Bilevel Direct Search Method for Leader-Follower Equilibrium Problems and Applications

Dali Zhang[1] and Gui-Hua Lin[2]

February 12, 2012

Abstract

In the paper, we propose a bilevel direct search method for the distributed computation of equilibria in leader-follower problems. This type of direct search methods is designed for characterizing the decision making process where the players' objective functions are not analytically available. We investigate the convergence of the accumulation points yielded by the method to the stationary points of the problems. Then, we apply the method to a health insurance problem and carry out several numerical examples to illustrate how the method performs when solving leader-follower problems.

Key words: *Leader-follower equilibrium, direct search method, stationary point, health insurance*

1 Introduction

Bilevel programming problem is a hierarchical optimization problem where a subset of the variables is constrained to be a solution of another optimization or equilibrium problem parameterized by the remaining variables. The leader-follower problem investigated in the paper, as a type of the hierarchical competition, can be considered as a version of the problem first introduced and investigated by the German economist von Stackelberg [35] in 1934, which has received much attention in both economics and game theory.

In recent decades, due to the increasing demand on characterizing the hierarchical structure in the practical settings, the bilevel structural games have been integrated into the realm of operations research. In the time horizon, this type of bilevel programming models can be divided into two stages: At the first stage, the upper level decision makers, i.e. the dominant players, choose their optimal positions and, at the second stage, the lower level decision makers, i.e. the other players, optimize their objectives given the dominant players' positions determined at the first stage. In the game setting, the dominant players at the first stage and the other players at

[1]Centre for Health Services Research, Singapore Health Services Pte Ltd, S150168, Jln Bukit Merah, Singapore. E-mail: zhang.dali@singhealth.com.sg.

[2]School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China. E-mail: lin_g_h@yahoo.com.cn.

the second stage are named as *leaders* and *followers* respectively to differentiate their strategic roles. One of the simplest leader-follower models, *a toll-setting model*, was first investigated by Colson et al. [7] consisting of a leader maximizing her revenue raised from tolls set on some links of a transportation network, and a representative network user acting in a follower role minimizing his travel costs. The entire problem can be represented as a *bilevel optimization model*. In this type of problems, given the rationality of both players, on the one hand, the follower makes his lower level optimal decision by inclusively considering the impact from the leader's decision predetemined before his lower level decision problem; on the other hand, when making the upper level optimal decision, the leader predictively takes the follower's lower level reaction into account.

In most of practical problems, the number of players at non-dominant position is usually more than one, where each player is driven by different objective. Consequently, for this type of problems, the followers' strategic behavior can not be formulated as an optimization model as in [7]. Jointly regarding the rivals' strategic behaviors, the followers' competition can be alternatively characterized essentially by a Nash equilibrium model. By formulating the solutions of the lower level Nash equilibrium problem for followers as the constraints in the leader's optimization problem, DeWolf and Smeers [12] made the first attempt to study a Stackelberg-Nash-Cournot equilibrium model through a mathematical program with equilibrium constraints (MPEC). Taking the random setting with continuous distribution into account, Xu [39] extended the model to a stochastic version and proposed a sample average approximation method to solve it. Another type of leader-follower game problems which has received attention is called two-stage *equilibrium problem with equilibrium constraints* (EPEC), which includes more than one dominant players acting in the leader role. This type of equilibrium problems are recently investigated by Shanbhag [33], and Kulkarni and Shanbhag [19], in which a paradigm is developed for claiming the existence of global equilibria for EPECs with shared constraints. The model reflects the hierarchical structure in some practical Nash equilibrium problems such as stochastic multi-leader Stackelberg-Nash-Cournot models for future market competition [10], EPEC for electricity markets [42, 46, 16], Nash equilibrium model in transportation [27] and signal transmission in wireless networks [27]. More recently, multi-leader multi-follower game models have been extended and applied to solve some practical problems in a uncertain environment. Xu and Zhang [45] explored a two-stage stochastic EPEC model and presented a Monte Carlo scheme to solve the stationary points. The survey paper [23] by Lin and Fukushima summarized the recent development on the stochastic version of single/multi-leader and multi-follower models.

However, the development of the modeling and theoretical results on the leader-follower problems has not been accompanied by an equal improvement in the computational algorithms for solving the stationary points or equilibria of these models. One direction of the research on algorithms is to solve the stationary points by transforming the optimization/equilibrium problems to a set of complementarity inequalities by Xu [40] and Leyffer and Munson [21] where an algorithm is proposed based on the Karush-Kuhn-Tucker (KKT) conditions of bilevel programming problems. Another type of approaches requiring the gradient information includes a smoothing projected gradient method for stochastic linear complementarity problems proposed by Zhang and Chen [44], and a quasi-Newton method for MPECs proposed by Jiang and Ralph [17]. One prerequisite condition of using these aforementioned methods is the availability of the analytical forms of the objective functions or local gradient information at both upper and lower levels.

In recent years, with growing concerns from practical perspective, more and more attention have been received for the optimization problems where the analytical forms and local gradient information of objective functions are not available. The word "black-box" has been used to characterize this type of problems. In the literature, to deal with the black-box optimization problems, a set of derivative free methods has been proposed; see Kolda et al. [18] and Lewis et al. [20]. In Section 3, we present a survey on the derivative free methods to introduce how the methods are developed over the two decades for general optimization problems.

In this paper, we propose a distributed direct search algorithm for solving a leader-follower problem with black-box objective functions for the decision makers at both levels. One feature of this algorithm is that, at each iteration, the decision makers in the game unilaterally select his/her descent directions, which well represents the natural process of the negotiation and decision making process in the practical leader-follower problems and enables parallel and distributed computation of equilibria. Another reason that the distributed direct search algorithm is important and highly relevant in the computation of equilibria, is particularly because that they could be implemented in a real world game situation where the game is incomplete, i.e., the decision makers do not have to know each other's cost functionals and parameters, and only have to communicate to each other their tentative decisions during each phase of computation. Both the reasons carry computational advantages over centralized schemes. One of the pioneer work on the distributed algorithm was done by Papavassilopoulos [28] for quadratic adaptive Nash games, and the first work related to the most important concept in the algorithm, contraction mapping theorem, has been explored by Bertsekas [3]. Since then, the algorithm was investigated by Li and Basar [22] to a class of nonquadratic convex Nash games for yielding the unique stable Nash equilibrium, and implemented by Bozma [5] into a parallel gradient descent framework which is applicable to the practical computation of Nash equilibria. More recently, the distributed and parallel computation idea has been further extended. A variation of distributed algorithms was proposed by Yuan [43] for deterministic Nash equilibrium problems, where the gradient descent with approximately updated by a trust region scheme.

The purpose of the paper is to propose a framework of the distributed algorithms for solving the leader-follower problem. For each decision making process in the bilevel problem, a direct search algorithm has been used to determine descent direction under the condition that the gradients of the objective functions are not locally available. Despite of being a standard option for solving the black-box optimization problems, the implementation of the direct search, or broadly derivative free methods, is still very limited to the bilevel optimization or equilibrium problems. To our knowledge, the first trial of applying the direct search algorithms to bilevel optimization problems was explored by Mersha and Dempe [24]. In a set of more recent results, Vicente and

Custódio [36] investigated a set of direct search algorithms for discontinuous functions. This is the first investigation where direct search methods were globalized using sufficient decrease and a set of directions dense in the unit sphere. Moreover, Conn and Vicente [8] proposed a derivative free optimization method to a robust optimization in which the analysis on the methods was accommodated within interpolation-based trust region framework. In the literature, some derivative free methods, including direct search method and trust region method, can be so far used for *partial* black-box bilevel problems, where the reason of using the term *partial* is because that the approaches in [24, 7] implicitly requires the knowledge of the followers' objective functions or equivalently the analytical form of the solutions at the second stage.

In the study, we step further to propose a distributed direct search method to solve the bilevel problems where the close forms of objective functions at both stages are of black-box form and subsequently the analytical form of the lower level equilibrium is not necessarily available. As aforementioned, this algorithm could be implemented in a real world game situation where the decision makers do not have to know each other's cost functionals and parameters, and only have to communicate to each other their tentative decisions during each phase of computation. This direct search approach is essentially designed in a hierarchical structure to characterize the interactions between the leaders and the followers. The algorithm particularly projects decision makers behaviors to a bilevel game setting: To estimate the followers' reaction, a lower level direct search algorithm is used to characterize the followers' decision making process where the objective functions are of a black-box form. Then, an upper level direct search algorithm is carried out to iteratively solve the leaders' decision problems by integrating the estimation of the followers' reaction yielded by into consideration.

The contribution of this paper can be concluded into following points: (a) We propose a distributed bilevel direct method to solve a "black-box" leader-follower problem. (b) In technique, we extend the research on the bilevel derivative free methods to the problems where the optimal value functions and solutions are not analytically available. (c) In theoretical results, we show the convergence of this type of bilevel direct search methods. (d) In application, we apply the method to a health insurance problem in which a leader-follower model is presented to characterize the interactions between a hospital administrator and a representative physician.

The rest of the paper is organized as follows. In Section 2, we present a description on a typical leader-follower problem. Some properties on the Clarke stationary points are given in the section. In Section 3, we propose a bilevel direct search method for solving the leader-follower problem with black-box objectives. The convergence analysis on this bilevel method is presented in Section 4. Finally, in Section 5, we apply the algorithms to a hospital competition problem under regulated price first proposed by Eggleston and Yip [13]. In this section, two computational experiments, a single-follower model and a multi-follower model, are carried out, where for the latter model, we replace the direct search algorithm at the lower level by a trust region algorithm by Yuan [43] for solving the equilibrium for the followers' competition. By doing so, we show a broader set of bilevel algorithms.

2 The Leader-Follower Problem

In this section, we start our investigation from a description on leader-follower competition problem which essentially can be seen as a Stackelberg game or a bilevel programming problem.

2.1 The model

We consider a bilevel programming problem where the upper level decision maker, the leader, controls her decision variable $x \in \mathcal{X} \subset \mathbb{R}^n$, and the lower level decision maker, the follower, controls his decision variable $y \in \mathcal{Y} \subset \mathbb{R}^m$. We underscore that each player wants to optimize her or his respective objectives. To this end, the decision makers choose their optimal strategies in a non-collaborative way, and presents a hierarchical structure: When making the decision, the feasible set of the leader's problem is restricted in part to the solution set mapping from another optimization problem, which is the empirical projection of the follower's rational reaction.

Now let us look at the follower's lower level decision process or reaction to the leader's. Given that a strategy $x \in \mathcal{X}$ is chosen by the leader in the upper level, the follower makes his optimal decision $y(x) \in \mathcal{Y}$ by solving the optimization problem

$$\min_{y \in \mathcal{Y}} \quad f(x, y), \tag{2.1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is sufficiently smooth, $\mathcal{X}$ and $\mathcal{Y}$ are the subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. Given a fixed value x at the upper level, to characterize the follower's decision on the lower level programming, we define the solution set

$$\Psi(x) := \left\{ y \in \mathcal{Y} \,\middle|\, y \text{ solves (2.1)} \right\}. \tag{2.2}$$

Now, let us consider the bilevel programming problem from the leader's point of view. If the leader is rationally enough, she can infer the follower's optimizaton behavior in his decision making process and expect an outcome $y(x)$ in $\Psi(x)$ of the lower level. Thus, the leader's decision problem can be characterized as the following constrained optimization problem:

$$\left\{ \begin{array}{ll} \min & F(x, y(x)) \\ \text{s.t.} & x \in \mathcal{X}, \\ & y(x) \in \Psi(x), \end{array} \right. \tag{2.3}$$

where $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is locally Lipschitz continuous near every point relevant to the discussion at hand. This single leader-follower programming problem has been first investigated by Von Stackelberg [35] named as the *Stackelberg competition.* The model has been extended to single-leader multiple-follower programming problem by Sherali, Soyster, and Murphy [34] and DeWolf and Smeers [12].

Here, a point is needed to be clarified: If the solution of the lower level problem (2.1) corresponding to a parameter x is not unique, i.e., $\Psi(x)$ is not a singleton, then the content of the term '$y(x)$' in the objective function and the constraint of problem (2.3) is ambiguous. For these cases, in practice, what the leader usually do is to select a $y(x)$ from $\Psi(x)$ for making her decision, and naturally this selection behavior is influenced by her attitude, optimistic or pessimistic, torwards the outcome of the follower's decision problem; See Shapiro and Xu [32] and Zhang and Xu [45].

In this paper, what we are concerned is on the computation of equilibrium and an algorithm with a hierarchical and distributed structure which can be adapted to reflect the real negotiation process. Therefore, we concentrate our study on a set of problems with a unique solution at the lower level. To this end, we investigate the bilevel problem with a convex optimization model in the lower level, which can be concluded as the following assumption.

Assumption 2.1 *For any $x \in \mathcal{X}$, $f(x, y)$ is strictly convex with respect to (w.r.t.) $y \in \mathcal{Y}$, and the feasible set $\mathcal{Y}$ is compact and convex.*

Lemma 2.2 *Under Assumption 2.1, there exists a unique solution $y(x)$ to $\min_{y \in \mathcal{Y}} f(x, y)$ for any fixed $x \in \mathcal{X}$, that is, $\Psi(x)$ is a singleton.*

In Assumption 2.1, the strictly convexity of $f(x, y)$ ensures that the follower's decision problem has one optimal solution at most. The related results will be presented in the section of bilevel direct search method.

Now, given the uniqueness of the lower level solution $y(x)$ for any fixed $x \in \mathcal{X}$, we can reformulate the leader's objective function at the upper level as

$$\mathcal{F}(x) := \min_{y \in \Psi(x)} F(x, y),$$

where the set $\Psi(x)$ is singleton. Note that, in literature (see [45]), $\mathcal{F}(x)$ is defined as the optimal value function of the upper level problem. Then the leader's decision problem can be written in an *implicit* form as

$$\min_{x \in \mathcal{X}} \mathcal{F}(x) = \min_{x \in \mathcal{X}} F(x, y(x)), \tag{2.4}$$

where "implicit" means that (2.4) does not include the details of the lower level problem. Under some moderate conditions, (2.4) coincides with (2.3), see Proposition 5 in [31, Chapter 1].

2.2 Lipschitz continuity

In the remainder of this section, we present some assumptions to guarantee the Lipschitz continuity of the optimal value function at the lower level.

First, let us look at the solution $y(x)$ of the lower level optimization problem (2.1). For fixed $x \in \mathcal{X}$, we can write the KKT condition of the lower level problem as follows,

$$0 \in \nabla_y f(x, y) + \mathcal{N}_{\mathcal{Y}}(y), \tag{2.5}$$

where $\mathcal{N}_{\mathcal{Y}}(y)$ is normal cone to $\mathcal{Y}$ at y, which is defined in as follows,

$$\mathcal{N}_{\mathcal{Y}}(y) := \left\{z \in \mathbb{R}^m : z^T(y' - y) \leq 0, \ \ \forall\, y' \in \mathcal{Y}\right\}, \ \text{ if } y \in \mathcal{Y}.$$

The solution to (2.5) is the optimal solution $y(x)$ to the follower's decision problem. What we need to look at is the continuity of the solution to (2.5). Given the sufficiently smoothness of $f(x, y)$, we can straightforwardly have the following properties for the gradient of $f(x, y)$ and Assumption 2.1, that is, for any fixed $x \in \mathcal{X}$,

(C1) $\nabla_y f(x, y)$ is a Lipschitz continuous function of y with a Lipschitz module denoted by M_f;

(C2) $\nabla_y f(x, \cdot)$ is uniformly strongly monotone on $\mathcal{Y}$, that is, there exists a constant $c > 0$ such that, for any given x,

$$\left(\nabla_y f(x, y') - \nabla_y f(x, y)\right)^T (y' - y) \geq c\|y' - y\|^2, \quad \forall\, y',\, y \in \mathcal{Y}. \tag{2.6}$$

Under condition (C2), it follows from [14, Theorem 2.3.3] that the lower level optimization problem (2.1) has a unique solution $y(x)$ for every given x. Moreover, under condition (C1) and (C2), we can show the Lipschitzness of the solution function $y(\cdot)$.

Lemma 2.3 *Under Assumption 2.1, $y(x)$, the solution to (2.1), is a Lipschitz continuous function of x on $\mathcal{X}$.*

The proof of Lemma 2.3 can be seen in [14, Theorem 2.3.3].

Now, let us look at the upper level optimization problem. It is well known that the optimal value function of a parametric optimization problem is often nonconvex. In our context, this means that $\mathcal{F}(x)$ might be nonconvex and consequently we may obtain a local optimal solution or a stationary point by solving the leader's problem (2.3). The concept of stationary points is important in optimization as it provides some information of optimality. This is particularly so in MPEC where obtaining a global optimal solution is often difficult and consequently various of stationary points are investigated [25, 26].

We start with the definition. Based on Assumption 2.1, we have that the optimal value function $\mathcal{F}(x) := F(x, y(x))$ is usually not continuously differentiable even when the functions F and f are sufficiently smooth. Therefore, the concept of the generalized gradient is needed to characterize the first order optimality conditions. Here, we use the *Clarke generalized gradient* for the analysis. The Clarke generalized gradient of the optimal value function $\mathcal{F}$ coincides with the usual gradient at the points where $\mathcal{F}$ is strictly differentiable.

It has been shown that the directional derivative of any locally Lipschitz continuous function exists. The directional derivative is defined as follows: Let g be locally Lipschitz near a given vector x, and let d be a given direction. Then the directional derivative of g at x in the direction d, denoted by $g'(x; d)$, is defined as

$$g'(x; d) := \lim_{t \downarrow 0} \frac{g(x + td) - g(x')}{t}. \tag{2.7}$$

Proposition 2.4 (Theorem 4.8 in [11]) *Under Assumption 2.1, the functions $y(\cdot)$ and $\mathcal{F}$ are directionally differentiable.*

The *Clarke generalized derivative* of g at point x in direction d is defined as

$$g^o(x;d) := \limsup_{x' \to x, t \downarrow 0} \frac{g(x'+td) - g(x')}{t}.$$

The function g is said to be Clarke *regular* at x if the usual one sided directional derivative $g'(x;d)$ exists and $g^o(x;d) = g'(x;d)$ for all $d \in \mathbb{R}^n$. Moreover, from [6, Chapter 2], we have the following definition of *Clarke generalized gradient* (also known as *Clarke subdifferential*):

$$\partial g(x) := \left\{ \zeta \,|\, \zeta^T d \leq g^o(x;d) \right\}.$$

Consequently, we can restate Proposition 2.4 as follows.

Proposition 2.5 (Theorem 4.8 in [11]) *Under Assumption 2.1, the functions $y(\cdot)$ and $\mathcal{F}$ are Clarke directionally differentiable.*

Under Proposition 2.5, we can proceed to the concept of local minimum within a Clarke subdifferential framework. Note that a point x is a local minimum of (2.4) if there exists a neighborhood of x such that $\mathcal{F}(x) \leq \mathcal{F}(x')$ for any $x' \in \mathcal{X}$ in this neighborhood. If x is a local minimum of (2.4), then it must be $\mathcal{F}^o(x;d) \geq 0$ for any feasible direction $d \in \mathbb{R}^m$. This necessary condition motivates the following definition.

Definition 2.6 (Clarke Stationary Point [30]) *A point $x \in \mathcal{X}$ is said to be a Clarke stationary point of $\mathcal{F}(\cdot)$ if for any feasible direction d it holds that $\mathcal{F}^o(x;d) \geq 0$. A vector d is a descent direction of $\mathcal{F}$ at x if $\mathcal{F}^o(x;d) < 0$.*

Remark 2.7 *Similar definitions on Clarke stationarity can be seen in [24] and have been extended to the concept of Nash-C-Stationary points in Nash game setting by Xu and Zhang [41]. From the rsults in [24], we can link the definitions of local minimums and Clarke stationary points: Let the lower level problem (2.1) satisfy Assumption 2.1. If a point (x^*, y^*) with $y^* \in \Psi(x^*)$ is a local minimum of the bilevel programming problem (2.3), then it is a Clarke stationary point for the bilevel programming problem.*

3 Bilevel Distributed Direct Search Method

3.1 Direct search method

Direct search method is one of the best known iterative optimization techniques that do not explicitly use derivative or any approximation of gradients. Hooke and Jeeves coined the phrase "direct search" in [15] and described a sequential examination of trial points by comparing each point with the optimal one obtained up to that time together with a strategic procedure for determining what the next candidate solution will be, where the direction towards the next candidate point is selected from a special finite set. Direct search methods were formally proposed and widely applied in the 1960s. They were implemented to solve difficult problems that arise from economics and engineering because they can easily be used for almost any optimization problem. Here, we should note that there exist more than one class of direct search methods in the literature: The earliest version of direct search methods was first proposed by Davidon [9], where the method uses the pattern which is in the shape of a set of coordinate directions. Polak [29] investigated an optimization text for the pattern search algorithm in [9] and recognized that pattern search methods contain sufficient structure to support a global convergence result. Moreover, Kolda, Lewis and Torczon [18], and Lewis, Torczon and Trosset [20] presented a review and made a summary on *general pattern search* (GPS) algorithms. The convergence analysis for this type of algorithms was first proposed by Torczon in [37, 38]. Popović and Teel [30] extended the GPS algorithms to nonsmooth optimization problem and demonstrated the theoretical guarantee on the convergence of the methods in a nonsmooth optimization framework. Besides, another type of direct search algorithms *mesh adaptive direct search* (MADS) methods, was recently investigated by Audet and Dennis [2] which overcomes the limitation of exploring through a finite number of directions. The authors established a convergence result for Clarke stationary points in the nonsmooth optimization problem. More recently, Abramson and Audet [1] proceeded to propose the second-order convergence analysis for both MADS methods and GPS methods.

The method considered in the paper is motivated by the analysis on pattern search methods, and the relevant investigation can be extended to the general direct search methods. The term *direct search method* hereafter should be understood in the following sense: For minimizing a function $U(x)$, at the $k-$th iteration of the algorithm, the next iterate $x_{k+1} \neq x_k$ is obtained if there exists a direction $d_k \in \mathcal{D}_k$ and a positive scalar δ_k such that $U(x_k + \delta_k d_k) < U(x_k)$, where $\mathcal{D}_k$ is the set of all feasible directions at iteration k of the direct search method, and x_{k+1} is defined as $x_k + \delta_k d_k$. For the use of the method, the definition and characterization of the set of directions is the key to the convergence of direct search methods.

In this study, we propose a variation of direct search method focusing on the hierarchical structure of the leader-follower optimization problem (2.3). The direct search method for bilevel programming problems was first proposed by Mersha and Dempe [24]. In [24], the direct search algorithm and the correspond convergence analysis are implemented to a single-level Lipschitz optimization problem by substituting the close form of the solution in the lower level to the objective function in the upper level, i.e. problem (2.4) in the paper. However, solving the single-level problem implicitly assumes that the close form of the follower's optimization problem (2.1) or an accurate solution $y(x)$ to (2.1) is known when the leader chooses her optimal decision. Note that, in most research, the availability of the close form is generally not assumed for the optimization problem for which the direct search method are applied. In fact, the availability of the close form essentially suggests that the gradient or derivative information can be used for solving the optimization problem, that is, derivative free methods can be replaced by derivative-based methods such as Newton methods, quasi Newton methods and etc, or by algebraically solving the Karush-Kuhn-Tucker (KKT) optimality conditions.

3.2 Bilevel direct search algorithms

In the paper, we propose a distrbuted bilevel direct search method for solving the leader-follower optimization problem. In accordance with the structure of the leader-follower problems, the method is hierarchically designed into two levels: An upper level algorithm is implemented to select a descent direction for the leader's problem at each trial point x_k of iteration k; on the other hand, a lower level algorithm is iteratively implemented at iteration k of the upper level algorithm to yield the lower's strategic reaction y_k when the leader's decision is x_k. Here, we

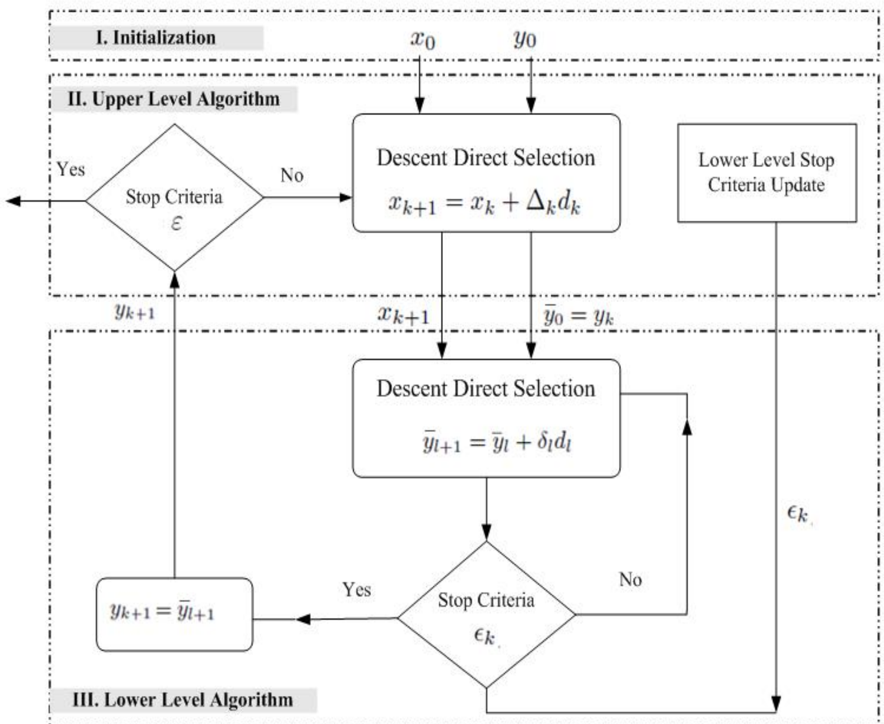

Figure 1: The bilevel structure of the algorithm

present the structure of this distributed bilevel direct search method in Figure 1. Moreover, the lower level and the upper level algorithms in Figure 1 are given in Algorithms 1 and 2 respectively.

In this section, we focus our analysis by applying a class of direct search methods, *general pattern search* (GPS). The concept of GPS is generic enough to capture many features of direct search algorithms, including classical pattern search, while still remaining simple enough to discuss with a minimum of notation and without handling a variety of special cases. In Algorithms 1 and 2, we use the GPS scheme to illustrate how the direct search methods can be implemented to solve the bilevel optimization methods. Note that, we will not review the details of a general GPS methods in this section. For the existing proof of completeness and a history of the problem in general we refer the readers to an excellent review paper by Kolda, Lewis and Torczon [18]. In the following content, we summary some fundamental concepts in the GPS methods [18], which are also used in our bilevel algorithms.

First, the GPS algorithm must be equipped with a set of search directions that includes a descent direction. To avoid poor search directions, there must be a descent direction that is not "too orthogonal" to the direction of steepest descent. To ensure that the above conditions are satisfied without using knowledge of explicit gradients, GPS methods use multiple search directions. The set of these directions is known as the *generating set*. In every iteration of the methods, we define $\mathcal{D}$ as a set of search direct positively spanning the space $\mathrm{I\!R}^d$ where $d = n, m$ in the study. For example, in the simplest case, $\mathcal{D}$ in an iteration in the algorithms consists of $2d$ coordinate directions, defined as the positive and negative unit coordinate vectors,

$$\mathcal{D} := \{e_1, e_2, \cdots, e_d, -e_1, -e_2, \cdots, -e_d\},$$

being a constant set at every iteration, where e_i' is a unit d dimensional vector with its i'th component being 1 for $i' = 1, 2, \cdots, d$. Generally, in every iteration of a direct search algorithm, a set of search direction $\mathcal{D}$ contains a generating set $\mathcal{G}$ for $\mathrm{I\!R}^d$ and additional directions $\mathcal{H}$. A set $\mathcal{G} = \{d^{(1)}, \cdots, d^{(r)}\}$ of $r > d+1$ vectors in $\mathrm{I\!R}^d$ generates $\mathrm{I\!R}^d$ if $\mathcal{G}$ positively spans $\mathrm{I\!R}^d$, that is, for any vector $v \in \mathrm{I\!R}^d$, there exist $\lambda^{(1)}, \cdots, \lambda^{(r)} \geq 0$ such that $v = \sum_{i=1}^{r} \lambda^{(i)} d^{(i)}$. A generating set must contain a minimum of $d+1$ vectors. For example, in a two dimensions, a minimal generating set with $d+1 = 3$ vectors is

$$\mathcal{G} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}.$$

One of the measures to quantify the worst-case distance between the steepest descent direction $\nu = -\nabla f(x)$ and the vector in $\mathcal{D}$ that makes the smallest angle with ν can be formulated is called *cosine measure* and defined as follows,

$$\kappa(\mathcal{D}) := \min_{v \in \mathrm{I\!R}^d} \max_{d \in \mathcal{D}} \frac{v^T d}{\|v\| \|d\|},$$

where $\mathcal{D}$ is a generating set in $\mathrm{I\!R}^d$. In order to prevent a slow rate of descent, in the algorithm, the cosine measure of generating sets in every iteration must be bounded below by a constant.

The nonnegative functions ρ and ψ are called the *forcing function*. Choosing ρ and ψ to be identically zero imposes a simple decreasing condition on the acceptance of the step. Under a set of regularity conditions (for example, when the forcing function ρ and ψ are positive, together with some other relatively mild conditions), the running of each algorithm can yield an accumulation point.

Algorithm 1: Direct search algorithm for the lower level programming.

input : initial point (x, y), error tolerance ϵ where $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $\epsilon > 0$.

output: the estimation $\bar{y}(x, y, \epsilon)$ to the follower's solution $y(x)$, the estimation $\bar{f}(x, y, \epsilon)$ follower's objective value $f(x, y(x))$ and the estimation $\bar{\mathcal{F}}(x, y, \epsilon)$ leader's optimal value function $\mathcal{F}(x)$.

Initialization:

Let $\bar{y}_0 = y$ be the initial guess.

Let $\delta_{\text{tol}} = \epsilon$ be the error tolerance used to terminate the lower level algorithm.

Let $\delta_0 > \delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\theta_M \in (0, 1)$ be an upper bound on the contraction parameter.

Let $\rho : [0, +\infty) \to \mathbb{R}$ be a continuous function such that $\rho(t)$ is decreasing as $t \to 0$ and $\rho(t)/t \to 0$ as $t \downarrow 0$. The choice $\rho \equiv 0$ is acceptable.

Let $\beta_M \geq \beta_m > 0$ be upper and lower bounds, respectively, on the lengths of the vectors in any generating set.

Let $\kappa_m > 0$ be a lower bound on the cosine measure of any generating set.

Algorithm: For each iteration $l = 0, 1, 2, \cdots$

Step 1. Let $\bar{\mathcal{D}}_l = \bar{\mathcal{G}}_l \bigcup \bar{\mathcal{H}}_l$. Here $\bar{\mathcal{G}}_l$ is a generating set from $\mathbb{R}^m$ satisfying $\beta_m \leq \|d\| \leq \beta_M$ for all $d \in \bar{\mathcal{G}}_l$ and its cosine measure $\kappa(\bar{\mathcal{D}}_l) \geq \kappa_m$, and $\bar{\mathcal{H}}_l$ is a finite (possibly empty) set of additional search directions such that $\beta_m \leq \|d\|$ for $d \in \bar{\mathcal{H}}_l$.

Step 2. **Successful Iteration:** If there exists $d_l \in \bar{\mathcal{D}}_l$ such that $\bar{y}_l + \delta_l d_l \in \bar{\mathcal{D}}_l$ and

$$f(x, \bar{y}_l + \delta_l d_l) < f(x, \bar{y}_l) - \rho(\delta_l),$$

then do the following:

- Set $\bar{y}_{l+1} = \bar{y}_l + \delta_l d_l$ (change the iterate).
- Set $\delta_{l+1} = \phi_l \delta_l$, where $\phi_l \geq 1$ (optionally expand the step-length parameter).

Step 3. **Unsuccessful Iteration:** Otherwise, if $\bar{y}_l + \delta_l d_l \notin \bar{\mathcal{D}}_l$ or

$$f(x, \bar{y}_l + \delta_l d) \geq f(x, \bar{y}_l) - \rho(\delta_l)$$

for all $d \in \bar{\mathcal{D}}_l$, do the following:

- Set $\bar{y}_{l+1} = \bar{y}_l$ (no change to the iterate).
- Set $\delta_{l+1} = \theta_l \delta_l$, where $0 < \theta_l < \theta_M < 1$ (contract the step-length parameter).

Step 4. If $\delta_{k+1} < \delta_{\text{tol}}$, then **terminate** (set the terminate iterate to be L). Otherwise, go to Step 1 and $l = l + 1$.

Return: $\bar{y}(x, \epsilon) = \bar{y}_L$, $\bar{f}(x, \epsilon) = f(x, \bar{y}_L)$ and $\bar{\mathcal{F}}(x, \epsilon) = F(x, \bar{y}_L)$ where all these values are depends on ϵ.

Algorithm 2: Direct search algorithm for the upper level programming.

input : initial point x^0, error tolerance ε.

output: the estimation x_K of the leader's position x^* in equilibrium and the estimation y_k of the follower's position y^* in equilibrium, and follower's objective $f(x^*, y^*)$ and leader's objective $F(x^*, y^*)$ at the equilibrium.

Initialization:

Let $x_0 = x^0 \in \mathcal{X}$ be the initial guess for the leader's problem.

Let $y_0 = y^0 \in \mathcal{Y}$ be the initial guess for the follower's decision.

Let $\Delta_{\text{tol}} = \varepsilon > 0$ be the terminate condition.

Let $\Delta_0 > \Delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\Theta_M \in (0, 1)$ be an upper bound on the contraction parameter.

Let $\gamma_M \in (0, 1)$ be an upper bound on the contraction of the lower level error tolerance.

Let $\psi : [0, +\infty) \to \mathbb{R}$ be a continuous function such that $\psi(t)$ is decreasing as $t \to 0$ and $\psi(t)/t \to 0$ as $t \downarrow 0$. The choice $\psi \equiv 0$ is acceptable.

Let $\tau_M \geq \tau_m > 0$ be upper and lower bounds, respectively, on the lengths of the vectors in any generating set.

Let $\varrho_m > 0$ be a lower bound on the cosine measure of any generating set.

Algorithm. For each iteration $k = 0, 1, 2, \cdots$

Step 1. Let $\mathcal{D}_k = \mathcal{G}_k \bigcup \mathcal{H}_k$. Here $\mathcal{G}_k$ is a generating set from $\mathbb{R}^n$ satisfying $\tau_m \leq \|d\| \leq \tau_M$ for all $d \in \mathcal{G}_k$ and its cosine measure $\kappa(\mathcal{D}_k) \geq \varrho_m$, and $\mathcal{H}_k$ is a finite (possibly empty) set of additional search directions such that $\tau_m \leq \|d\|$ for $d \in \mathcal{H}_k$.

Step 2. Set the lower level error tolerance as $\epsilon_k = \gamma_k \epsilon_{k-1}$ where $0 < \gamma_k < \gamma_M < 1$, and $y_k = \bar{y}(x_{k-1}, y_{k-1}, \epsilon_{k-1})$ which is obtained from the **output** in Algorithm 1.

Step 3. Successful Iteration: If there exists $d_k \in \mathcal{D}_k$ such that $x_k + \Delta_k d_k \in \mathcal{D}_k$ and

$$\bar{\mathcal{F}}(x_k + \Delta_k d_k, y_k, \epsilon_k) < \bar{\mathcal{F}}(x_k, y_k, \epsilon_{k-1}) - \psi(\Delta_k),$$

where $\bar{\mathcal{F}}(x, y, \epsilon)$ is calculated from Algorithm 1 with input x, y and ϵ. Then we do the following:

- Set $x_{k+1} = x_k + \Delta_k d_k$ (change the iterate).
- Set $\Delta_{k+1} = \Phi_k \Delta_k$, where $\Phi_k \geq 1$ (optionally expand the step-length parameter).

Step 4. Unsuccessful Iteration: Otherwise, if $x_k + \Delta_k d_k \notin \tilde{\mathcal{D}}_k$ or

$$\bar{\mathcal{F}}(x_k + \Delta_k d, y_k, \epsilon_k) \geq \bar{\mathcal{F}}(x_k, y_k, \epsilon_{k-1}) - \psi(\Delta_k)$$

for all $d \in \tilde{\mathcal{D}}_k$, do the following:

- Set $x_{k+1} = x_k$ (no change to the iterate).
- Set $\Delta_{k+1} = \Theta_k \Delta_k$, where $0 < \Theta_k < \Theta_M < 1$ (contract the step-length parameter).

Step 5. If $\Delta_{k+1} < \Delta_{\text{tol}} = \varepsilon$, then **terminate** (set the terminate iterate to be K). Otherwise, go to Step 1 and $k = k + 1$.

Return: $x^* = x_K$, $y^* = y_K$, $f(x_K, y_K)$ (Algorithm 1) and $F(x_K, y_K)$, where all these values are depends on x^0, y^0 and ε.

To our knowledge, differing from the derivative free methods used for the bilevel optimization problem so far [24], the distributed bilevel algorithm has the following features: (a) One of the most important features is that the analytical forms of lower objective functions and the gradient information are not required in the computation. (b) The bilevel scheme of the algorithms reflects the empirical decision making process in bilevel optimization problems. At every iteration, when the leader's decision has been known, the follower always reacts in an optimal way by implementing an optimal search algorithm to his decision making process. (c) The methods can be implemented in a real world game situation where the game is incomplete, i.e., the decision makers do not have to know each other's cost functionals and parameters, and only have to communicate to each other their tentative decisions during the end of each iteration. The structure described in Figure 1 enables parallel and distributed computation of equilibria or stationary points, that is, the upper and the lower algorithms can be implemented in different computers.

4 Convergence Analysis

In this section, we present some convergence results on the bilevel direct search method consisting of the lower level algorithm (Algorithm 1) and the upper level algorithm (Algorithm 2). The convergence analysis is related to the results on the direct search methods for the optimization problems with Lipschitz continuous objective functions. We encourage interested readers to refer [18, 24, 30].

4.1 Lower level algorithm

First, we investigate Algorithm 1 for the lower level programming problem for a fixed $x = x_k$, $u = y_k$ and $\epsilon = \epsilon_k$ (see the **input** in Algorithm 1), which implies that the upper level algorithm is at the $k-$th step. Consider the $l-$th iterate y_l in the lower level algorithm. The next iterate $y_{l+1} \neq y_l$ is produced if there exists a scalar $\delta_l > 0$ and a direction $d_l \in \bar{\mathcal{D}}_l$ such that $y_l + \delta_l d_l \in \mathcal{Y}$ and $f(x, y_l + \delta_l d_l) < f(x, y_l)$. In this case, we have $y_{l+1} = y_l + \delta_l d_l$. To know how the algorithm works, we need to look at the definition of the generating set $\bar{\mathcal{D}}_l$.

The definition and characterization of the aforementioned set of direction, $\bar{\mathcal{D}}_l$, is the key to the convergence of the direct search method. The existence of such a finite set of search directions at a nonstationary point within a certain compact set will be shown in the following proposition. First, recall that the objective function f is sufficiently smooth near $y \in \mathcal{Y}$ and $x \in \mathcal{X}$. A vector d is said to be a *descent direction* of $f(x, \cdot)$ at y if $f'_y(x, y; d) < 0$. A set $\bar{\mathcal{D}} \subset \mathrm{I\!R}^m$ is a *descent set* of $f(x, \cdot)$ in some set X if, for every $x \in X$, there exists $d \in \bar{\mathcal{D}}$ such that $f'_y(x, y; d) < 0$. Then, the next result shows the existence of a finite set of descent directions at a nonstationary point.

Proposition 4.1 *Let the lower level problem satisfy Assumption 2.1. Let $f(x,\cdot)$ be a sufficiently smooth function of $y \in \mathcal{Y}^0$ for any fixed $x \in \mathcal{X}$, where $\mathcal{Y}^0 \subset \mathcal{Y}$ is a compact set that does not contain any local minimum of problem (2.1). Then, for any fixed x and $y \in \mathcal{Y}^0$, there exist a finite set $\bar{\mathcal{D}}$ of vectors and a positive number α such that*

$$\min_{d\in\bar{\mathcal{D}}} \quad f'_y(x,y;d) \leq -\alpha. \tag{4.1}$$

Proof. Suppose that there exist $x \in \mathcal{X}$ and $y \in \mathcal{Y}^0$ such that, for every direction $d \in \mathbb{R}^m$, we have $f'_y(x,y;d) \geq 0$. This implies that $\nabla_y f(x,y) = 0$. Therefore, y^* is a stationary point for problem (2.1). Hence, under the strict convexity of function $f(x,\cdot)$, we have that y^* is a local minimum of problem (2.1), which is a contradiction by the assumptions of the proposition. ■

Note that, differing from the classic direct search algorithms, the output of Algorithm 1 is deemed as an approximation of the lower level optimal solution to (2.1) used in Algorithm 2, where Algorithm 1 is performed for the trial points at each iteration of Algorithm 2. This hierarchical structure implies that we can not drive the number of iterations in Algorithm 1 to infinity if the number of iterations in Algorithm 2 is finite. (Otherwise, the computational time for each iteration in Algorithm 2 will go to infinity.) Therefore, in Algorithm 1, for each step, $\epsilon = \epsilon_k$ is used to terminate the algorithm within a finite iterations where $k = 1, 2, \cdots$ and $k < +\infty$. On the other hand, ϵ also introduces some approximation errors at every iteration in Algorithm 2. In this following result, we are concerned with the deviation of the approximate solution yield by Algorithm 1, i.e. the output $y(x, y_k, \epsilon_k)$ provided by Algorithm 1, to the true optimal solution $y(x)$ to (2.1) when the trial point is set at x in the $k-$th iteration of Algorithm 2.

Proposition 4.2 *Let f satisfy Assumption 2.1. Suppose that for any fixed x, $y^*(x)$ is a minimizer of problem (2.1) and $\nabla^2_y f(x,y^*)$ is positive definite for any fixed x. For Algorithm 1, assume that*

(i) *$\phi_k = 1$ for all k in successful iterations;*

(ii) *$\rho(t) = \alpha t^p$ for some fixed $\alpha > 0$ and fixed $p \geq 2$;*

(iii) *$\beta_m \leq \|d\| \leq \beta_M$ for all $d \in \bar{\mathcal{D}}_l$ and all l.*

Then, we have

(a) *given the input of Algorithm 1 are x and ϵ, for unsuccessful iteration l, there exists a constant $c(x)$ independent of l, such that $\|\bar{y}_l - y^*(x)\| \leq c(x)\delta_l$;*

(b) *the output of Algorithm 1 satisfies $\|\bar{y}_L - y^*(x)\| \leq c(x)\epsilon_k$, that is, $\|y(x, y_k, \epsilon_k) - y^*(x)\| \leq c(x_k)\epsilon_k$ when the upper level algorithm is at iteration k.*

Proof. (a) The conclusion can be easily derived from Theorem 3.15 in [18].

(b) From Algorithm 1, we can see that the algorithm is terminated at some unsuccessful iteration where the step-length is contracted. Therefore, by (a), we have $\|\bar{y}_l - y^*(x)\| \le c(x)\delta_l$. Moreover, because of the terminate condition $\delta_{l+1} < \delta_{\text{tol}} = \epsilon_k$, we have $\|\bar{y}_l - y^*(x)\| \le c(x)\epsilon_k$ when the iteration in Algorithm is at k. ■

4.2 Upper level algorithm

Let (x', y') be an arbitrary point and the set

$$X(x', y') := \left\{ x \mid F(x, y) \le F(x', y'),\ y \in \Psi(x) \right\}$$

be compact and nonempty. Note that, $X(x', y')$ can be seen as a generalized definition of level function of $F(x, y)$. In the study, we concentrate our investigation on the cases where the stationary points are within a bounded set $\hat{X}$ and the set of arbitrary points (x', y') such that $X(x', y') \subset \hat{X}$.

Let $\mathcal{C}$ be the set of Clarke stationary points of (2.3). Let σ be a sufficiently small positive number. Set

$$S_{\mathcal{C}} := \hat{X} \backslash \bigcup_{x \in \mathcal{C}} B(x, \sigma),$$

where $B(x, \sigma)$ denotes the closed ball centered at x with radius $\sigma > 0$.

For the investigation of the upper level algorithm, we need to consider the leader's optimal value function $\mathcal{F}(x)$. Due to the smoothness of F and the Lipschitz continuity of $y(\cdot)$, the generalized directional derivative $\mathcal{F}^o(x; d)$ is well defined. Moreover, we have the closedness of set $S_{\mathcal{C}}$.

Proposition 4.3 (Theorem 4.1 in [24]) *Under Assumption 2.1, for every $x \in S_{\mathcal{C}}$, there exist a finite set $\mathcal{D}$ of vectors and a positive number $\alpha > 0$ such that*

$$\min_{d \in \mathcal{D}} \quad \mathcal{F}^o(x; d) \le -\alpha. \tag{4.2}$$

This proposition essentially suggests a similar result for the lower level programming problem as Proposition 4.1. Next, we define the *descent cone* and the *aperture* of a convex cone. See [24] for a reference.

Definition 4.4

(a) *A* descent cone *of $F(\cdot, y(\cdot)) = \mathcal{F}(\cdot)$ at a point $x \in \mathcal{X}$ is defined as*

$$C(x) := \{ v \in \mathbb{R}^n \mid \mathcal{F}^o(x; v) < 0 \}.$$

(b) *Let C be a convex cone that does not contain zero vector, and $\overline{C}$ be the closure of C. The* aperture *of the cone C, denoted by $\varphi(C)$, is defined as*

$$\varphi(C) := \arccos\left(\min_{w\neq 0, w\in\overline{C}} \sup_{z\in\mathbb{R}^n/\overline{C}, z\neq 0} \frac{w^T z}{\|w\|\|z\|}\right).$$

(c) *The* descent aperture *of $\mathcal{F}(\cdot)$ of some set S is defined as the smallest aperture of all descent cones of $\mathcal{F}(\cdot)$ on the set, that is,*

$$\hat{\varphi}(\mathcal{F}, S) := \inf_{x\in S} \varphi(C(x)),$$

where $C(x)$ is a convex cone contained in a descent cone $\{d \,|\, \mathcal{F}^o(x; d) < 0\}$.

Now we state one result that helps us to prove our main result.

Lemma 4.5 (Lemma 4.8 in [24]) *Let Assumption 2.1 be satisfied. Let $\mathcal{D}$ be any finite generating set of vectors such that the vector density $\kappa(\mathcal{D}) > \cos(\hat{\varphi}(\mathcal{F}, S_{\mathcal{C}}))$ and $\|d\| \in [\tau_m, \tau_M]$ for all $d \in \mathcal{D}$. Then, for sufficiently small $\Delta > 0$ and t^*, there exists $d \in \mathcal{D}$ such that*

$$\mathcal{F}(x + td) - \mathcal{F}(x) \leq -\Delta t$$

for all $t \in (0, t^*)$ and for all $x \in S_{\mathcal{C}}$.

Lemma 4.6 (Convergence of the Step-Length) *Let $\mathcal{X}(x_0) := \{x |\ \mathcal{F}(x) \leq \mathcal{F}(x_0)\}$ be compact and $\{x_k\}$ be a sequence of iterates produced by Algorithm 2. Then, under Assumption 2.1,* $\lim_{k\to\infty} \Delta_k = 0$.

Proof. By contradiction, suppose that there exists a subsequence $\{\Delta_{k_i}\}_{i=0}^{\infty}$ with $\lim_{i\to\infty} \Delta_{k_i} = \zeta > 0$. Then, either $\lim_{k\to\infty} \Delta_k = \zeta$ or the limit does not exist. Since $\Delta_{k_i} > 0$, then step k_i is a successful iteration. Then we have the following descent result according to successful steps in Algorithm 2:

$$F\left(x_{k_i+1}, y_{k_i}, \epsilon_{k_i+1}\right) - F\left(x_{k_i}, y_{k_i}, \epsilon_{k_i}\right) \leq -\psi\left(\Delta_{k_i}\right), \quad i = 1, 2, \cdots \tag{4.3}$$

with $\psi(\Delta_{k_i}) > 0$, where $x_{k_i+1} := x_{k_i} + \Delta_{k_i} d_{k_i}$. Moreover, we denote the set of the index of successful iteration by $\mathcal{S}$ where $\{k_i\}$ is a subset of $\mathcal{S}$. Then, for any iteration $k' \in \mathcal{S}$ we have

$$\bar{\mathcal{F}}\left(x_{k'+1}, y_{k'}, \epsilon_{k'+1}\right) - \bar{\mathcal{F}}\left(x_{k'}, y_{k'}, \epsilon_{k'}\right) \leq -\psi\left(\Delta_{k'}\right) < 0, \tag{4.4}$$

where $x_{k'+1} := x_{k'} + \Delta_{k'} d_{k'}$. On the other hand, we have $x_{k''+1} = x_{k''}$ for any iteration $k'' \notin \mathcal{S}$ and hence

$$\begin{aligned}
& \bar{\mathcal{F}}\left(x_{k''+1}, y_{k''}, \epsilon_{k''+1}\right) - \bar{\mathcal{F}}\left(x_{k''}, y_{k''}, \epsilon_{k''}\right) \\
= \;& \bar{\mathcal{F}}\left(x_{k''}, y_{k''}, \epsilon_{k''+1}\right) - \bar{\mathcal{F}}\left(x_{k''}, y_{k''}, \epsilon_{k''}\right) \\
= \;& F\left(x_{k''}, \bar{y}\left(x_{k''}, y_{k''}, \epsilon_{k''+1}\right)\right) - F\left(x_{k''}, \bar{y}\left(x_{k''}, y_{k''}, \epsilon_{k''}\right)\right) \\
\leq \;& M_F \left\|\bar{y}\left(x_{k''}, y_{k''}, \epsilon_{k''+1}\right) - \bar{y}\left(x_{k''}, y_{k''}, \epsilon_{k''}\right)\right\| \\
\leq \;& M_F c(x_{k''}) \left(\epsilon_{k''} + \epsilon_{k''+1}\right),
\end{aligned} \tag{4.5}$$

where the first inequality is from the Lipschitz continuity of F and the second inequality is from Proposition 4.2. Then, by summing up the inequalities (4.3), (4.4) and (4.5), at iteration k, we have that

$$\begin{aligned}
& \bar{\mathcal{F}}\left(x_k, y_k, \epsilon_k\right) - \bar{\mathcal{F}}\left(x_0, y_0, \epsilon_0\right) \\
\leq\ & \sum_{k' \in \{k_i\}, k' < k} -\psi\left(\Delta_{k_i}\right) + \sum_{k' \in \mathcal{S} \setminus \{k_i\}, k' < k} -\psi\left(\Delta_{k'}\right) + \sum_{k'' \notin \mathcal{S}, k'' < k} M_F c(x_{k''})\left(\epsilon_{k''} + \epsilon_{k''+1}\right) \\
\leq\ & \sum_{k' \in \{k_i\}, k' < k} -\psi\left(\Delta_{k_i}\right) + \sum_{k'' \notin \{k_i\}, k'' < k} M_F c\left(\epsilon_{k''} + \epsilon_{k''+1}\right) \\
\leq\ & \sum_{k' \in \{k_i\}, k' < k} -\psi\left(\Delta_{k_i}\right) + \sum_{k''=1}^{\infty} M_F c\left(\epsilon_{k''} + \epsilon_{k''+1}\right) \\
\leq\ & \sum_{k' \in \{k_i\}, k' < k} -\psi\left(\Delta_{k_i}\right) + M_F c \frac{\epsilon_1 + \epsilon_2}{1 - \gamma_k}
\end{aligned}$$

where the term $M_F c(\epsilon_1 + \epsilon_2)/(1 - \gamma_k)$ is upper bounded and c is an upper bound of $c(x)$ for all $x \in \mathcal{X}(x_0)$. Then, since $\psi(\cdot) > 0$ for $\Delta > 0$ and $\lim_{i \to +\infty} \Delta_{k_i} = \zeta > 0$, we can easily verify that

$$\lim_{i \to \infty} \bar{\mathcal{F}}\left(x_{k_i}, y_{k_i}, \epsilon_{k_i}\right) \leq \bar{\mathcal{F}}\left(x_0, y_0, \epsilon_0\right) + \sum_{k' \in \{k_i\}, k' < k} -\psi\left(\Delta_{k_i}\right) + M_F c \frac{\epsilon_1 + \epsilon_2}{1 - \gamma_k} = -\infty.$$

On the other hand, we look at the deviation of $\bar{y}(x, y, \epsilon)$ calculated from Algorithm 1 for the lower level problem to the true solution. From Proposition 4.2, we have

$$\|\bar{y}(x, y, \epsilon) - y^*(x)\| \leq c\epsilon.$$

From the Lipschitz continuity and the definition of $\bar{\mathcal{F}}\left(x_{k_i}, y_{k_i}, \epsilon_{k_i}\right)$ in Algorithm 1, we have

$$\begin{aligned}
& \left|\bar{\mathcal{F}}\left(x_{k_i}, y_{k_i}, \epsilon_{k_i}\right) - F\left(x_{k_i}, y^*(x_{k_i})\right)\right| \\
=\ & \left|\bar{\mathcal{F}}\left(x_{k_i}, y\left(x_{k_i}, y_{k_i}, \epsilon_{k_i}\right)\right) - \bar{\mathcal{F}}\left(x_{k_i}, y^*(x_{k_i}, 0)\right)\right| \\
\leq\ & M_F c(x_{k_i}) \epsilon_{k_i},
\end{aligned}$$

where the true function value $F\left(x_{k_i}, y^*(x_{k_i})\right)$ can be seen as the output of Algorithm 1 with terminate criteria parameter $\epsilon = 0$. Consequently, we have

$$F(x_{k_i}, y^*(x_{k_i})) - F\left(x_{k_i}, y^0, \epsilon_{k_i}\right) \leq M_F c \epsilon_{k_i},$$

which implies that

$$\lim_{i \to \infty} \mathcal{F}(x_{k_i}) = \lim_{i \to \infty} F(x_{k_i}, y^*(x_{k_i})) = -\infty.$$

This contradicts the boundedness from below of $\mathcal{F}$. Hence, this infinite subsequence can not exist and hence $\lim_{k \to \infty} \Delta_k = 0$. ∎

Theorem 4.7 *Let $\mathcal{X}(x_0) := \{x \mid \mathcal{F}(x) \leq \mathcal{F}(x_0)\}$ be compact and Assumption 2.1 be satisfied. Let $\{x_k\}$ be a sequence of iterates produced by Algorithm 2. Then every limit point of $\{x_k\}$ is a stationary point of $\mathcal{F}$.*

Proof. Denote by $\mathcal{C}$ the set of stationary points of $\mathcal{F}$. By Lemma 4.6, we have $\lim_{k\to\infty} \Delta_k = 0$. This implies that there are infinitely many unsuccessful iterates for Algorithm 2. Here, we use δ to denote an arbitrary small positive number and define

$$S_{\mathcal{C}}(\delta) := \hat{X} \backslash \bigcup_{x\in\mathcal{C}} B(x,\delta).$$

Here we use $S_{\mathcal{C}}(\delta)$ instead of δ to emphasis that the definition of $S_{\mathcal{C}}$ varies for different δ. Since $x_k \in \hat{X}$, we have either $x_k \in S_{\mathcal{C}}(\delta)$ or $x_k \in \bigcup_{x\in\mathcal{C}} B(x,\delta)$. Suppose that $x_k \in S_{\mathcal{C}}(\delta)$. Since $\kappa(\mathcal{D}) > \cos(\hat{\varphi}(\mathcal{F}, S_{\mathcal{C}}(\delta)))$, we obtain from Lemma 4.5 that, for such x_k and for all $t^* > 0$ and σ, there exists $d \in \mathcal{D}$ such that, for every $\Delta \in (0, t^*]$,

$$\bar{\mathcal{F}}(x_k + \Delta d, y_k, \epsilon_{k+1}) - \bar{\mathcal{F}}(x_k, y_k, \epsilon_k) < -\sigma\Delta.$$

for any arbitrary iterate y_k, ϵ_{k_i+1} and ϵ_{k_i}.

On the other hand, from Algorithm 1 for the lower level programming problem, we have

$$\|\bar{y}(x_k, y_k, \epsilon_k) - y^*(x_k)\| \leq c\epsilon_k$$

and hence

$$\left|\bar{\mathcal{F}}(x_k, y_k, \epsilon_k) - F(x_k, y^*(x_k))\right| \leq M_F c\epsilon_k,$$

where M_F is the Lipschitz modulus of the function F. Therefore, the number of x_k satisfying $\bar{\mathcal{F}}(x_k + \Delta d, y_k, \epsilon_{k+1}) - \bar{\mathcal{F}}(x_k, y_k, \epsilon_k) < -\sigma\Delta$ is finite. Otherwise, we have $\bar{\mathcal{F}}(x_k, y_k, \epsilon_k) \to -\infty$, which implies that $\mathcal{F}(x_k) \leq \bar{\mathcal{F}}(x_k, y_k, \epsilon_k) + M_F c\epsilon_k$ goes to $-\infty$ and contradicts to the assumption that $\mathcal{F}$ is bounded below.

Thus, there must exist k_0 such that

$$x_k \in \bigcup_{x\in\mathcal{C}} B(x,\delta), \quad k \geq k_0.$$

Since $\delta > 0$ is arbitrary chosen, this implies that all accumulation points of $\{x_k\}$ belong to $\mathcal{C}$. ■

Note that, by the boundedness of X, the sequence $\{x_k\}$ has one accumulation point at least.

5 Numerical Applications and Experiments

In this section, we present a leader-follower model for a hospital competition problem under regulated price, which is first explored by Eggleston and Yip [13]. By doing this, we show the use of the method in practice and its applications in the computation of health services systems.

Moreover, we present a numerical experiment for a single-leader-single-follower problem to verify the convergence of the method, and extend our method to solve a single-leader-multi-follower problem, where the direct search algorithm for the lower level problem (Algorithm 1) is replaced by a variation of trust region method by Yuan [43] for the competition for the multiple followers.

5.1 Hospital competition under regulated prices

In recent decades, several developing countries are making their efforts to provide social protection for their urban residents. In China, borrowing from the Singaporean model of individual Medical Savings Accounts (MSAs) combined with a social risk pooling fund for catastrophic expenditures, the urban labor insurance schemes (LIS) and the government employee insurance scheme (GIS) are currently covering approximately 50% of the urban population. Within these schemes, inpatient care is financed after the employee pays a deductible equal to 10% of his or her annual wage. In these developing countries, released from a government-owned centralized system, the delivery of health services heavily relies on the hospitals which may increasingly rely on the fee-for-services (FFS) and the profits from sales of pharmaceutical to cover their operation costs. One of the consequences is that regulated fees for some high-quality diagnostics are set far more above average cost. For example, the State Price Bureau in China allows hospital pharmacies to charge 15 percent markup on the wholesale price of drugs, which gives hospitals incentive to encourage overuse of profitable services rather than basic services. To illustrate how this system affects hospital revenues and public health outcome, we present a leader-follower model to characterize physicians' and hospital administrators' strategic behaviors under these health insurance schemes, where the model is first proposed by Eggleston and Yip [13] particularly for the urban health sector reforms in China.

In this system, there are two kinds of participators: physicians and hospital administrators. First, let us consider the strategic behaviors at the physician side. To well characterize patients' behavior, we index the patients by i and classify them into two sets where $i = H$ are the high-income, insured patients, and $i = L$ are the low-income, uninsured patients. Correspondingly, in a hospital, a physician provides different levels of health services indexed by s, where $s = 1$ represents profitable services with high-quality and high-technology services and drugs, and $s = 2$ represents basic services priced below marginal cost.

In the model, we use m_s^i to represent the spending on health service s delivered to patient i and use function $f_s^i(m_s^i)$ to characterize the increasing and concave benefit that patient i derives from receiving health care using the resources m_s^i. By summing up the services, the total service-related utility from the treatment is $F^i(m^i) = \sum_{s=1}^{2} f_s^i(m_s^i)$ on the one hand. On the other hand, to obtain the treatment, patient have to pay a coinsurance rated at C_0^i with $C_0^i \in [0, 1]$. If the patient is in the insured set, then we averagely have $C_0^H = 0.35$; otherwise, for the uninsured, by definition $C_0^L = 1$. Thus, the patient utility can be written as

$$F^i = \sum_{s=1}^{2} \left[f_s^i \left(m_s^i \right) - C_0^i P_s m_s^i \right], \tag{5.1}$$

where P_s is a distorted regulated price on different type of services for $s = 1$ and 2. Particularly, in the model, following [13] we investigate the case where the patient (consumer) utility is of linear marginal benefit and takes the following form,

$$F^i = \sum_{s=1}^{2} \left[\left(a_s m_s^i - \frac{b_s}{2}(m_s^i)^2 \right) - C_0^i P_s m_s^i \right], \tag{5.2}$$

where a_s and b_s are positive. Now, let us look at the physician behavior. Physicians are primary decision makers for health care, who decide which patients received how much of which services. Ideally, a physician is hired by patients as a specialized agent to perform a specific service-diagnosis and treatment for a medical condition. However, the evidence in practice suggests that physicians may also be influenced by the financial incentives, i.e. physicians have to respond to their employers or the other payers such as social health insurance. Unlike in the developed countries, one can consider physician in developing countries as an agent both for the patient and for the hospital.

In the model, similarly as its counterpart in [13], we focus on a representative physician with a linear compensation contract, where the income of a physician consists of two parts: a fixed payment per patient, R, plus reimbursement $(1 - w_s)m_s^i$ for each service s with $w_s \leq 1$. Given this linear compensation contract, the physician's net income from treatment patient i can be formulated as $\pi^i = R + m_1^i + m_2^i - w_1 m_1^i - w_2 m_2^i$ $(i = H, L)$.

The utility U of a representative physician features a constant marginal rate of substitution $\beta \geq 0$ between patient benefit and physician income, that is, a representative physician puts weight β on net revenue and weight 1 on the patient benefit, when deciding on how to treat patients. The physician's utility function is

$$U = \sum_{i=L,H} D_i F^i + \beta \sum_{i=L,H} D_i \pi^i = \sum_{i=L,H} D_i [F^i + \beta \pi^i], \tag{5.3}$$

where D_i is inelastic demand from patients $i = L, H$. From (5.3), we can see that the higher is β, the more finance incentives influence clinical decision. Given these incentive scheme, a representative physician in the two-service and two-patient type model choose a way to allocate the spending, i.e. the vector $m = (m_1^H, m_2^H, m_1^L, m_2^L)$, to maximize

$$\begin{aligned} U(m) \quad = \quad & D_H \left[\sum_{s=1,2} \left(a_s m_s^H - \frac{b_s}{2}(m_s^H)^2 - C_0^H P_s m_s^H \right) \right] \\ & + D_L \left[\sum_{s=1,2} \left(a_s m_s^L - \frac{b_s}{2}(m_s^L)^2 - C_0^L P_s m_s^L \right) \right] \\ & + D_H \left(R + m_1^H + m_2^H - w_1 m_1^H - w_2 m_2^H \right) \\ & + D_L \left(R + m_1^L + m_2^L - w_1 m_1^L - w_2 m_2^L \right). \end{aligned} \tag{5.4}$$

Although physicians play the primary role in deciding which patients receive how much of which services, hospital administrators also shape clinical decisions in many ways. In the problem, at the hospital level, we focus on the behavior of an administrator who seeks to maximize net revenue, where he or she will design an optimal scheme of physicians' compensation for profit-maximizing. To do so, the administrator have to rely on long-run experience to decide the reimbursement rate w_1 and w_2. Given the linear payment system, the hospital's revenue per patient treated is $r^i + \sum_{s=1,2}(1-w_s)m_s^i(w)$ where $w := (w_1, w_2)'$ is the decision variable of hospital administrator and r^i is the constant part of the revenue for treating one patient. Cost is total spending on patient care, $\sum_{s=1}^2 m_s^i(w)$ per patient i, and total physician compensation W which is fixed by physicians' reservation utility. Thus the hospital administrator's net revenue optimization problem can be formulated as

$$\begin{array}{ll} \max\limits_{w} & \Pi(w) := \sum\limits_{i=L,H} D_i \left[r^i + \sum\limits_{s=1}^{2}(1-w_s)\hat{m}_s^i(w) - \sum\limits_{s=1}^{2}\hat{m}_s^i(w) \right] - W \\ \text{s.t.} & \left(\hat{m}_1^L, \hat{m}_1^H, \hat{m}_2^L, \hat{m}_2^M\right)' \text{ solves} \max\limits_{m} U(m). \end{array}$$

where $U(m)$ is defined through (5.4). We can reformulate the net revenue function $\Pi(w)$ as $\sum_{i=L,H} D_i \left[r^i - \sum_{s=1}^2 w_s m_k^i(w) \right] - W$.

Note that, in [13], the Karush-Kuhn-Tucker conditions are included to solve the physician and administrator's optimization problem, which requires the knowledge of the gradients $\partial U/\partial m_k^i$ for $i = L, H$ and $k = 1, 2$. In the study, we applied the distributed bilevel direct search algorithm to solve (5.5) and compared the results with its counterpart in [13]. The most difference from the results in [13] is that the algorithm here extended the model to a case where the representative physician only can figure out an optimal response to fixed reimbursement rates set by the administrator, instead of knowing the optimal function in the lower level which is very hard for a physician to figure out it. In the numerical test, the parameters in problem (5.5) are set at the same values as in [13]. See Tables 2 and 3 in the appendix of [13]. The results are presented in Table 1. In [13], the physician's decisions at the lower level are estimated for the cases with the administrator's decision (w_1, w_2) at $(0.10, 0.50)$ and $(-0.27, 0.23)$ respectively. Here, we compare the first and second columns in Table 1 with their counterparts in [13].

Table 1: Summary of results yielded by the bilevel method

The value of (w_1, w_2)	(0.10, 0.50)	(-0.27, 0.23)
Private market share (D^H, D^L)	(0.62, 0.41)	(0.62, 0.41)
Distorted regulated prices (P_1, P_2)	(1.2, 0.8)	(1.2, 0.8)
Average spending, insured patient	3751.80	4450.20
Average spending, insured patient in [13]	3845.72	4379.10
Average spending, uninsured patient	1950.00	2479.30
Average spending, uninsured patient in [13]	1993.19	2455.36
Physician's utility	6068.50	7200.50
Administrator's net revenue	2878.10	3873.30

5.2 Numerical experiments for oligopolistic market models

5.2.1 Case I: Single follower problem

Consider an oligopolistic market with two firms which supply a homogeneous product in a noncooperative manner.

One of the firms is the dominant supplier in the market, hereafter named the leader, and has to decide now what her future supply x will be. We can imagine that this firm has not yet installed production capacities, and chooses her future supply level by taking into account the reaction of the other producer. The other firm, hereafter named the followers, then reacts after knowing the leader's supply level. In the model, we assume that the follower chooses his supply level maximizing his profit.

At the *demand side*, the effect, of the total supply level from both firms on the market price, is characterized by an (inverse) demand function $p(x+y)$. The inverse demand function $p(Q)$ gives the price at which consumers will demand a quantity $Q = x+y$. In the model, we consider a linear demand function, and set

$$p(Q) = A - BQ, \tag{5.5}$$

where the intersection $A = 5$ and the slope $B = 1$. Moreover, the cost functions of both firms are assumed to be linear as

$$\begin{array}{ll} \text{Leader's cost:} & c_l(x) = c_l x, \\ \text{Follower's cost:} & c_f(y) = c_f y, \end{array}$$

where $c_l = 3$ and $c_f = 2$. Therefore, the leader's decision problem can be written as the following bilevel programming problem:

$$\left\{ \begin{array}{ll} \max & p(x+y)x - c_l(x) \\ \text{s.t.} & x \in \mathcal{X}, \\ & y(x) \in \left\{ y \in \mathcal{Y} \middle| \ y \text{ solves } \max p(x+y)y - c_f(y) \right\}. \end{array} \right. \tag{5.6}$$

To perform the bilevel method, we set the initial point at $(x, y) = (2, 2)$ which is not an optimal solution. In the example, we also set the initial error tolerant as $\epsilon = 0.01$, and the initial guess of the leader and follower as $x^0 = 2$ and $y^0 = 2$. In Figures 2 and 3, we present the numerical results on the leader's and follower's decisions and profits at each iteration, where the true optimal solution of the problem is $x = 1/2, y = 5/4$ and the leader's optimal profit is 0.1250 and the follower's optimal profit is 1.5625.

Figure 2 shows the convergence of the leader's solution x_k and profit $F(x_k, y_k)$, in which the values of x_k and $F(x_k, y_k)$ fall into a neighborhood close to their true counterparts within 120 iterations in the upper level algorithm. In the same way, Figure 3 shows the convergence of the follower's solution y_k and profit $f(x_k, y_k)$, where the values of y_k and $f(x_k, y_k)$ fall into a neighborhood close to their true counterparts within 120 iterations in the upper level algorithm.

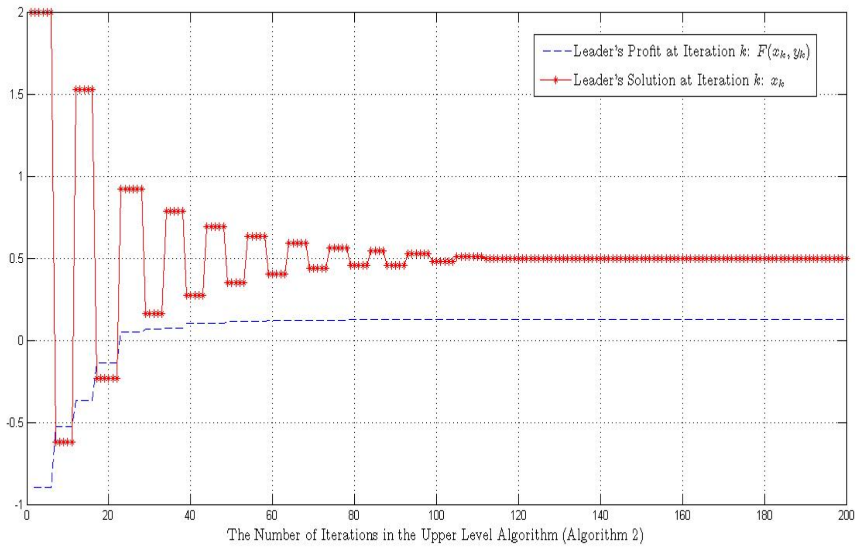

Figure 2: The leader's solution and profit at every iteration

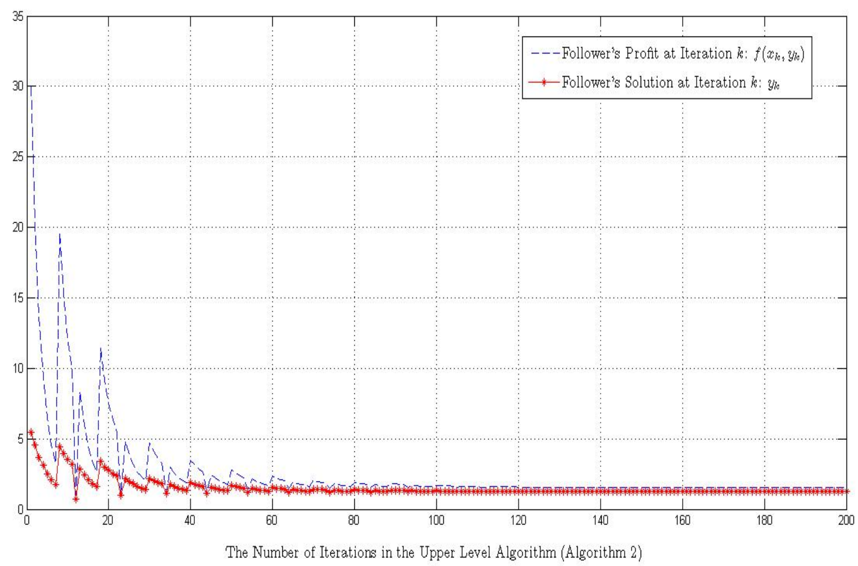

Figure 3: The follower's solution and profit at every iteration

From Algorithms 1 and 2, we can see that, in this bilevel method, the upper level and the lower level algorithms are linked by a parameter γ_k which controls the terminate criteria of the lower level algorithm by the condition $\epsilon_{k+1} = \gamma_k \epsilon_k$ in the upper level algorithm. In Figures 4 and 5, we present the results on how the different values of γ_k impacts the convergence of the algorithm.

Note that, the choice of the parameter γ_k in our algorithm currently is heuristic, where for

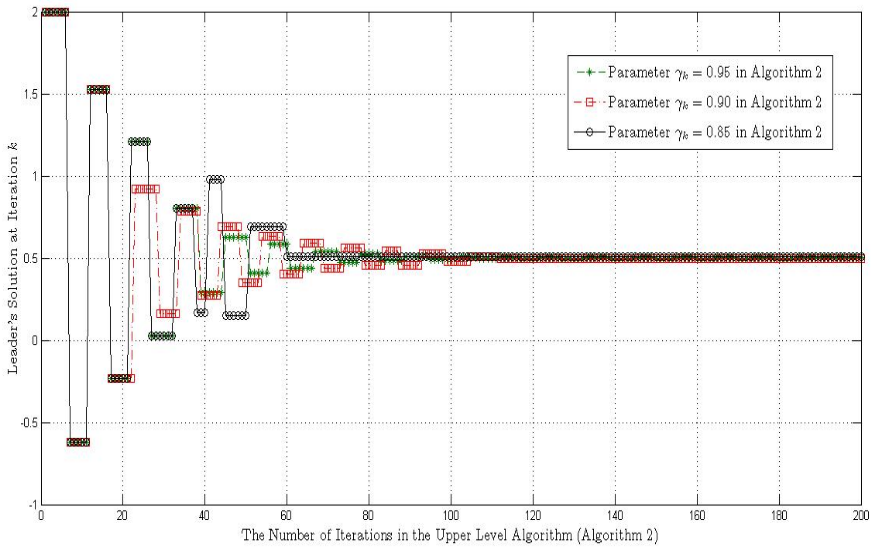

Figure 4: The leader's solution and profit at every iteration of the bilevel method

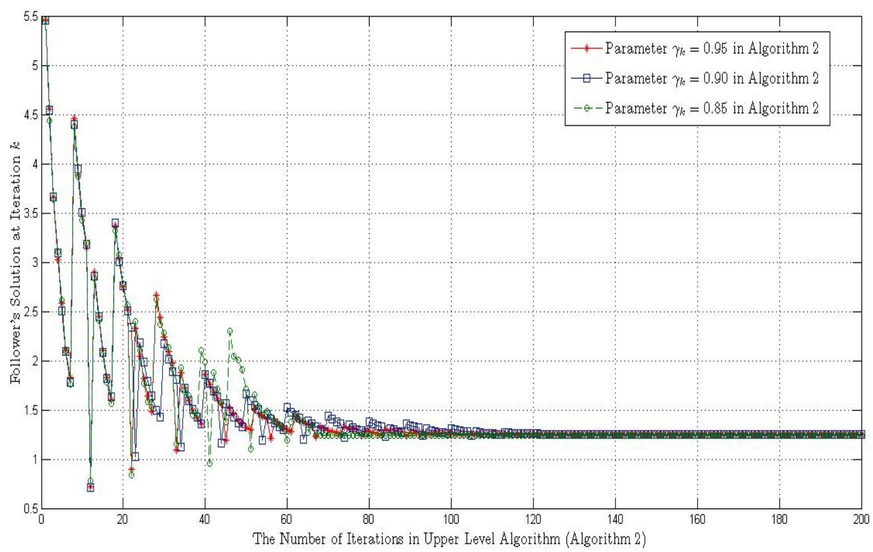

Figure 5: The follower's solution and profit at every iteration of the bilevel method

the same ϵ_k, a smaller γ_k increase the number of iterations at the lower level algorithm to satisfy a smaller error tolerance ϵ_{k+1} at the one side; At the other side, a larger γ_k might make the convergence of the upper level algorithm slower due to a solution y_{k+1} with larger deviation returned from the lower level algorithm. In our test, we perform the comparative analysis on the impact from the parameter γ_k to the convergence of the leader's and the follower's solutions. From Figure 4 with γ_k ranging from 0.95 to 0.85, we can observe that the case with $\gamma_k = 0.85$ yields a faster convergence for the leader's solution compared with the other two cases. Moreover, similar result is also observed in Figure 5. The result suggests that a smaller value of γ_k yields a faster convergence of the sequence $\{\epsilon_k,\ k = 1, 2, \cdots\}$ and hence enhances the convergence rate of the bilevel method, e.g. $\gamma_k = 0.85$ requires a less number of iterations to reach a neighborhood close to the equilibrium solutions.

Meanwhile, in the numerical tests, we also observe a fact that a small value of γ_k will also make ϵ_k extremely small from the beginning steps, and hence will increase the computation time for the lower level algorithm (Algorithm 1). Consequently, when choosing γ_k, a trade off should be made between the number of iterations in the upper level algorithm and the computation time for each iteration. To clarify this point, in the following content, we numerically study the convergence of the bilevel direct search method integrally taking the number of iterations in the lower level algorithm into account, where the number of the iterations in the lower level algorithm determines the computation time for each iteration in the upper level algorithm.

All the results in Figures 2 and 3 give how x_k, y_k, $F(x_k, y_k)$ and $f(x_k, y_k)$ change along with the number of iterations in the upper level algorithm. However, every iteration in Algorithm 2 integrally generate a different mount of computation load in the lower level algorithm, i.e. Algorithm 1 with input initial guess (x_k, y_k) and error tolerance ϵ_k. In the rest of content, we present the convergence of the bilevel algorithm with respet to the number of iterations in the lower level algorithm (see x-axle in Figures 6 and 7). In Figures 6 and 7, the x-axle represents the number of both upper and lower level iterations, that is, $k' = \sum_{t<k} L_t + l$ with k being the number of current iteration number in the upper level algorithm, l being the number of current iteration in the lower level algorithm and L_t being the total number of iterations at the lower level when the upper level algorithm is at iteration t for $t = 1, 2, \cdots, k-1$.

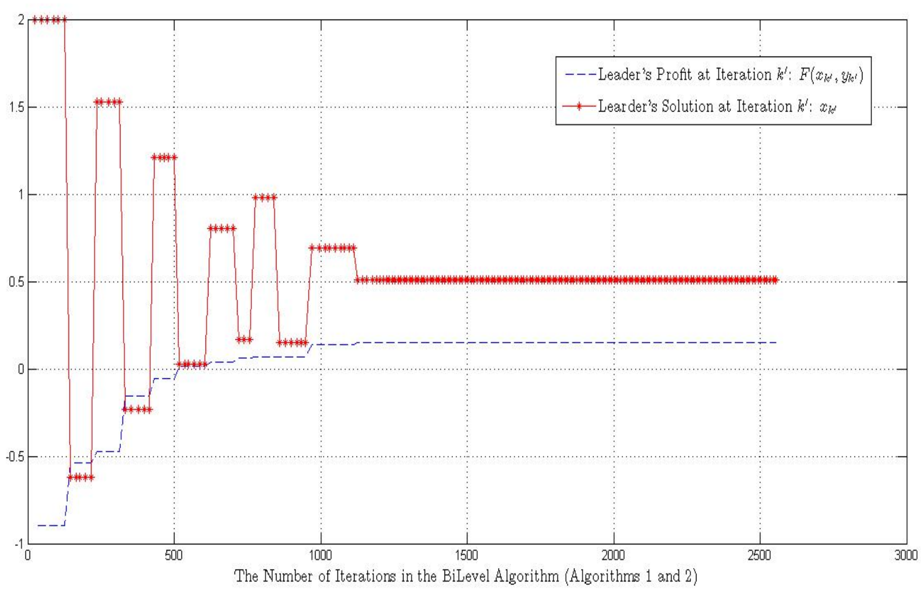

Figure 6: The leader's solution at every iteration for different γ_k

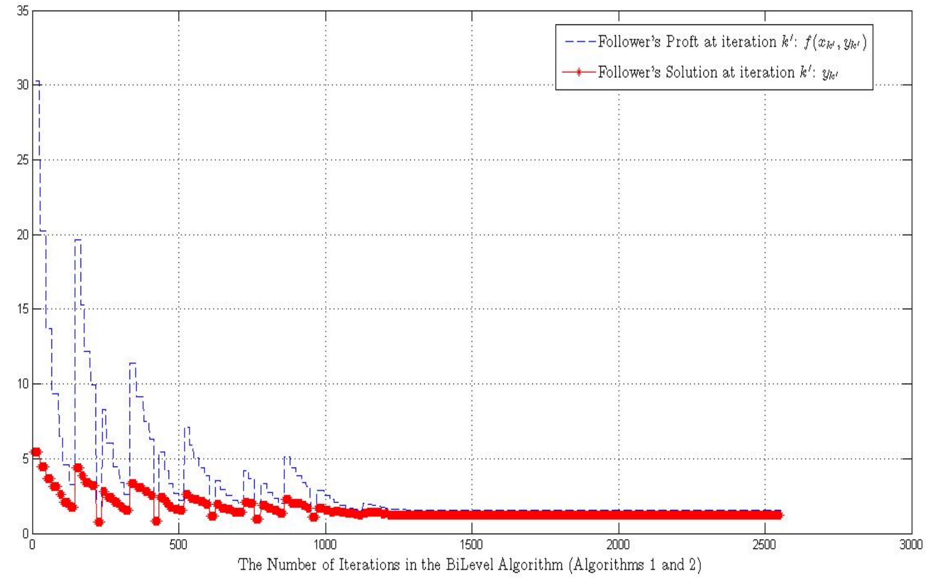

Figure 7: The follower's solution at every iteration for different γ_k

5.3 Case II: Multi-follower problem

In this subsection, we extend our bilevel method to solve the single-leader-multi-follower problems. To this end, we need to generalize the lower level algorithm to solve a Nash equilibrium problem, where in Section 3, Algorithm 1 can be only used for solving a single-follower's optimization problem. To our knowledge, the established results on the derivative free methods for Nash equilibrium is very limited, where the most recent one is a variation of trust region method proposed by Yuan [43]. Since our strengthen in this subsection is not in designing a new bilevel method, we will use Algorithm 2.1 in [43] (*A Jocobi-Type Trust Region Algorithm for Nash Equilibrium Problem*) for solving the lower level problem.

The underlying reason of implementing this trust region method is two-fold. By doing so, on the one hand, we show that the bilevel method can be applied for solving a broader set of leader-follower problems. On the other hand, we can prove that the direct search method is not the only option for this type of algorithms with two-stage structure and "black-box" objective functions. Algorithm 2.1 in [43] is designed for solving a single-level Nash equilibrium problem, and some variations have to be made to integrate it into the bilevel level problems. Due to the aim of giving an example to show a broader application field for the bilevel method (Algorithm 1 and Algorithm 2) in Section 3, we omit the convergence analysis on the trust region algorithm for lower level Nash equilibrium problem here. We suggest the readers to refer to [43] for details about the lower level trust region algorithm.

In this subsection, we consider an oligopolistic market with a dominant firm in the leader position and three firms in the follower position, which supply a homogeneous product in a noncooperative manner. On the one hand, the leader, being the dominant supplier in the market, has to decide now what her future supply x will be. We can imagine that this firm has not yet installed production capacities, and chooses her future supply level by taking into account the reaction of the other producer. On the other hand, the followers indexed by $f = 1, 2$ and 3 react after knowing the leader's supply level. In the model, we assume that every participator chooses his supply level maximizing his profit.

At the *demand side*, the effect, of the total supply level from both firms on the market price, is characterized by an (inverse) demand function $p(x+Y)$ with $Y = y_1 + y_2 + y_3$ being the total supply from the followers. The inverse demand function $p(Q)$ gives the price at which consumers will demand a quantity $Q = x + Y$. In the model, we consider a linear demand function, and set

$$p(Q) = A - BQ, \tag{5.7}$$

where the intersection $A = 5$ and the slope $B = 1$. Moreover, the cost functions of both firms are assumed to be linear as

$$\begin{array}{ll} \text{Leader's cost:} & c_l(x) = c_l x, \\ \text{Follower's cost:} & c_f(y) = c_f y, \end{array}$$

where $c_l = 2$ and $c_f = 2, 2.2$ and 1.6 for $f = 1, 2,$ and 3 respectively. Therefore, the leader's decision problem can be written as the following bilevel programming problem:

$$\left\{ \begin{array}{ll} \max & p(x+y)x - c_l(x) \\ \text{s.t.} & x \in \mathcal{X}, \\ & y(x) \in \left\{ y \in \mathcal{Y} \,\middle|\, y \text{ solves } \max p(x+y)y - c_f(y) \right\}, \quad f = 1, 2, 3. \end{array} \right. \tag{5.8}$$

To perform the algorithm, we set the initial point at $x = 0$ and $y = (2, 2, 2)$ which is far away from the optimal solution. In the example, we also set the initial error tolerant as $\epsilon = 0.1$. In the following table, we present the summary of the iterates in the method, where the true optimal solution of the problem is $x = 1.4, y = (0.35, 0.15, 0.75)$.

Table 2: Summary of the iterates

# of iterate	leader's decision and profit	follower's decision and profit
N = 10	2.621, 0.224	(0.447, 0.020), (-0.155, 0.024), (0.445, 0.198)
N = 50	1.564, 0.483	(0.309, 0.096), (0.109, 0.012), (0.709, 0.503)
N = 90	1.379, 0.490	(0.354, 0.126), (0.155, 0.024), (0.753, 0.571)
N = 130	1.402, 0.490	(0.350, 0.122), (0.150, 0.022), (0.750, 0.562)
N = 170	1.400, 0.490	(0.350, 0.121), (0.150, 0.022), (0.750, 0.563)

6 Conclusions

We have proposed a distirbuted bilevel structured method for solving a leader-follower bilevel optimization problem with "black-box" objective functions for both the leader and the follower. By doing so, we have also showed that different types of derivative free methods, including direct search methods and trust region methods, can be jointly applied to solve this type of problems. The structure of the proposed methods also enables a distributed and parallel computation for the equilibrium, where the tentative decision of each decision maker only need to be communicated to each other during the end of each phase of computation. Moreover, we have presented both theoretical analysis and numerical evidence to show the convergence of the proposed method. The bilevel method proposed in the paper extended the application of direct search methods in bilevel algorithm to cases where the solutions to the lower level problem is not implicitly available. In the future work, we will try to extend our study to the application of derivative free methods for bilevel programming problems under uncertain environment and with coupling constraints.

Acknowledgements. The authors would like to thank Professor Luis Nunes Vicente for his valuable comments on an earlier version of the paper and bringing to their attention of references [8, 36]. The first author would gratefully acknowledge helpful comments on the health insurance model from Professor Oh Hong-Choon.

References

[1] M.A. Abramson and C. Audet, Convergence of mesh adaptive direct search to second-order stationary points, *SIAM Journal on Optimization*, Vol. 17, pp. 606–619, 2006.

[2] C. Audet and Jr. J.E. Dennis, Analysis of generalized pattern search algrithm, *SIAM Journal on Optimization*, Vol. 13, pp. 889–903, 2003.

[3] D.P. Bertsekas, Distributed asynchronous computation of fixed points, *Mathematical Programming*, Vol. 27, pp. 107–120, 1983.

[4] D.P. Bertsekas and J.N. Tsitsiklis, Some aspects of parallel and distributed iterative algorithms: A survey, *Automatica*, Vol. 27, pp. 3–21, 1991.

[5] H.I. Bozma, Computation of Nash equilibria: Admissibility of parallel gradient descent, *Journal of Optimization Theory and Applications*, Vol. 90, pp. 45–61, 1996.

[6] F.H. Clarke, Optimization and Nonsmooth Analysis, Wiley, New York, 1983.

[7] B. Colson, P. Marcotte and G. Savard, A trust-region method for nonlinear bilevel programming: Algorithm and computational experience, *Computational Optimization and Applications*, Vol. 30, pp. 211–227, 2007.

[8] A.R. Conn and L.N. Vicente, Bilevel derivative free optimization and its application to robust optimization, to appear in *Optimization Methods and Software*, 2012.

[9] W.C. Davidon, Variable metric method for minimization, *SIAM Journal on Optimization*, Vol. 1, pp. 1–17, 1991.

[10] V. DeMiguel and H. Xu, A stochastic multiple leader Stackelberg model: Analysis, computation, and application, *Operations Research*, Vol. 57, pp. 1220–1235, 2009.

[11] S. Dempe, Foundations of Bilevel Programming, Kluwer Academic, Dordrecht, 2002.

[12] D. DeWolf and Y. Smeers, A stochastic version of a Stackelberg-Nash-Cournot equilibrium model, *Management Sciences*, Vol. 43, pp. 190–197, 1997.

[13] K. Eggleston and W. Yip, Hospital competition under regulated prices: Application to urban health sector reforms in China, *International Journal of Health Care Finance and Economics*, Vol. 4, pp. 343–368, 2004.

[14] F. Facchinei and J.S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Springer, 2003.

[15] R. Hooke and T.A. Jeeves, Direct search solution of numerical and statistical problems, *Journal of the Association for Computing Machinery*, Vol. 8, pp. 212–229, 1961.

[16] X. Hu and D. Ralph, Using EPECs to model bilevel games in restructured electricity markets with locational prices, *Operations Research*, Vol. 55, pp. 809–827, 2007.

[17] H. Jiang and D. Ralph, Extension of quasi-Newton methods for mathematical programs with complementarity constraints, *Computational Optimization and Applications*, Vol. 25 pp. 123–150, 2003.

[18] T.G. Kolda, R.M. Lewis and V. Torczon, Optimization by direct search: New perspectives on some classical and morden methods, *SIAM Review*, Vol. 45, pp. 385–482, 2003.

[19] A.A. Kulkarni and U.V. Shanbhag, Global equilibria of EPECs with shared constraints, *Manuscript*, Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, 2010.

[20] R.M. Lewis, V. Torczon and M.W. Trosset, Direct search methods: Then and now, *Journal of Computational and Applied Mathematics*, Vol 124, pp. 191–207, 2000.

[21] S. Leyffer and T. Munson, Solving multi-leader-follower games, *Optimization Methods and Software*, Vol. 25, pp. 601–623, 2010.

[22] S. Li and T. Basar, Distributed algorithms for the computation of noncooperative equilibria, *Automatica*, Vol. 23, pp 523–533, 1987.

[23] G.H. Lin and M. Fukushima, Stochastic equilibrium problems and stochastic mathematical programs with equilibrium constraints: A survey, *Pacific Journal of Optimization*, Vol. 6, pp. 455–487, 2010.

[24] A.G. Mersha and S. Dempe, Direct search algorithm for bilevel programming problems, *Computational Optimization and Applications*, Vol. 49, pp. 1–15, 2011.

[25] B.S. Mordukhovich, Equilibrium problems with equilibrium constraints via multiobjective optimization, *Optimization Methods and Software*, Vol. 19, pp. 479–492, 2004.

[26] B.S. Mordukhovich, Optimization and equilibrium problems with equilibrium constraints, *Omega*, Vol. 33, pp. 379–384, 2005.

[27] M.H. Ngo and V. Krishnamurthy, Game theoretic cross-layer transmission policies in multipacket reception wireless networks, *IEEE Transaction on Signal Processing*, Vol. 55, pp. 1911–1926, 2007.

[28] G. P. Papavassilopoulos, Adaptive repeated Nash games, *SIAM Journal on Control and Optimization*, Vol. 24, pp. 821–834, 1986.

[29] E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press, New York, 1971.

[30] D. Popović and A.R. Teel, Direct search methods for nonsmooth optimization, *Proceeding of the 43rd IEEE Conference on Decision and Control*, 2004.

[31] A. Ruszczyński and A. Shapiro, Optimality conditions and duality in stochastic programming, *Stochastic Programming*, Handbooks in OR & MS, Vol. 10, A. Ruszczyński and A. Shapiro eds, North-Holland Publishing Company, Amsterdam, pp.141–211, 2003.

[32] A. Shapiro and H. Xu, Stochastic mathematical programs with equilibrium constraints, modeling and sample average approximation, *Optimization*, Vol. 57, pp. 395–418, 2008.

[33] U. Shanbhag, Decomposition and Sampling Methods for Stochastic Equilibrium Problems, PhD thesis, Stanford University, December 2005.

[34] H.D. Sherali, A.L. Soyster and F.H. Murphy, Stackelberg-Nash-Cournot equilibria: Characterizations and computations, *Operations Research*, Vol.31, pp. 253–276, 1983.

[35] H. Von Stackelberg, Marktform und Gleichgewicht, English Translation, The Theory of The Market Economy, Oxford University Press, 1952.

[36] L.N. Vicente and A.L. Custódio, Analysis of direct searches for discontinuous functions, to appear in *Mathematical Programming.*

[37] V. Torzcon, On the convergence of the multidirectional search algorithms, *SIAM Journal on Optimization*, Vol. 1, pp. 123–145, 1991.

[38] V. Torzcon, On the convergence of pattern search algorithm, *SIAM Journal on Optimization*, Vol. 7, pp. 1–25, 1997.

[39] H. Xu, An MPCC approach for stochastic Stackelberg-Nash-Cournot equilibrium, *Optimization,* Vol. 54, pp. 27–57, 2005.

[40] H. Xu, An implicit programming approach for a class of stochastic mathematical programs with linear complementarity constraints, *SIAM Journal on Optimization*, Vol. 16, pp. 670–696, 2006.

[41] H. Xu and D. Zhang, Stochastic Nash equilibrium problems: sample average approximation and applications, to appear in *Computational Optimization and Applications*, 2012.

[42] J. Yao, S.S. Oren and I. Adler, Two-settlement electricity markets with price caps and Cournot generation firms, *European Journal of Operational Research,* Vol. 181, pp. 1279–1296, 2007.

[43] Y. Yuan, A trus region algorithm for Nash equilibrium problems, *Pacific Journal of Optimization*, Vol. 7, pp. 125–138, 2011.

[44] C. Zhang and X. Chen, Smoothing projected gradient method and its application to stochastic linear complementarity problems, *SIAM Journal on Optimization*, Vol. 20, pp. 627–649, 2009.

[45] D. Zhang and H. Xu, Two stage stochastic equilibrium problems with equilibrium constraints: Modeling and numerical schemes, to appear in *Optimization*, 2011.

[46] D. Zhang, H. Xu and Y. Wu, A two stage stochastic equilibrium model for electricity markets with two way contracts, *Mathematical Methods of Operations Research*, Vol. 71, pp. 1–45, 2010.