amat, B. McCarl, A representation and economic interpretation of a two-level programming problem, J. Oper. Res. Soc. 32 (1981) 783–792.

[24] P. Hansen, B. Jaumard, G. Savard, New branch-and-bound rules for linear bilevel programming, SIAM J. Sci. Stat. Comput. 13 (1992) 1194–1217.

[25] S. Reza Hejazi, Azizollah Memariani, G. Jahanshahloo, Mohammad Mehdi Sepehri, Linear bilevel programming solution by genetic algorithm, Comput. Oper. Res. 29 (13) (2002) 1913–1925.

[26] X. Hu, D. Ralph, Using EPECs to model bilevel games in restructured electricity markets with locational prices, Oper. Res. 55 (5) (2007) 809–827.

[27] Y. Ishizuka, E. Aiyoshi, Double penalty method for bilevel optimization problems, Ann. Oper. Res. 34 (1992) 73–88.

[28] Yan Jiang, Xuyong Li, Chongchao Huang, Xianing Wu, Application of particle swarm optimization based on chks smoothing function for solving nonlinear bilevel programming problem, Appl. Math. Comput. 219 (9) (2013) 4332–4339.

[29] C. Kolstad, L. Lasdon, Derivative evaluation and computational experience with large bilevel mathematical programs, J. Optim. Theor. Appl. 65 (1990) 485–499.

[30] M. Labbé, P. Marcotte, G. Savard, A bilevel model of taxation and its application to optimal highway pricing, Manag. Sci. 44 (12) (1998) 1608–1622.

[31] Hecheng Li, A genetic algorithm using a finite search space for solving nonlinear/linear fractional bilevel programming problems, Ann. Oper. Res. (2015) 1–16.

[32] Xiangyong Li, Peng Tian, Xiaoping Min, A hierarchical particle swarm optimization for solving bilevel programming problems, in: Leszek Rutkowski, Ryszard Tadeusiewicz, Lotfi A. Zadeh, Jacek M. Zurada (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2006, Volume 4029 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 1169–1178.

[33] Guoshan Liu, Jiye Han, Shouyang Wang, A trust region algorithm for bilevel programing problems, Chin. Sci. Bull. 43 (10) (1998) 820–824.

[34] Patrice Marcotte, Gilles Savard, D.L. Zhu, A trust region algorithm for nonlinear bilevel programming, Oper. Res. Lett. 29 (4) (2001) 171–179.

[35] R. Mathieu, L. Pittard, G. Anandalingam, Genetic algorithm based approach to bi-level linear programming, Oper. Res. 28 (1) (1994) 1–21.

[36] Athanasios Migdalas, Bilevel programming in traffic planning: models, methods and challenge, J. Global Optim. 7 (4) (1995) 381–405.

[37] M.G. Nicholls, Aluminium production modeling - a nonlinear bilevel programming approach, Oper. Res. 43 (2) (1995) 208–218.

[38] V. Oduguwa, R. Roy, Bi-level optimization using genetic algorithm, in: Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02), 2002, pp. 322–327.

[39] J.V. Outrata, On the numerical solution of a class of Stackelberg problems, Zeitschrift Fur Oper. Res. AC 34 (1) (1990) 255–278.

[40] Martín Safe, Jessica Carballido, Ignacio Ponzoni, Nélida Brignole, On stopping criteria for genetic algorithms, in: Brazilian Symposium on Artificial Intelligence, Springer, 2004, pp. 405–413.

[41] G. Savard, J. Gauvin, The steepest descent direction for the nonlinear bilevel programming problem, Oper. Res. Lett. 15 (1994) 275–282.

[42] Chenggen Shi, Jie Lu, Guangquan Zhang, An extended kuhn–tucker approach for linear bilevel programming, Appl. Math. Comput. 162 (1) (2005) 51–63.

[43] A. Sinha, Z. Lu, K. Deb, P. Malo, Bilevel Optimization Based on Iterative Approximation of Multiple Mappings, arXiv preprint arXiv:1702.03394, 2017.

[44] A. Sinha, P. Malo, K. Deb, Efficient Evolutionary Algorithm for Single-objective Bilevel Optimization, arXiv preprint arXiv:1303.3901, 2013.

[45] A. Sinha, P. Malo, K. Deb, An improved bilevel evolutionary algorithm based on quadratic approximations, in: 2014 IEEE Congress on Evolutionary Computation (CEC-2014), IEEE Press, 2014, pp. 1870–1877.

[46] A. Sinha, P. Malo, K. Deb, Test problem construction for single-objective bilevel optimization, Evolut. Comput. J. 22 (3) (2014) 439–477.

[47] A. Sinha, P. Malo, K. Deb, Transportation policy formulation as a multi-objective bilevel optimization problem, in: 2015 IEEE Congress on Evolutionary Computation (CEC-2015), IEEE Press, 2015.

[48] A. Sinha, P. Malo, K. Deb, Solving optimistic bilevel programs by iteratively approximating lower level optimal value function, in: 2016 IEEE Congress on Evolutionary Computation (CEC-2016), IEEE Press, 2016.

[49] A. Sinha, P. Malo, K. Deb, Approximated set-valued mapping approach for handling multiobjective bilevel problems, Comput. Oper. Res. 77 (2017) 194–209.

[50] A. Sinha, P. Malo, K. Deb, Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping, Eur. J. Oper. Res. 257 (2) (2017) 395–411.

[51] A. Sinha, P. Malo, K. Deb, A review on bilevel optimization: from classical to evolutionary approaches and applications, IEEE Trans. Evol. Comput. 22 (2) (2018) 276–295.

[52] A. Sinha, P. Malo, K. Deb, P. Korhonen, J. Wallenius, Solving bilevel multi-criterion optimization problems with lower level decision uncertainty, IEEE Trans. Evol. Comput. 20 (2) (2016) 199–217.

[53] A. Sinha, P. Malo, A. Frantsev, K. Deb, Multi-objective stackelberg game between a regulating authority and a mining company: a case study in environmental economics, in: 2013 IEEE Congress on Evolutionary Computation (CEC-2013), IEEE Press, 2013.

[54] A. Sinha, P. Malo, A. Frantsev, K. Deb, Finding optimal strategies in a multi-period multi-leader-follower stackelberg game using an evolutionary algorithm, Comput. Oper. Res. 41 (2014) 374–385.

[55] W.R. Smith, R.W. Missen, Chemical Reaction Equilibrium Analysis: Theory and Algorithms, John Wiley & Sons, New York, 1982.

[56] H. Stackelberg, The Theory of the Market Economy, Oxford University Press, New York, Oxford, 1952.

[57] H. Tuy, A. Migdalas, P. Värbrand, A global optimization approach for the linear two-level program, J. Global Optim. 3 (1993) 1–23.

[58] L. Vicente, G. Savard, J. Júdice, Descent approaches for quadratic bilevel programming, J. Optim. Theor. Appl. 81 (1994) 379–399.

[59] Y. Wang, Y.C. Jiao, H. Li, An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 32 (2) (2005) 221–232.

[60] Yuping Wang, Hong Li, Chuangyin Dang, A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence, Inf. J. Comput. 23 (4) (2011) 618–629.

[61] L. Wein, Homeland security: from mathematical models to policy implementation: the 2008 Philip McCord morse lecture, Oper. Res. 57 (4) (2009) 801–811.

[62] U. Wen, S. Hsu, Linear bi-level programming problems - a review, J. Oper. Res. Soc. 42 (1991) 125–133.

[63] D. White, G. Anandalingam, A penalty function approach for solving bi-level linear programs, J. Global Optim. 3 (1993) 397–419.

[64] Y. Yin, Genetic algorithm based approach for bilevel programming models, J. Transport. Eng. 126 (2) (2000) 115–120.

[65] Xiaobo Zhu, Qian Yu, Xianjia Wang, A hybrid differential evolution algorithm for solving nonlinear bilevel programming with linear constraints, in: Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on, vol. 1, IEEE, 2006, pp. 126–131.