Bi-level Decision Making with Fuzzy Sets and Particle Swarm Optimisation

A thesis submitted for the degree of
Doctor of Philosophy

By

Ya Gao

June 2010

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgement

I wish to express my earnest and deepest gratitude and admiration to my principal supervisor A/Professor Guangquan Zhang. He has given me his precious guidance and wholehearted help during the whole of my PhD period. His profound knowledge and sincere attitude has guided me through my PhD, and these things will benefit me in my future research career. I also wish to express my earnest and deepest gratitude and admiration to my co-supervisor Professor Jie Lu. She has given me wholehearted advice and tremendous help, without which my PhD research could not even have begun. She has given me precious guidance in the academic area through her profound knowledge, and has also provided sincere solicitude and advice for my life. The dedicated help and love from these two people have not only encouraged me to finish my PhD research and this thesis, but are also things I will treasure all my life. This thesis would never have been written without their invaluable advice and suggestion, enormously dedicated assistance, careful reading and troubleshooting many, many times. I would also like to express my appreciation to my external supervisor, Professor Tharam Dillon, from the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, for his generous comments and suggestions on my PhD research.

I would thank Professor Guoli Zhang, from the Department of Mathematics and Physics, North China Electric Power University, China and Professor Hui-Ming Wee, from the Department of Industrial and Systems Engineering, Chung Yuan Christian University, Taiwan for their precious help and comments during our joint research on the applications of bi-level decision making.

Thanks to the Australian Research Council (ARC) for supporting the work presented in this thesis under discovery grant DP0557154. The work presented in this thesis was undertaken while I was the recipient of a research scholarship award from the Faculty of Engineering and Information Technology, UTS and an Australian Postgraduate Award. The support was most welcome.

My gratitude to all members from the Decision Systems and e-Service Intelligence Laboratory. They have given me many precious suggestions and comments which have been tremendously helpful to my PhD research.

I want to express my heartfelt appreciation to my family. It is the love and support from my husband, Jun Ma, my parents Siwei Gao and Jinzhi Wang, my parents-in-law Zuhua Ma and Fengqin Wu, that have helped me to make it through so many hard times in accomplishing this thesis.

I cannot forget all my friends in both China and Australia whose love and concern have made me strong and brave, and will always continue to do so.

Finally, I would like to express my gratitude to the Faculty of Engineering and Information Technology, UTS for providing me with the necessary facilities and financial support, and to all staff of the faculty for their help and encouragement during my study at UTS.

Abstract

Bi-level programming techniques are developed for decentralized decision problems with decision makers located in a two-level decision making system; the upper decision maker is termed the leader while the lower is the follower. Both the leader and the follower try to optimise their own objective functions and the corresponding decisions do not control but do affect those of the other level.

This research aims at solving bi-level decision problems with five extensions, i.e. multiple leaders/followers/objectives, fuzzy coefficients and goals. By using particle swarm optimisation and/or cut set and/or goal programming and/or Nash equilibrium concept, related mathematical models and corresponding algorithms are developed to solve fuzzy linear bi-level decision problems, fuzzy linear multi-objective bi-level decision problems, fuzzy linear multi-follower multi-objective bi-level decision problems, fuzzy linear goal bi-level decision problems, multi-leader one-follower bi-level decision problems, one-leader multi-follower bi-level decision problems, and multi-leader multi-follower bi-level decision problems. A fuzzy bi-level decision support system is then developed which implements all the algorithms to support bi-level decision making with different features. Finally, by using these bi-level models and algorithms, we explore possible applications in the fields of railway train set organisation, railway wagon flow management, strategic bidding in the electricity market, and supply chains to solve real world bi-level decision problems. The results of experiments show that the models and algorithms are effective for solving real world bi-level decision problems.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

Bi-level programming technologies, initiated by Von Stackelberg (1952), are mainly developed for solving decentralised management problems with decision makers in a hierarchical organisation, with the upper termed the leader and the lower the follower (Bard 1998). In a bi-level decision making, the control of decision factors is partitioned amongst the leader and follower who seek to optimise their individual objective functions, and the corresponding decisions do not control but do affect that of the other level (Aiyoshi & Shimizu 1981). The leader attempts to optimise his or her objective function but he or she must anticipate all possible responses from the follower (Lai 1996). The follower observes the leader's decision and then responds to it in a way that is personally optimal. Because the set of feasible choices available to either decision maker is interdependent, the leader's decision affects both the follower's payoff and allowable actions, and vice versa.

To motivate a quick understanding of bi-level decision problems, we now give an example as an illustration. The example is from the relationship between a manufacturer and a retailer. Suppose the articles involved are newspapers: the retailer orders newspapers from the manufacturer and sells them to the readers. Both the manufacturer and the retailer wish to make as much profit as possible from their newspaper sale. The equations to calculate the manufacturer's profits (F) and the retailer's profit (f) are as follows:

$$F = (C - D) \cdot Q$$
$$f = \begin{cases} (A - C)\xi,\ Q < \xi \\ \\ (A - C)\xi - C(Q - \xi),\ Q \geqslant \xi \end{cases}$$

where D is the manufacturing cost, C is the wholesale price per unit, Q is the quantity ordered by the retailer, ξ is the quantity sold by the retailer, and A is the retail price.

To maximise the profit, the manufacturer wishes the wholesale price and order quantity to be as large as possible. However, the manufacturer can only control the wholesale price, while order quantity is determined by the retailer. Even if the wholesale price is controlled by the manufacturer, it is not the case that the larger the wholesale price, the higher the profit, because when the wholesale price increases, the retailer probably decreases the order quantity to avoid profit loss. Now the problem is how to decide the value of the wholesale price for the manufacturer to maximise profit. This is a typical bi-level decision problem. The manufacturer is the leader and the retailer is the follower.

To formulate a bi-level decision problem, suppose the leader controls the vector $x \in X \subseteq R^n$, while the follower has control over $y \in Y \subseteq R^m$. The leader moves first by selecting an x in an attempt to minimise his or her objective function $F(x, y)$ subject to certain constraints. Then, the follower observes the leader's action and reacts by choosing a y to minimise his or her own objective function $f(x, y)$ under some constraints as well. Thus, a bi-level decision problem is formatted as follows:

Definition 1.1.1. (Bard 1998) For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, a bi-level decision problem is defined as:

$$\min_{x \in X} F(x, y) \tag{1.1a}$$

$$\text{subject to } G(x, y) \leqslant 0 \tag{1.1b}$$

$$\min_{y \in Y} f(x, y) \tag{1.1c}$$

$$\text{subject to } g(x, y) \leqslant 0 \tag{1.1d}$$

where $F : R^n \times R^m \to R^1$, $G : R^n \times R^m \to R^p$, $f : R^n \times R^m \to R^1$, and $g : R^n \times R^m \to R^q$.

Once all functions defining the bi-level decision problem in (1.1) are restricted to being affine, i.e. have linear formats, the problems will become linear bi-level decision problems.

In a real case, to formulate bi-level decision problems, the coefficients of the objective functions and the constraints are sometimes obtained through experiments or experts' understanding of the nature of those coefficients. It has been observed that, in most situations, the possible values of these coefficients are often only imprecisely or ambiguously known to the experts and cannot be described by precise values. With this observation, it would certainly be more appropriate to interpret the experts'understanding of the coefficients as fuzzy numerical data which can be represented by means of fuzzy sets (Zadeh 1965). Bi-level linear decision problems in which the coefficients are characterised by fuzzy numbers are called fuzzy linear bi-level decision problems.

Definition 1.1.2. (Zhang, Lu & Dillon 2007a) A fuzzy linear bi-level (FLB) decision problem is defined as :

For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, $F : X \times Y \to F(R)$, and $f : X \times Y \to F(R)$,

$$\min_{x \in X} F(x, y) = \tilde{c}_1^T x + \tilde{d}_1^T y \tag{1.2a}$$

$$\text{subject to } \tilde{A}_1 x + \tilde{B}_1 y \preceq \tilde{b}_1 \tag{1.2b}$$

$$\min_{y \in Y} f(x, y) = \tilde{c}_2^T x + \tilde{d}_2^T y \tag{1.2c}$$

$$\text{subject to } \tilde{A}_2 x + \tilde{B}_2 y \preceq \tilde{b}_2 \tag{1.2d}$$

where $\tilde{c}_1$, $\tilde{c}_2 \in F^n(R)$, $\tilde{d}_1$, $\tilde{d}_2 \in F^m(R)$, $\tilde{b}_1 \in F(R^p)$, $\tilde{b}_2 \in F^q(R)$, $\tilde{A}_1 = (\tilde{a}_{ij})_{p \times n}$, $\tilde{a}_{ij} \in F(R)$, $\tilde{B}_1 = \left(\tilde{b}_{ij}\right)_{p \times m}$, $\tilde{b}_{ij} \in F(R)$, $\tilde{A}_2 = (\tilde{e}_{ij})_{q \times n}$, $\tilde{e}_{ij} \in F(R)$, $\tilde{B}_2 = (\tilde{s}_{ij})_{q \times m}$, $\tilde{s}_{ij} \in F(R)$, and $F(R)$ is the set of all finite fuzzy numbers.

In a bi-level decision problem, the decision makers from either level may have several objectives which should be considered simultaneously. Often, these objectives may be in conflict with each other, with any improvement in one achieved only at the expense of others. Fuzzy multi-objective linear bi-level decision problems are thus defined to model and solve linear fuzzy bi-level decision problems in which several conflicting objectives, for either the leader or the follower, are to be optimised simultaneously.

Definition 1.1.3. (Zhang, Lu & Dillon 2007b) A fuzzy multi-objective linear bi-level (FMOLB) decision problem is defined as:

For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, $F : X \times Y \to F^s(R)$, and $f : X \times Y \to F^t(R)$,

$$\min_{x \in X} F(x, y) = (\tilde{c}_{11}^T x + \tilde{d}_{11}^T y, \tilde{c}_{21}^T x + \tilde{d}_{21}^T y, ..., \tilde{c}_{s1}^T x + \tilde{d}_{s1}^T y)^T \tag{1.3a}$$

$$\text{subject to } \tilde{A}_1 x + \tilde{B}_1 y \preceq \tilde{b}_1 \tag{1.3b}$$

$$\min_{y \in Y} f(x, y) = (\tilde{c}_{12}^T x + \tilde{d}_{12}^T y, \tilde{c}_{22}^T x + \tilde{d}_{22}^T y, ..., \tilde{c}_{t2}^T x + \tilde{d}_{t2}^T y)^T \tag{1.3c}$$

$$\text{subject to } \tilde{A}_2 x + \tilde{B}_2 y \preceq \tilde{b}_2 \tag{1.3d}$$

where $\tilde{c}_{h1}$, $\tilde{c}_{i2} \in F^n(R)$, $\tilde{d}_{h1}$, $\tilde{d}_{i2} \in F^m(R)$, $h = 1, 2..., s$, $i = 1, 2, ...t$, $\tilde{b}_1 \in F^p(R)$,

$\tilde{b}_2 \in F^q(R)$, $\tilde{A}_1 = (\tilde{a}_{ij})_{p\times n}$, $\tilde{B}_1 = \left(\tilde{b}_{ij}\right)_{p\times m}$, $\tilde{A}_2 = (\tilde{e}_{ij})_{q\times n}$, $\tilde{B}_2 = (\tilde{s}_{ij})_{q\times m}$, $\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{e}_{ij}, \tilde{s}_{ij} \in F(R)$.

Although much research has been carried out in the area of bi-level programming, existing technologies have mainly focused on a specific situation comprising only one leader and one follower. In cases of real world bi-level decision problems, however, the lower level of a bi-level decision may involve more than one decision unit. The leader's choice is therefore affected by the objectives and strategies of his or her lower counterparts. For each possible decision from the leader, those followers may have their own different reactions. The relationships among these multiple followers could be complex: they may or may not share their decision variables; they may have individual objectives and constraints but work with others cooperatively, or they may have common objectives or common constraints (Lu, Shi & Zhang 2006). For example, in the newsboy problem, more than one retailer (followers) may be involved. The manufacturer (leader) tries to establish the most suitable wholesale price to enlarge his or her profits that is bound to be influenced by the responses from different retailers. Each retailer has his or her own individual policies to optimise the objective towards different wholesale prices decided by the manufacturer. These followers may share the same decision variables, or may have the same objectives or constraints when specific interests are involved. In such cases, the decision of the manufacturer (the leader) is partially dependent on the environment data put forward by all these retailers (the followers). This is a typical multi-follower bi-level decision problem.

In real world bi-level decision problems, there are also decision situations, although relatively scarce, in which more than one leader is involved. For instance, in an electricity bidding market, each generating company needs to submit a set of hourly generation prices and available capacities for each of their electricity generation units for the following day. According to these data and an hourly load forecast, the market operator allocates the generation output for each unit. The generating companies, located in the upper level, wish their profit to be as high as possible, while the operator, as the follower, pursues the lowest cost. In this bi-level decision problem, unlike classical bi-level problems, there is more than one leader.

In a bi-level decision making process, the leader or follower may set a goal for the objective that he/she wishes to attain. A preferred solution is then defined to minimise the deviation from the goal. We again take the newsboy problem as the example. Although both the manufacturer and retailer wish to maximise their profits, they may have some expected profit levels (goals) on their minds. As long as the actual profits reach these levels, they will be satisfied. Therefore to set goals for objectives aims to yield a satisfactory solution rather than an optimal one.

This research addresses bi-level decision problems with these particularities: fuzzy coefficients, multiple objectives for decision makers located either in the upper or the lower level, more than one followers, many leaders, and goals. Based on the combination of these specialties, seven bi-level decision problems are identified, i.e. FLB decision problems, FMOLB decision problems, fuzzy linear multi-follower multi-objective (FMMLB) bi-level decision problems, fuzzy linear bi-level goal (FLBG) decision problems, multi-leader one-follower bi-level (MLOFB) decision problems, one-leader multi-follower (OLMFB) bi-level decision problems, and multi-leader multi-follower bi-level (MLMFB) decision problems.

These seven problems are the precise research issues focused on this thesis.

1.2 Objectives

Based on the research issues discussed in Section 1.1, three objectives are proposed in this research.

(1) To develop approaches for bi-level decision problems.

In this research, bi-level decision problems are classified into different categories by considering factors of fuzzy/crisp coefficients, linear/non-linear formulas, multiple/single objective(s), multiple/single leader(s), multiple/single follower(s), and objective/goal optimisation strategy. Based on these factors, which impose tremendous influence when formulating bi-level decision problems, seven different kinds of bi-level decision problems are addressed in this research: FLB decision problems, FMOLB decision problems, FMMLB decision problems, FLBG decision problems, MLOFB decision problems, OLMFB decision problems, and MLMFB decision problems. The mathematical definitions of these particular bi-level decision problems will be provided and corresponding algorithms for solutions will be developed.

(2) To develop a software system to support bi-level decision making.

This decision support system will be able to identify and build up frameworks for the seven kinds of bi-level decision problems discussed above. Solutions for each of these problems will be derived with the help of the algorithms integrated in the system at the back-end. As a decision making system, it also has functions of problem identification, data collection and processing, model input, method selection, and visualised solution supply.

(3) To explore applications for the bi-level decision making system.

To apply the proposed techniques in this study and to test the correctness and efficiency of the proposed models and algorithms, applications in railway transportation, electricity market, and supply chains are developed.

1.3 Contributions

Corresponding to the objectives of my research described in Section 1.2, my doctorial research produces mainly eight contributions:

(1) Mathematical models for FLBG, FMMLB, MLOFB, OLMFB and MLMFB decision problems. (Objective 1)

(2) PSO-based algorithms for FLB, MLOFB, OLMFB and MLMFB decision problems. (Objective 1)

(3) λ-cut and goal programming-based algorithms for FLBG and FMOLB decision problems. (Objective 1)

(4) A fuzzy bi-level decision support system that supports bi-level decision making from multiple angles. (Objective 2)

(5) A bi-level decision model for railway train set organising optimisation. (Objective 3)

(6) An OLMFB decision model on railway wagon flow management. (Objective 3)

(7) An MLOFB decision model in electricity markets. (Objective 3)

(8) Bi-level pricing models in supply chains. (Objective 3)

1.4 Organisation of this Thesis

This doctorial thesis consists of ten chapters:

Chapter 1 presents an overview of this research, including research issues, research objectives, and research contributions.

Chapter 2 reviews the related research areas, including bi-level programming, linear bi-level programming, multi-objective linear bi-level (MOLB) programming, multi-follower bi-level (MFB) programming, multi-leader bi-level (MLB) programming and fuzzy bi-level programming, the relationship between bi-level programming and other optimisation problems, complexity and optimality conditions of bi-level problems, applications of bi-level programming techniques, bi-level decision support systems, and particle swarm optimisation techniques.

Chapter 3 studies FLB decision problems by presenting a cutset-based decision model and developing a particle swarm optimisation (PSO)-based algorithm for solving them.

Chapter 4 studies FMOLB decision problems by presenting a cutset-based decision model and developing a $\lambda-$cut and goal-programming-based algorithm for solving them.

Chapter 5 studies FMMLB decision problems. A model framework is proposed to define FMMLB problems by different cooperation in objectives, constraints, and decision variables among followers. Then three algorithms, i.e. a Branch-and-Bound-based algorithm, a Kth-Best-based algorithm, and a PSO-based algorithm are developed for solving them.

Chapter 6 studies FLBG decision problems by proposing a cutset-based decision model and developing an approximate algorithm for solving them.

Chapter 7 studies general bi-level problems where the objectives and constraints for leaders and followers may have arbitrary formats. In particular, MLOFB, OLMFB, and MLMFB decision problems are addressed by giving the mathematical definitions and developing PSO-based algorithms to solve them.

Chapter 8 presents the development of a fuzzy bi-level decision support system, which implements the algorithms developed in Chapters 3, 4, 5, 6 and 7 to support bi-level decision making.

Chapter 9 applies the bi-level programming techniques developed in this study on real world problems including railway transportation, electricity markets, and supply chains.

Chapter 10 summarises the entire thesis and highlights the future research work.

1.5 Publications Related to this Thesis

Below is the list of my published, accepted and submitted papers during my PhD study.

Published and accepted:

(1) Ya Gao, Guangquan Zhang, Jun Ma, Jie Lu, "A λ-cut and Goal Programming based Algorithm for Fuzzy Linear Multiple Objective Bi-level Optimisation", IEEE Transactions on Fuzzy Systems, 2010, Vol. 18, No. 1, pp. 1-13, ISSN: 1063-6706.

(2) Ya Gao, Guangquan Zhang, Jie Lu, Hui-Ming Wee, "Particle Swarm Optimization for Bi-level Pricing Problems in Supply Chains", accepted by Journal of Global Optimization, 2009, ISSN: 0925-5001.

(3) Ya Gao, Guangquan Zhang, Jie Lu, Hui-Ming Wee, "A Fuzzy Bi-level Pricing Model and a PSO Based Algorithm in Supply Chains", The 16th International Conference on Neural Information Processing (ICONIP 2009), December 1-5, 2009, Bangkok, Thailand, pp. 226-233, ISBN: 978-3-642-10682-8.

(4) Ya Gao, Guangquan Zhang, Jie Lu, Hui-Ming Wee, "A Bi-level Pricing Model and a PSO Based Algorithm in Supply Chains", The 4th International Conference on Intelligent Systems & Knowledge Engineering (ISKE 2009), November 27-28, 2009, Hasselt, Belgium, pp. 394-401, ISBN: 978-981-4295-05-5.

(5) Hui-Ming Wee, Jie Lu, Guangquan Zhang, Huai-En Chiao, Ya Gao "A Decision Making Model for Vendor-buyer Inventory Systems", The Third International Conference on Pattern Recognition & Machine Intelligence (PReMI'09), December 16-18, 2009, Delhi, India, pp. 336-343, ISBN: 978-3-642-10645-3.

(6) Guangquan Zhang, Guoli Zhang, Ya Gao, Jie Lu, "A Fuzzy Bilevel Model and a PSO-based Algorithm for Day-ahead Electricity Market Strategy Making", The 13th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2009), September 28-30, 2009, Santiago, Chile, pp. 736-744, ISBN: 978-3-642-04594-3.

(7) Guoli Zhang, Guangquan Zhang, Ya Gao, Jie Lu, "A Bilevel Optimisation Model and a PSO-based Algorithm in Day-ahead Electricity Markets", The 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC2009), October 11-14, 2009, Texas, USA, pp. 617-622.

(8) Ya Gao, Guangquan Zhang, Jie Lu, "A Particle Swarm Optimization Approach to Fuzzy Bilevel Decision Making with Constraints-shared Followers", the 24th

Annual ACM Symposium on Applied Computing (SAC 2009), March 08-12, 2009, Hawaii, USA, pp. 1075-1079.

(9) Ya Gao, Guangquan Zhang, Jie Lu, "A Fuzzy Multi-Objective Bilevel Decision Support System", International Journal of Information Technology and Decision Making, 2009, Vol. 8, No. 1, pp. 93-108, ISSN: 0219-6220.

(10) Ya Gao, Guangquan Zhang, Jie Lu, Tharam Dillon, Xiangyi Zeng, "A λ-cut Approximate Algorithm for Goal-based Bilevel Risk Management Systems", International Journal of Information Technology and Decision Making, 2008, Vol. 7, No. 4, pp. 589-610, ISSN: 0219-6220.

(11) Ya Gao, Guangquan Zhang, Jie Lu, "A particle swarm optimisation based algorithm for fuzzy bilevel decision making", 2008 International Conference on Fuzzy Systems (FUZZ2008), 1-6 June, 2008, Hong Kong, pp. 1452-1457, ISBN: 978-1-4244-1823-7.

(12) Ya Gao, Guangquan Zhang, Jie Lu, "A Decision Support System for Fuzzy Bilevel Decision Making", the 8th International FLINS Conference On Computational Intelligence in Decision and Control, September 21-24, 2008, Madrid, Spain, pp. 763-768.

(13) Ya Gao, Guangquan Zhang, Jie Lu, "A Particle Swarm Optimization based Algorithm for Fuzzy Bilevel Decision Making with Objective-shared Followers", the 7th International Conference on Simulated Evolution And Learning (SEAL'08), December 07-10, 2008, Melbourne, Australia, pp. 190-199.

(14) Guangquan Zhang, Jie Lu, Ya Gao, "Fuzzy bilevel programming: Multi-objective and multi-follower with shared variables", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2008, Vol. 2, No.2, pp. 105 - 133, ISSN: 0218-4885.

(15) Guangquan Zhang, Jie Lu, Ya Gao, "A Fuzzy Multi-objective Multi-follower Partial Cooperative Bilevel Programming Algorithm", International Journal of Intelligent & Fuzzy Systems, 2008, Vol. 4,5, No. 19, pp. 303-319, ISSN: 1064-1246.

(16) Ya Gao, Guangquan Zhang, Jie Lu, Xianyi Zeng, "A λ-cut Approximate Approach to Supporting Fuzzy Goal Based Bilevel Decision Making in Risk Man-

agement", The First International Conference on Risk Analysis and Crisis Response, September 25-26, 2007, Shanghai, China, pp. 132-137.

(17) Ya Gao, Guangquan Zhang, Jie Lu, Siwei Gao, "A Bilevel Model for Railway Train Set Organizing Optimization", 2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE2007), October 15-16, 2007, Chengdu, China, pp. 777-782.

Submitted:

(18) Ya Gao, Guangquan Zhang, Jie Lu, Hui-Ming Wee, "Bi-level pricing models and a PSO-based algorithm in a supply chain", submitted to the Journal of Applied Mathematics and Computation, 2009.

(19) Guangquan Zhang, Ya Gao, Jie Lu, "An Interactive Approximation Branch-and-bound Algorithm for Bi-level Linear Programming with Multiple Uncooperative Followers and Fuzzy Coefficients", submitted to the Journal of Multi-Valued Logic and Soft Computing, 2009.

(20) Guangquan Zhang, Guoli Zhang, Ya Gao, Jie Lu, "Competitive Strategic Bidding Optimization in Electricity Markets using Bi-level Programming and Swarm Technique", submitted to IEEE Transactions on Industrial Electronics, 2009.

2 Literature Review

2.1 Bi-level Programming

From a historical point of view, bi-level programming is closely related to the economic problem of kelberg (Stackelberg 1952) in the field of game theory. The original formulation for bi-level programming appeared in 1973, in a paper authored by J. Bracken and J. McGill (1973), although it was W. Candler and R. Norton (1977) who first used the designation "bi-level" programming. However, it was not until the early nineteen eighties that these problems started to receive the attention they deserved.

2.1.1 Bi-level Programming and Other Optimisation Problems

The fact that some important mathematical programs, such as minimax problems, linear integer problems, bilinear problems, and quadratic programming can be viewed as special instances of bi-level problems illustrates the importance of these problems in researching bi-level problems.

Although it is simple to view a minimax problem as a bi-level problem, it was not until 1977 that Gallo and Ulkucu (1977) first exploited the reduction of a bilinear problem to a linear bi-level problem. This result also established that any integer of concave quadratic problem can be converted to a bi-level problem. However, this conversion is not entirely possible since the reciprocal result indicates that there exist a penalised bilinear problem whose global optimal solutions are also global solutions of the corresponding bi-level linear problem.

Although some researchers have attempted to establish a link between two objective optimisation and bi-level problems (Bard 1984a; Unlu 1987), none of them succeeded so far in proposing conditions which guarantee that the optimal solution of a given bi-level problem is pareto optimal or efficient for both upper and lower level objective functions (Shi 2005).

A static Stackelberg problem can differ from a bi-level problem insofar as the upper level function is minimised. If the reaction set of the follower is not a singleton for some selections from the leader, then a solution of the static Stackelberg problem may not be a solution of the bi-level problem (Shi 2005).

2.1.2 Complexity of Bi-level Programming

The difficulty and complexity of solving a bi-level problem is easily confirmed by looking at its simplest version, the linear bi-level problem. Examples of linear bi-level problem with an exponential number of local minima will be generated using the method proposed by Calamai and Vicente (1994). The tightest complexity result is from Hansen et al. (1992), where it has been established that a linear bi-level problem is strongly NP-hard. A linear bi-level problem was then shown to be NP-hard by Jeroslow (1985) using satisfiability arguments common in computer science. Moreover, Vicente et al. (1994) have shown that only checking local optimality in a linear bi-level problem is a NP-hard problem. Bard (1991) then provided an alternative proof by constructively reducing the problem of maximising a strictly convex quadratic function over a polyhedron to a linear minimax problem.

2.1.3 Optimality Conditions of Bi-level Programming

Research on optimality conditions is one of the central topics in bi-level study. Several optimality conditions in bi-level programming have been proposed in the literature.

To formulate optimality conditions, it is often necessary to use the single-level reformulation of a bi-level problem. An early research in this direction of replacing the lower level problem with an infinite number of constraints is shown in (Bard 1984a). The following attempt, which formulates necessary and sufficient optimality conditions, assumes the lower level problem has a unique strongly stable optimal solution (Dempe 2001).

Necessary optimality conditions using the reformulation of a bi-level problem under the help of the optimal value function of the lower level problem were studied by Liu and Han (1997) and Ye (1997). Applying Duality theory to the lower level problem can derive a minimax problem where optimality conditions can be developed (Malhotra & Arora 1999). Necessary optimality conditions of Kuhn-Tucker were studied by Chen et al. (Chen & Florian 1994).

Optimality conditions for set-valued optimisation problems have been derived under different assumptions and have used various differentiability tools. These tools are sometimes restrictive for bi-level problems due to their special structures (Shi 2005). The use of a Farkas-Minkowski theorem of the alternatives is demonstrated by Hwang

(2002). An explicit description of the set valued objective function by finite functions is used in the literature (Craven & Luu 1997).

2.2 Linear Bi-level Programming

2.2.1 Definition of Linear Bi-level Programming

A large part of the research on bi-level programming techniques has centred on its linear version, the linear bi-level decision problem, in which all formulas of both the objective functions and constraints from a leader and the follower are linear functions. The general form of bi-level programming can be defined as:

Definition 2.2.1. (Bem-Ayed 1993)

$$\min_{x \in X} F(x, y) = c_1^T x + d_1^T y \tag{2.1a}$$

$$\text{subject to } A_1 x + B_1 y \leqslant b_1 \tag{2.1b}$$

$$\min_{y \in Y} f(x, y) = c_2^T x + d_2^T y \tag{2.1c}$$

$$\text{subject to } A_2 x + B_2 y \leqslant b_2 \tag{2.1d}$$

where $c_1, c_2, d_1, d_2, b_1, b_2$ are constant vectors; A_1, A_2, B_1, B_2 are constant matrices; x, y are vectors of the decision variables of the upper and lower problems, respectively; F, f are the objective functions of the upper and lower problems, respectively.

Although the definitions of a bi-level decision problem vary considerably from one reference to another, most recent publications tend to agree on Definition 2.2.1 as the general form. Nevertheless, there are many attempts to extend the model and expand its use to more general problems.

2.2.2 Properties for Linear Bi-level Programming

Property 2.2.1. Semi-feasibility (Bem-Ayed 1993)

A point (x, y) is said to be semi-feasible if and only if $A_1 x + B_1 y \leqslant b_1$, $A_2 x + B_2 y \leqslant b_2$, and $x, y > 0$. The set of semi-feasible points is a polyhedron called the semi-feasible region.

Property 2.2.2. Feasibility (Bem-Ayed 1993)

A point (x_0, y_0) is said to be feasible if and only if $x_0 \geqslant 0$ and y_0 is optimal to the lower problem. The set of feasible points is called the feasible region.

(1) The feasible region is composed of a piecewise linear constraint region consisting of a set of edges and hypersurfaces of the semi-feasible region (Bard 1984a).

(2) The feasible region is connected (Gallo & Ulkucu 1977).

Property 2.2.3. Optimality (Bem-Ayed 1993)

A point (x^*, y^*) is said to be optimal if and only if:

(1) (x^*, y^*) is feasible;

(2) for all feasible points (x_0, y_0), $c_1^* + d_1^* \geqslant c_1x_0 + d_1y_0$;

(3) for all feasible points $(x^*, \bar{y})$, if $d_2\bar{y} = d_2y$ then $d_1y^* = d_1\bar{y}$.

The third condition states that if the lower decision maker is indifferent between y^* and y when x is fixed to x^*, then the upper decision maker must also be indifferent between x^*, y^* and $x^*, \bar{y}$.

Property 2.2.4. Convexity properties

The feasible region of a linear bi-level decision problem does not need be convex since it is composed of a set of faces of the semi-feasible polyhedron.

Although not necessarily convex, the feasible region of a linear bi-level decision problem has some of the properties of convex sets:

(1) If the feasible region is compact, then no feasible point can be expressed as a convex combination of points that are semi-feasible or not feasible. Equivalently, if a feasible point x is expressed as a convex combination of semi-feasible points, then the points are also feasible (Gallo & Ulkucu 1977).

(2) Any extreme point of the feasible region is also an extreme point of the semi-feasible region (Bialas & Karwan 1982).

(3) At least one optimal solution of a linear bi-level decision problem, if there is one, occurs at a vertex of the feasible region (Bard 1984a).

(4) At least one optimal solution of a linear bi-level decision problem, if there is one, occurs at a vertex of the polyhedron (Bard 1984a).

Property 2.2.5. Kuhn-Tucker condition (Kuhn & Tucker 1951)

A necessary condition that (x^*, y^*) solved a bi-level decision problem defined by Definition 2.2.1 is that there exist row vectors u^* and v^* such that (x^*, y^*, u^*, v^*) solves:

$$\min_{x \in X} F(x, y) = c_1 x + d_1 y \tag{2.2a}$$

$$\text{subject to } A_1 x + B_1 y \leqslant b_1 \tag{2.2b}$$

$$A_2 x + B_2 y \leqslant b_2 \tag{2.2c}$$

$$u(b_2 - A_2 x - B_2 y) + vy = 0 \tag{2.2d}$$

$$x \geqslant 0, y \geqslant 0, u \geqslant 0, v \geqslant 0 \tag{2.2e}$$

This Kuhn-Tucker condition provides a single-level equivalent formulation for a linear bi-level decision problem.

Property 2.2.6. Inclusion of infimum and supremum constraints (Bem-Ayed 1993)

It means an infimum (inf) constraint has the following form:

$$w = \inf\{y_j : j = 1, \ldots, m\} \tag{2.3}$$

where w and y_j are variables.

Above constraint is equivalent to the following linear programming:

$$\max w \tag{2.4a}$$

$$\text{subject to } w \leqslant w_j, j = 1, \ldots, m \tag{2.4b}$$

The constraints of this linear programming guarantee that w is a lower bound while its objective function guarantees it is the greatest one; therefore it guarantees that w is an infimum since it is the greatest lower-bound of the set $\{y_j : j = 1, \ldots, m\}$. The same concepts hold for a supremum (sup) constraint since the constraint is equivalent to:

$$-w = \sup\{-y_j : j = 1, \ldots, m\}$$

Assume that the original mathematical program containing constraint (2.3) has its objective function and its other constraints all linear. Replacing (2.3) by (2.4) results in the inclusion of a lower linear objective function as a constraint and hence the formulation of the problem as linear bi-level programming. Supposing the objective is a maximisation problem, three cases are considered:

(1) when w has a (strictly) positive coefficient in the original objective function, the lower objective in (2.4) is redundant and is discarded. The problem is then solved as a linear programming;

(2) when w has a (strictly) negative coefficient in the original objective function, the lower objective in (2.4) is necessary in the formulation. The problem must be solved as a linear bi-level programming;

(3) when w has a zero coefficient in the original objective, it is sometimes possible to multiply it by a suitable coefficient and add it to the original constraint; thereby, we can solve the problem as a linear programming. However, most of the time this suitable coefficient is hard, and often impossible, to find. Therefore, in this case, it is usually solved as a linear bi-level programming.

The capability of linear bi-level programming to include infimum constraints allows important extensions such as the formulation of any piecewise linear function. In particular, the first case applies to concave functions and the second case applies to convex functions.

2.2.3 Methods for Linear Bi-level Decision Problems

A linear bi-level decision problem has the important property that at least one global optimal solution is attained at an extreme point of the constraint region (Zhang, Lu & Dillon 2007c). This result was first established by Candler and Townsley (1982) for a linear bi-level decision problem with no upper level constraints and with unique lower level solutions. Later Bard (1984b), Bialas and Karwan (1984b) proved this result under the assumption that the constraint region is bounded. Based on these results, there have been nearly two dozen algorithms proposed for solving linear bi-level decision problems (Bard & Moore 1990; Shi, Lu & Zhang 2005a; Lu, Shi, Zhang & Ruan 2007b; Shi, Lu & Zhang 2005b; Shi, Lu, Zhang & Zhou 2006; Li, Tian & Min 2006; White & Anandalingam 1993). These algorithms can be roughly classified into three categories: the vertex enumeration based approaches (Bard & Moore 1990; Shi *et al.* 2005a), which use the important characteristic that at least one global optimal solution is attained at an extreme point of the constraints set; the Kuhn-Tucker approaches (Lu *et al.* 2007b; Shi *et al.* 2005b; Shi *et al.* 2006) in which a bi-level decision problem is transferred into a single level problem that solves the leader's problem while including the follower's optimality conditions as extra constraints; and the heuristics (Li *et al.* 2006; White & Anandalingam 1993), which are known as global optimisation techniques based on convergence analysis.

The Kth-Best method proposed by Bialas and Karwan (1984a) is one vertex enumeration approach. The method first ranks all extreme points by the upper level optimisation problem, then starts from the first point and checks whether it is also an optimal solution to the follower or not. If the first point is not the Stackelberg solution, the procedure continues to examine the next best solution to the leader and so on.

The Kuhn-Tucker method is used by Bialas and Karwan (1984a) in their parametric complementary pivot algorithm. Bard et al. (1990) replaces the complementarity constraint (complementary slackness condition) with a separable representation and applies a general Branch-and-Bound algorithm. Bard (1983) formulates a two-level programming problem as an equivalent semi-infinite problem and develops his grid search algorithm through a parametric linear program technique. Unlu (1987) proposed an algorithm based on bi-criteria programming by using the result of Bard (1983).

Despite the remarkable success with which Kth-Best and Kuhn-Tucker approaches have been applied to linear bi-level decision problems, they cannot, however, handle linear bi-level decision problems well when the constraint functions at the upper-level are of arbitrary linear forms (Shi *et al.* 2005b). Shi, Lu, and Zhang (2005c) extended the definition of linear bi-level solution by adding the constraints from the upper level to the follower's feasible set. Based on this definition, they also updated the Kuhn-Tucker theory for linear bi-level decision problems (Shi *et al.* 2005b) and developed the extended Kth-Best approach (Shi *et al.* 2005a) together with the extended Branch-and-Bound approach (Shi *et al.* 2006) to solve a wider class of linear bi-level decision problems.

Genetic algorithms and simulated annealing algorithms are two up-to-date heuristic directions towards linear bi-level decision problems. Mathieu et al. (1994) proposed the genetic algorithm-based bi-level programming algorithm by reproducing the leader's decision vector and solving the lower level linear problem to obtain the follower's decision vector. The fitness test involves only the leader's objective function and the reproductive plan is controlled by both the population size and selection strategy. Simulated annealing algorithms are derived from statistical mechanics with the aim of finding near optimal solutions to large-scale problems (Bard 1998). Anandalingam et al. (1992) developed a simulated annealing based bi-level programming algorithm for linear bi-level decision problems. This algorithm makes use of the fact that for a given leader-controlled variable, the follower's rational reaction can be obtained by solving the lower level linear programming, which implies that only components of the leader-controlled variables need to be generated randomly. By introducing a probability of the replacement of the current point by a new point, it promises a globally optimal solution statistically. Meanwhile, an additional control parameter known as the temperature is applied to promote the convergence by lowering it in steps when virtually no change occurs any more.

2.3 Multi-objective Bi-level Programming

For real world cases, decision making often has multi-objective characteristics, which have been studied in single level decision making, but only a few studies have been conducted in bi-level decision making situations (Wen & Hsu 1991). In a bi-level decision model, the selection of a solution by the leader is also affected by the follower's optimal reactions. Therefore, a solution for the leader who has multiple objectives needs to consider both the solution of the leader's multiple objectives and the follower's decision.

For multi-objective bi-level programming, Shi and Xia (1997) have presented an interactive algorithm. It first sets goals for a leader's objectives, then obtains many solutions that are close enough to the goals (larger than some certain "satisfactoriness"). Fixing the preferences from the leader, the follower's responses will be obtained one by one. The final solution is obtained once the follower's choice is near enough to that of the leader. However, to set "satisfactoriness" is not a direct job: if it was too big, there would be no solution at all, while huge computation would be caused by too small a value.

Recently, an approximation Branch-and-Bound algorithm, which handles multiple objectives by the weighting method, has been proposed to solve multi-objective bi-level decision problems with fuzzy demands by Zhang and Lu et al. (2007b).

For one level multi-objective decision problems, there are generally two sets of methods (Ehrgott & Gandibleux 2003), exact solution methods (Sakawa 1993; Zadeh 1963b; Kuhn & Tucker 1951; Zadeh 1963a), such as scalarisation methods (Sakawa 1993; Zadeh 1963b) and goal programming (Sakawa 1993), and heuristic solution methods (Augusto, Rabeau, Dépincé & Bennis 2006; Gandibleux, Mezdaoui & Freville 1997; Murata & Ishibuchi 1995), such as genetic algorithm (Murata & Ishibuchi 1995; Augusto *et al.* 2006).

For scalarisation methods, several computational methods have been proposed by different characterising approaches to Pareto optimal solutions. Among many possible ways of scalarising, the weighting method, the constraint method, and the weighted maxmini method have been the most widely used.

The term "goal programming" was first put forward by A. Charnes and W. W. Cooper to deal with multi-objective linear programming problems in 1961. It is assumed that a decision maker can specify the goals or aspirating levels for the objective functions. Subsequent studies on goal programming approaches have been numerous. The key idea behind goal programming is to minimise the deviations from goals or aspiration levels set by the decision maker. Goal programming therefore, in most cases, seems to yield a satisfying solution rather than an optimising one. By introducing the auxiliary variables, the linear goal programming problem can be converted to an equivalent linear programming problem (Charnes & Cooper 1977; Hwang & Yoon 1981). Goal programming has been further developed by Lee (1972), Ignizio (1983; 1976), Charnes and Cooper (1977). Recent research on goal programming can be found in (Lu, Wu & Zhang 2007d; Saad 2005; Hu, Teng & Li 2007; Pramanik & Roy 2007; Li, Wu & Yang 2004).

Since Schaffer (1984) first applied the genetic algorithm for multi-objective programming, many researchers have made efforts in this direction (Murata & Ishibuchi 1995; Augusto *et al.* 2006). Schaffer proposed the vector evaluated genetic algorithm (1984) for finding Pareto optimal solutions of multi-objective decision problems. In his work, a population is divided into disjoint sub-populations that are governed by different objective functions. Although Schaffer reported some successful results, it seems that only extreme solutions can be found as the search directions are parallel to the axes of the objective space (Murata & Ishibuchi 1995). To avoid this problem, Tadahiko (1995) presented a new genetic algorithm where the selection procedure takes the weights attached to multiple objectives not as a constant but randomly specified for each selection, thus utilising various search directions. In this field, some researches have been dedicated to improve the computing efficiency for "real time control" problem. Augusto (2006) proposed a genetic algorithm with more efficiency than regular genetic algorithms by the fact that it replaces the worst individuals by the offspring from the better one while stabilising the population size.

2.4 Multi-leader Bi-level Programming

Although there exists extensive research on bi-level programming with a single leader, studies on multi-leader bi-level programming are relatively scarce (Nie 2007).

A multi-leader Stackelberg decision model is generally viewed as one that leads to chaos or economic warfare (Sherali 1984). Okuguchi (1976; 1978) and Furth (1979) have presented consistent extensions of Stackelberg's decision model to the multi-leader situation in a way that gives equilibrium rather than disequilibrium solutions. Okuguchi (1976; 1978) considers a leader-leader duopoly, and Furth (1979) extended this development to include prices. Both these models are based on the leader-firms using linear estimators or predictors to describe the reactive behaviour of the other firms. As pointed out by Furth (1979), these models are therefore not true leader-follower decision models. The leaders make no attempt to manipulate the outputs or prices of the followers by using true reaction curves. In contrast, in Hanif's model (1984), the leader-firms employ true follower reaction curves, and hence yield a true leader-follower extension to Stackelberg's model. Subsequent studies (Yu & Wang 2007; Ehrenmann 2004) mainly focus on seeking a Stackelberg-Nash-Cournot equilibrium.

2.5 Multi-follower Bi-level Programming

The original bi-level programming technique mainly deals with one leader and one follower decision problems. In real world applications, multiple followers, that is, multiple decision units at the low level may be involved. Thus, a leader's decision will be affected not only by those followers' individual reactions but also by the relationships among them. For each possible solution of the leader, those followers may have their different reactions. The multiple followers may or may not share their decision variables. They may have their individual objectives and constraints but work with others cooperatively, or may have their common objectives or common constraints.

For a bi-level programming with multiple followers, a framework has been established and a total of 9 sub problems are identified according to different levels of cooperation among follower-controlled variables, objectives, and constraints respectively (Lu *et al.* 2006).

In a multi-follower bi-level decision problem, the followers may share or partially share their decision variables in their objectives and constraints. However, there are eight different sub-cases within the cooperative situation which are determined by the relationships among the objectives and constraints of the followers (Lu *et al.* 2006).

(1) followers with shared decision variables have the same objectives and the same constraints;

(2) followers with shared decision variables have the same objectives but different

constraints;

(3) followers with shared decision variables have different objectives but the same constraints;

(4) followers with shared decision variables have different objectives and different constraints;

(5) followers with partially shared decision variables have the same objectives and the same constraints;

(6) followers with partially shared decision variables have the same objectives but different constraints;

(7) followers with partially shared decision variables have different objectives but the same constraints;

(8) followers with partially shared decision variables have different objectives and different constraints.

The approach to each sub problem was presented based on the extended Kuhn-Tucker theorem (Lu, Shi, Zhang & Dillon 2007a).

2.6 Fuzzy Bi-level Programming

Shih et al. (1996) and Lai (1996) first applied the fuzzy approach to bi-level programming, although the bi-level problems addressed do not involve fuzzy coefficients. Their approach is based on the idea that the follower optimises an objective function, taking the goal of the leader into consideration. Both the leader and the follower elicit membership functions of fuzzy goals for their objective functions, and in particular, the leader also specifies those of fuzzy goals for his or her decision variables. The follower solves a fuzzy programming problem with a constraint on a satisfactory degree of the leader. This method, however, might cause a final solution that is undesirable because of inconsistency between the fuzzy goals of the objective function and the decision variables (Sakawa, Nishizaki & Uemura 2000).

To overcome this problem, Sakawa et al. (2000) developed an interactive fuzzy approach by deriving a satisfactory solution and updating the satisfactory degrees of decision makers with considerations of overall satisfactory balance among all levels. They have used the λ-cut method to defuzzify fuzzy numbers:

Definition 2.6.1. (Zadeh 1965) The λ-cut of a fuzzy set $\tilde{A}$ is defined as an ordinary set A_λ such that:

$$A_\lambda = \{x|\mu_{\tilde{A}}(x) \geqslant \lambda\}, \lambda \in [0, 1]$$

If A_λ is a non-empty bounded closed interval, it can be denoted by:

$$A_\lambda = [A_\lambda^L, A_\lambda^R] \tag{2.5}$$

where A_λ^L and A_λ^R are the lower and upper bounds of the interval respectively.

In their method, the decision maker at the upper level first denotes the satisfactory degree for all fuzzy coefficients in question, thus transforming the fuzzy bi-level problem into a non-fuzzy one. Decision makers at both levels specify their own membership functions by goals that the objective functions should be substantially less than or equal to some values. As there exist infinite "choices" that meet the requirement of the satisfactory degree, it is reasonable for both decision makers to optimise their objectives within these "choices" by maximising the membership functions. Then, the leader specifies the minimum satisfactory level, and the follower solves the single level optimisation problem by adding the leader's specification as an extra constraint. To make an overall satisfactory balance between both levels, Sakawa applied the method suggested by Zimmermann (1978) and thus transformed the problem into a linear programming problem. Finally, if the solution, denoted by satisfactory degrees of the leader and follower, of this linear programming problem meets two requirements: first, the leader's satisfactory degree is not less than a minimum satisfactory level; second, the ratio of satisfactory degrees between both levels is not beyond the lower and upper bounds specified by the leader, the algorithm stops, and current solution is obtained as the final one. Otherwise the leader adjusts the minimum satisfactory level and recalculates the linear programming problem until the solution is within the limits.

The methods of both Shih et al. (1996) and Sakawa et al. (2000) are based on the assumption that decision makers from different levels can essentially cooperate with each other. For classical bi-level problems, such as the Stackelberg problem (Stackelberg 1952), which assumes that cooperation is inhibited among decision makers in different levels, further investigation is still to be carried out.

In our lab, an approximation approach has been developed (Zhang *et al.* 2007a; Zhang, Lu & Dillon 2006a; Zhang *et al.* 2007b; Zhang *et al.* 2007c; Lu, Wu & Zhang 2007c) based on the FMMLB framework building and models formatting (Lu *et al.*

2006; Shi *et al.* 2005b; Shi, Zhang & Lu 2005c; Shi *et al.* 2006; Shi *et al.* 2005a; Lu *et al.* 2007a). To compare to fuzzy numbers, the following ranking method has been used:

Definition 2.6.2. (Zhang, Wu, Remias & Lu 2003) For any n-dimensional fuzzy vectors $\tilde{a} = (\tilde{a_1}, \ldots, \tilde{a_n})$, $\tilde{b} = (\tilde{b_1}, \ldots, \tilde{b_n})$, $\tilde{a_i}, \tilde{b_i} \in F(R)$, under a certain satisfactory degree $\alpha \in [0, 1]$, we define

$$\tilde{a} \preceq_\alpha \tilde{b} \text{ iff } {a_i}^L_\lambda \leqslant {b_i}^L_\lambda \text{ and } {a_i}^R_\lambda \leqslant {b_i}^R_\lambda, \quad i = 1, 2, \cdots, n, \forall \lambda \in [\alpha, 1]$$

Definition 2.6.2 means, when comparing two fuzzy numbers, that all values with membership grades smaller than α are neglected. When two fuzzy numbers cannot be compared under a certain α by this ranking method, we can adjust α to a larger degree to achieve the comparison.

For a problem defined by Definition 1.1.2, the solution can be reached by solving the associated multi-objectives bi-level decision making problem (2.6) under different cut sets λ_j, $j = 0, 1, \ldots, n$. (Zhang *et al.* 2007a):

$$\begin{aligned} \min_{x \in X} (F(x, y))^L_{\lambda_i} &= {c_1}^L_{\lambda_i} x + {d_1}^L_{\lambda_i} y, \\ \min_{x \in X} (F(x, y))^R_{\lambda_i} &= {c_1}^R_{\lambda_i} x + {d_1}^R_{\lambda_i} y, \end{aligned} \tag{2.6a}$$

$$\begin{aligned} \text{subject to } & {A_1}^L_{\lambda_i} x + {B_1}^L_{\lambda_i} y \leqq {b_1}^L_{\lambda_i}, \\ & {A_1}^R_{\lambda_i} x + {B_1}^R_{\lambda_i} y \leqq {b_1}^R_{\lambda_i}, \end{aligned} \tag{2.6b}$$

$$\begin{aligned} \min_{y \in Y} (f(x, y))^L_{\lambda_i} &= {c_2}^L_{\lambda_i} x + {d_2}^L_{\lambda_i} y, \\ \min_{y \in Y} (f(x, y))^R_{\lambda_i} &= {c_2}^R_{\lambda_i} x + {d_2}^R_{\lambda_i} y, \end{aligned} \tag{2.6c}$$

$$\begin{aligned} \text{subject to } & {A_2}^L_{\lambda_i} x + {B_2}^L_{\lambda_i} y \leqq {b_2}^L_{\lambda_i}, \\ & {A_2}^R_{\lambda_i} x + {B_2}^R_{\lambda_i} y \leqq {b_2}^R_{\lambda_i}. \end{aligned} \tag{2.6d}$$

where $i = 0, 1, \ldots, n$.

The weighting method is then adopted to further transfer this defuzzified MOLB decision problem (2.6) into a linear bi-level decision problem, which can be solved by the extended Branch-and-Bound algorithm (Shi *et al.* 2006) or extended Kth-Best approach (Shi *et al.* 2005a). The final solution is reached when solutions under two adjacent cut sets are near enough. Effective as this approach is, it suffers from expensive calculation when handling large-sized problems.

2.7 Applications of Bi-level Programming Techniques

The investigation of bi-level decision problems is strongly motivated by real world applications, and bi-level programming techniques have been applied with remarkable success in different domains, such as transportation network design (Clegg, Smith, Xiang & Yarrow 2001), production planning (Lukac, Soric & Rosenzweig 2006) and logistics (Zhang & Lu 2007a).

Ben-Ayed et al. have applied bi-level formulations to the network design problem (1988) arising from transportation systems. In the accompanying formulation, a central planner controls investment costs at the system level, while operational costs depend on traffic flows, which are determined by the individual users' route selection. Because users are assumed to make decisions to maximise their peculiar utility functions, their choices do not necessarily coincide with the choices that are optimal for the system. Nevertheless, the central planner can influence the users' choices by improving some links, making them relatively more attractive than others. In deciding on these improvements, the central planner tries to influence the users' preferences in such a way that total costs are minimised. The partition of the control variables between the upper and lower levels naturally leads to a bi-level formulation (Bard 1998).

A fuzzy bi-level model has been built up by Feng and Wen to control traffic flow in a disaster area after an earthquake (2005). When a severe earthquake occurs, the roadway systems usually experience different degrees of damage, thus reducing the capacity of those roadways, and causing traffic congestion. How to maintain traffic functions reasonably to facilitate saving more lives will be the utmost mission task after quakes. The commander of the Emergency-Response Centre of government at county and city level (the upper level) aims at allowing traffic to go through the disaster areas as much as possible within the roadway's capacity, while the road users (located at the lower level) always choose the shortest route to actualise emergency rescues. To solve this decision problem, the bi-level technique has been used to provide an efficient traffic control strategy for recovery from chaos post-earthquake.

Recently, Ji and Shao (2006) formulated a bi-level programming model for a newsboy problem. The classical newsboy problem is to find the newspaper's order quantity so that it maximises the expected profit of the newsboy, which is addressed by most research on a single level system. Ji and Shao (2006) located the decision makers involved at different decision levels: the manufacturer is considered to be at the top level controlling the wholesale prices, and the retailers are followers at the lower level who decide the ordering quantities of newspaper. Both the manufacturer and retailers aim to maximise their own profits. By developing a hybrid intelligent algorithm, the classical newsboy problem with fuzzy demands and price discounts policy was solved under a two level framework.

Apart from the applications listed above, bi-level decision problems are frequent in many other real world cases, such as resources allocation (Onaland, Darmawan & Johnson 1995), network investigation (Hobbs, Metzler & Pang 2000), and engineering (Ferris & Tin-Loi 2001). These applications have been stimulating factors for the development of bi-level programming techniques.

2.8 Bi-level Decision Support Systems

A decision support system (DSS) is a system that supports technological and managerial decision making by assisting in the organisation of knowledge about semi-structured issues (Zhang *et al.* 2007b).

Since a bi-level programming is a NP-hard problem due to its non-convexity and non-differentiability (Pei, Tian & Huang 2006), it is almost impossible to calculate a solution without the help of a software system. DSSs have been developed for modelling decision situations involving more than one decision maker (Fang, Hipel, Kilgour & Peng 2003) or under multi-criteria (Mustajoki & Hämäläinen 2007), such as multi-objective DSSs (Wu, Lu & Zhang 2005) and group DSSs (Lu, Zhang, Ruan & Wu 2007e). However, few aforementioned DSSs fall into the category where decision makers are located hierarchically.

2.9 Particle Swarm Optimisation (PSO)

Particle swarm optimisation (PSO) is a heuristic algorithm proposed by James Kennedy and Russell Eberhart in 1995 (Kennedy & Eberhart 1995). It is one of the community-intelligent algorithms for searching a global solution, which comes from the study of a simple model of a bird community and bird behaviour simulation (Zhao & Gu 2006).

Inspired by the social behaviour of animals such as fish schooling and bird flocking, PSO is a kind of population-based algorithm. The population of PSO is called "swarm", and each individual in the swarm is called "particle". The similarity between PSO and other evolutionary algorithms lies in the fact that the individual in the community is moved to a good area according to its fitness to the environment. Unlike other evolutionary computation methods, however, each particle in PSO has an adaptable velocity (position change), according to which it moves in the search space (Parsopoulos & Vrahatis 2002). Moreover, each particle has a memory, remembering the best position it has ever visited in the search space (Eberhart & Kennedy 1995). Thus, its movement is an aggregated acceleration towards its best previously visited position and towards the best particle of a topological neighborhood.

Suppose current search space for PSO is $n-$dimensional, then the $i-$th particle of the swarm can be represented by a $n-$dimensional vector, $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T$. The velocity (position change) of this particle can thus be represented by another $n-$dimensional vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})^T$. The best previously visited position of the $i-$th particle is denoted as $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})^T$. Defining g as the index of the best particle in the swarm (i.e., the $g-$th particle is the best), and letting the superscripts denote the iteration number, the swarm is manipulated according to the following two equations (Eberhart, Simpson & Dobbins 1996):

$$\begin{aligned} v_{id}^{k+1} &= wv_{id}^{k} + cr_1^k(p_{id} - x_{id}^k) + cr_2^k(p_{gd}^k - x_{id}^k) \\ x_{id}^{k+1} &= x_{id}^k + v_{id}^{k+1} \end{aligned} \tag{2.7}$$

where $d = 1, \ldots, n$ denotes the $d-$dimensional vector, $i = 1, 2, \ldots, N$ denotes $i-$particle, N is the size of the swarm, w is the “inertia weight”, c is a positive constant, called “acceleration constant”, and r_1, r_2 are random numbers, uniformly distributed in $[0, 1]$, and $k = 1, 2, \ldots$ determines the iteration number.

To escape from local optimisations, “stretching” technique (Parsopoulos & Vrahatis 2002) can be used. The “stretching” on a objective functions $F(x, y)$ is defined by:

$$\begin{aligned} G(x, y^*) &= F(x, y) + \gamma_1 ||x - x^*||(sign(F(x, y^*) - F(x^*, y^*)) + 1) \\ H(x, y^*) &= G(x, y^*) + \gamma_2 \tfrac{sign(F(x,y^*) - F(x*,y^*)) + 1}{tanh(\mu(G(x,y^*) - G(x^*,y^*)))} \end{aligned} \tag{2.8}$$

where γ_1, γ_2, and μ are arbitrary chosen positive constant, and $sign(\cdot)$ defines the well known triple valued sign function.

$$sign(x) = \begin{cases} 1, & \text{if } x < 0; \\ 0, & \text{if } x = 0; \\ -1, & \text{if } x < 0. \end{cases}$$

As PSO requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed (Parsopoulos & Vrahatis 2002; Eberhart & Kennedy 1995), it has a good convergence performance and has been successfully applied in many fields such as neural network training (Zhang, Zhang, Lok & Lyu 2007e), integral programming (Kitayama & Yasuda 2006; Rudolph 1994), minimax problem (Luksan & J 2000), and multi-object optimisation (Ho1, Yang, Ni, Lo & Wong 2005).

2.10 Summary

In this chapter, we review the concepts, models, properties, and techniques of bi-level programming, linear bi-level programming, multi-objective bi-level programming, MFB programming, MLB programming, fuzzy bi-level programming and particle swarm optimisation techniques. This chapter also reviews the relationship between bi-level programming and other optimisation problems, complexity and optimality conditions of bi-level problems, applications of bi-level programming techniques, bi-level decision support systems, and the PSO method. Some of these research results have built up the foundation for this research in the following chapters.

3 PSO for Fuzzy Linear Bi-level Decision Making

This chapter addresses bi-level decision problems featured with fuzzy coefficients, linear objective functions and constraints. We call them FLB decision problems. Based on a fuzzy ranking method, we give a mathematical definition of FLB problems. Then applying the strategy of PSO method, a PSO-based algorithm is presented for solving FLB decision problems. Finally, some experiments are carried to analyse the parameter choosing.

3.1 A Model

In this thesis, R represents the set of all real numbers, R^n is a $n-$dimensional Euclidean space, $F(R)$ and $F^n(R)$ are the set of all finite fuzzy numbers and the set of all $n-$dimensional finite fuzzy numbers on R^n respectively. A finite fuzzy number is a fuzzy number whose 0-cut is an interval whose ends are finite numbers.

Definition 3.1.1. (Zhang *et al.* 2007a) A fuzzy linear bi-level (FLB) decision problem is defined as : For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, $F : X \times Y \to F(R)$, and $f : X \times Y \to F(R)$,

$$\begin{aligned} &\min_{x\in X} F(x,y) = \tilde{c}_1 x + \tilde{d}_1 y \\ &\text{subject to } \tilde{A}_1 x + \tilde{B}_1 y \preceq \tilde{b}_1 \\ &\qquad \min_{y\in Y} f(x,y) = \tilde{c}_2 x + \tilde{d}_2 y \\ &\qquad \text{subject to } \tilde{A}_2 x + \tilde{B}_2 y \preceq \tilde{b}_2 \end{aligned} \tag{3.1}$$

where $\tilde{c}_1, \tilde{c}_2 \in F^n(R)$, $\tilde{d}_1, \tilde{d}_2 \in F^m(R)$, $\tilde{b}_1 \in F^p(R)$, $\tilde{b}_2 \in F^q(R)$, $\tilde{A}_1 = (\tilde{a}_{ij})_{p\times n}$, $\tilde{a}_{ij} \in F(R)$, $\tilde{B}_1 = \left(\tilde{b}_{ij}\right)_{p\times m}$, $\tilde{b}_{ij} \in F(R)$, $\tilde{A}_2 = (\tilde{e}_{ij})_{q\times n}$, $\tilde{e}_{ij} \in F(R)$, $\tilde{B}_2 = (\tilde{s}_{ij})_{q\times m}$, $\tilde{s}_{ij} \in F(R)$, and $F(R)$ is the set of all finite fuzzy numbers.

3.2 A PSO-based Algorithm

In this section, we develop a PSO-based algorithm for a FLB problem defined by Definition 3.1.1. The reasons we choose the PSO method are based on the following considerations:

Since classical methods for the NP-hard bi-level problem are still inefficient and lack universality (Zhao & Gu 2006), artificial intelligence based methods offer additional possibilities. As one of evolutionary computation based methods, PSO can be used for single level optimisation problems by pushing every potential solution (particle) towards the best ones. Here we reasonably extend it towards two level situation. For a bi-level decision problem, we first apply the PSO technique on the leader's problem, then for each leader's particle fixed, we need to use the PSO technique again to find the optimal response from the follower.

There are many other evolutionary computation methods, such as genetical algorithms, which have been applied to solve crisp bi-level problem successfully. By these methods, only a small number of individuals keep their "identities" and offsprings are generated by the interaction in the group. Thus, to find a final optimal solution largely depends on the validity of the initial population, to guarantee which, the Simplex method can be used to fix the initial population within the constraint area. However, for optimisation problems involving with fuzzy coefficients, the Simplex method becomes invalid.

Unlike many other evolutionary optimisation techniques, a particle swarm system has memory, and knowledge of good solutions is retained by all particles. Individuals who fly past optima are tugged to return towards them. This speciality makes it possible to generate an initial swarm without having to worry about the fuzzy issues prematurely. Thus, our strategy of handling fuzzy coefficients, which is illustrated in the following paragraphs, can be integrated perfectly with the PSO technique.

For a problem defined by Definition 3.1.1, the majority of current researches apply the method that defuzzifies the fuzzy problem first by certain kind of method, then solves the crisp problem by crisp bi-level optimisation techniques. This method, however, will lost some information carried by the fuzzy coefficients in the defuzzifying process.

In this research, a different strategy is used, where optimisation techniques are applied directly on fuzzy problems. In the procedure of computation, the PSO method is used on a bi-level problem first without considering its fuzzy coefficients to generate a swarm, then the fuzzy issue will be handled while each particle is evaluated by comparing them. As each particle represents a crisp solution, we need to compare different objective function values which are fuzzy numbers under some certain solutions. Here we use the ranking method defined by Definition 2.6.2 to compare any two fuzzy numbers. This strategy fully considers the original information of the fuzzy coefficients, thus minimising the information loss. This is a different angle to solve fuzzy bi-level optimisation problems.

The notations used in subsequent paragraphs are explained in Table 3.1.

Table 3.1: The explanation of some notations for Algorithm 1

N	the number of candidate solutions (particles) by the leader within its swarm;
M	the number of candidate solutions (particles) by the follower within its swarm;
x_i	= $(x_{i1}, x_{i2}, \ldots, x_{in})^T$, $i = 1, \ldots, N$, the i^{th} candidate solution for the leader;
v_i	= $(v_{i1}, vi2, \ldots, v_{in})^T$, $i = 1, \ldots, N$, the velocity of x_i;
y_i	=$(y_{i1}, y_{i2}, \ldots, y_{im})^T$, the follower's choice for each x_i from the leader;
y_{ij}	= $(y_{ij1}, y_{ij2}, \ldots, y_{ijm})^T$, $j = 1, \ldots, M$, the j^{th} candidate solution by the follower for the choice x_i from the leader;
v_{ij}	$= (v_{ij1}, ...v_{ijm})^T$, $j = 1, \ldots, M$, the velocity of y_{ij};
p_i	$= (p_{i1}, p_{i2}, \ldots, p_{in})^T$, the best previously visited position of x_i;
p_{ij}	$= (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T$, the best previously visited position of y_{ij};
y_{pi}	$= (y_{pi1}, y_{pi2}, \ldots, y_{pim})^T$, the response from the follower for the choice p_i from the leader;
CS	$= (CS_1, CS_2, \ldots, CS_n)$, the recording vector to record if x_i is within constraint area;
g	the index of the best particle for the leader in the swarm;
k_l	current iteration number for the upper-level problem;
k_f	current iteration number for the lower-level problem;
$MaxK_l$	the predefined max iteration number for k_l;
$MaxK_f$	the predefined max iteration number for k_f.

Based on the PSO technique and the strategy for handling fuzzy coefficients, the algorithm is outlined in Figure 3.1.

First we initiate a swarm comprised by the leader-controlled variables (X_particles). For each particle (x_i) in the swarm, we generate the optimal response from the follower by solving the following problem:

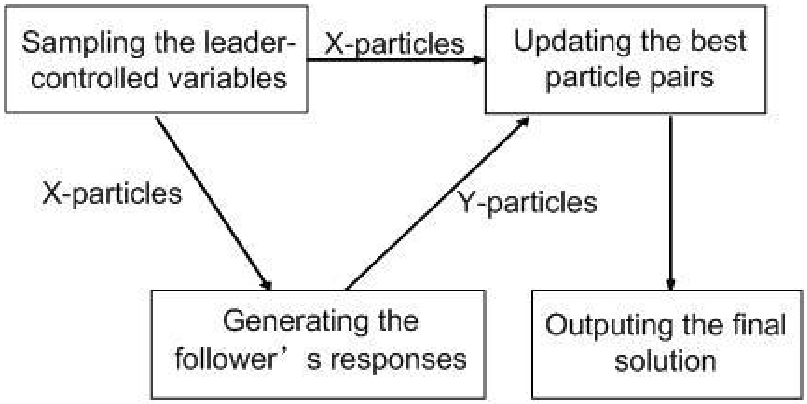

Figure 3.1: The outline of Algorithm 1

$$
\begin{aligned}
&\min_{y \in Y} f(x_i, y) = \tilde{c}_2 x_i + \tilde{d}_2 y \\
&\text{subject to } \tilde{A}_1 x_i + \tilde{B}_1 y \preceq \tilde{b}_1 \\
&\qquad\qquad \tilde{A}_2 x_i + \tilde{B}_2 y \preceq \tilde{b}_2
\end{aligned}
\tag{3.2}
$$

To solve Problem (3.2), we also need to generate a population (Y_particles), each of which has a velocity. From every particle pair (x_i, y_{ij}), a bunch of the follower's objective values can be generated, which are inevitably fuzzy numbers. These fuzzy objective values will be evaluated by comparing any two of them using Definition 2.6.2. Thus we can select the previously visited best positions for each y_particles and the best one among y_particles. The position y_{ij} and velocity v_{ij} for each in Y_particles will be updated using:

$$
\begin{aligned}
v_{ij}^{k_f+1} &= w v_{ij}^{k_f} + c r_1^{k_f} (p_{ij} - y_{ij}^{k_f}) + c r_2^{k_f} (y_i^{k_f} - y_{ij}^{k_f}) \\
y_{ij}^{k_f+1} &= y_{ij}^{k_f} + v_{ij}^{k_f+1}
\end{aligned}
\tag{3.3}
$$

Here, k_f is to record current loop. Once k_f is larger than some predefined value, y_i will be sent to the leader as the follower's response for x_i.

Above optimisation and computation procedure will also be applied to every particle pair (x_i, y_i) to update the position x_i and velocity v_i of every leader's particle:

$$
\begin{aligned}
v_i^{k_l+1} &= w v_i^{k_l} + c r_1^{k_l} (p_i - x_i^{k_l}) + c r_2^{k_l} (x_g^{k_l} - x_i^{k_l}) \\
x_i^{k_l+1} &= x_i^{k_l} + v_i^{k_l+1}
\end{aligned}
\tag{3.4}
$$

Once the iteration times k_l is large enough, current best particle pair (x_g, y_g) will be outputted as the final solution. This algorithm is specified in Algorithm 1.

Algorithm 1: A PSO-based algorithm for FLB decision problems

Input: the coefficients of Problem (3.1)
Output: (x_g, y_g)
Initialising: $k_l = 1$; $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})^T = (0, 0, \ldots, 0)^T$;
Sampling: $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T$; $v_i = (v_{i1}, v_{i2}, ...v_{in})^T$, $i = 1, \ldots, N$;
Generating the responses from the follower:
foreach x_i **do**
  $k_f = 1$; $p_{ij} = (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T = (0, 0, \ldots, 0)^T$;
  Sampling: $y_{ij} = (y_{ij1}, y_{ij2}, \ldots, y_{ijm})^T$; $v_{ij} = (v_{ij1}, v_{ij2}, ...v_{ijm})^T$, $j = 1, \ldots, M$;
1 CS_i = false;
  if $\tilde{A}_1 x + \tilde{B}_1 y \preceq \tilde{b}_1$ ***and*** $\tilde{A}_2 x_i + \tilde{B}_2 y \preceq \tilde{b}_2$ **then**
    CS_i = true;
  end
  if *(*$p_{ij} = (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T = (0, 0, \ldots, 0)^T$*)* ***or*** *(*$f(x_i, y_{ij}) \preceq f(x_i, p_{ij})$*)* **then**
    $p_{ij} = (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T = (y_{ij1}, y_{ij2}, \ldots, y_{ijm})^T$;
  end
  Searching the best response y_i from p_{ij}, $j = 1, 2, \ldots, M$; Updating velocities and positions using Equation (3.3); $k_f = k_f + 1$;
  if $k_f \geqslant$ *MaxK* **then**
    Goto 2;
  else
    Goto 1;
  end
end
2 **if** CS_i = *true* **then**
  if *(*$p_i = (p_{i1}, \ldots, p_{in})^T = (0, \ldots, 0)^T$*)* ***or*** *(*$F(x_i, y_i) \leqslant F(p_i, y_{pi})$*)* **then**
    $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})^T = (x_{i1}, x_{i2}, \ldots, x_{in})^T$;
    $y_{pi} = (y_{pi1}, y_{pi2}, \ldots, y_{pim})^T = (y_{i1}, y_{i2}, \ldots, y_{im})^T$;
  end
end
Searching (x_g, y_g) from p_i, and y_{pi}, $i = 1, \ldots, N$; Updating x_i and v_i using Equation (3.4); $k_l = k_l + 1$;
if $k_l \geqslant$ *MaxK* **then**
  Stop;
else
  Goto 2;
end

3.3 Experiments and Analysis

In this section, two numerical examples are employed to test this PSO-based algorithm. Based on the experiments, we discuss the choice of the parameters.

The two numerical examples are listed as follows:

Example1:

$$
\begin{aligned}
&\max_{x\in X} F(x,y) = \tilde{6}x + \tilde{3}y \\
&\text{subject to } -\tilde{1}x + \tilde{3}y \preceq \tilde{21} \\
&\max_{y\in Y} f(x,y) = -\tilde{3}x + \tilde{6}y \\
&\text{subject to } \tilde{1}x + \tilde{3}y \preceq \tilde{27}
\end{aligned}
\tag{3.5}
$$

Example2:

$$
\begin{aligned}
&\max_{x\in X} F(x,y) = \tilde{6}x_1 + \tilde{3}x_2 - \tilde{3}x_3 + \tilde{6}y_1 - \tilde{1}y_2 \\
&\text{subject to } \tilde{1}x_1 - \tilde{1}x_2 + \tilde{3}x_3 + \tilde{1}y_1 + \tilde{3}y_2 \preceq \tilde{21} \\
&\tilde{21}x_1 + \tilde{27}x_2 + \tilde{1}x_3 - \tilde{1}y_1 + \tilde{3}y_2 \preceq \tilde{27} \\
&\max_{y\in Y} f(x,y) = \tilde{3}x_1 + \tilde{1}x_2 + \tilde{3}x_3 + \tilde{21}y_1 + \tilde{27}y_2 \\
&\text{subject to } -\tilde{3}x_1 + \tilde{6}x_2 - \tilde{1}x_3 + \tilde{3}y_1 + \tilde{1}y_2 \preceq \tilde{21} \\
&\tilde{3}x_1 + \tilde{21}x_2 + \tilde{27}x_3 + \tilde{1}y_1 - \tilde{1}y_2 \preceq \tilde{27}
\end{aligned}
\tag{3.6}
$$

The membership functions of the coefficients in these examples are as follows:

$$
\mu_{\tilde{6}}(x) = \begin{cases} 0, & x < 5 \\ \frac{x^2-25}{11}, & 5 \leqq x < 8 \\ 1, & x = 6 \\ \frac{64-x^2}{28}, & 6 < x \leqslant 8 \\ 0, & x > 8 \end{cases}, \quad \mu_{\tilde{3}}(x) = \begin{cases} 0, & x < 2 \\ \frac{x^2-4}{5}, & 2 \leqq x < 3 \\ 1, & x = 3 \\ \frac{25-x^2}{16}, & 3 < x \leqslant 5 \\ 0, & x > 5 \end{cases},
$$

$$
\mu_{\tilde{-1}}(x) = \begin{cases} 0, & x < -2 \\ \frac{4-x^2}{3}, & -2 \leqq x < -1 \\ 1, & x = -1 \\ \frac{x^2-0.25}{0.75}, & -1 < x \leqslant -0.5 \\ 0, & x > -0.5 \end{cases}, \quad \mu_{\tilde{-3}}(x) = \begin{cases} 0, & x < -4 \\ \frac{16-x^2}{7}, & -4 \leqq x < -3 \\ 1, & x = -3 \\ \frac{x^2-1}{8}, & -3 < x \leqslant -1 \\ 0, & x > -1 \end{cases},
$$

$$
\mu_{\tilde{1}}(x) = \begin{cases} 0, & x < 0.5 \\ \frac{x^2-0.25}{0.75}, & 0.5 \leqq x < 1 \\ 1, & x = 1 \\ \frac{4-x^2}{3}, & 1 < x \leqslant 2 \\ 0, & x > 2 \end{cases}, \quad \mu_{\tilde{3}}(x) = \begin{cases} 0, & x < 2 \\ \frac{x^2-4}{5}, & 2 \leqq x < 3 \\ 1, & x = 3 \\ \frac{25-x^2}{16}, & 3 < x \leqslant 5 \\ 0, & x > 5 \end{cases},
$$

$$\mu_{\tilde{21}}(x) = \begin{cases} 0, & x < 19 \\ \frac{x^2-361}{80}, & 19 \leqq x < 21 \\ 1, & x = 21 \\ \frac{625-x^2}{184}, & 21 < x \leqslant 25 \\ 0, & x > 25 \end{cases}, \mu_{\tilde{27}}(x) = \begin{cases} 0, & x < 25 \\ \frac{x^2-625}{104}, & 25 \leqq x < 27 \\ 1, & x = 27 \\ \frac{961-x^2}{232}, & 27 < x \leqslant 31 \\ 0, & x > 31 \end{cases}.$$

These examples are run by the PSO-based algorithm proposed in this chapter, which was implemented by Visual Basic 6.0, and tested on a desktop computer with CPU Pentium 4 2.8GHz, RAM 1G, Windows XP.

In the experiments, the inertia weight w is initially set as 1.2, and is gradually declined towards 0, and the population size is set as 20. Now we adjust the coefficients of c_1 and c_2 from 0.5 to 2 respectively. Under every pair of specific c_1 and c_2, the two examples are run in the PSO-based algorithm by five times, and different solutions have been obtained. To evaluate the performance, we compare the solutions obtained from the PSO-based algorithm with those from the classical fuzzy bi-level algorithms of the extended Branch-and-Bound algorithm (Zhang *et al.* 2006a) and the extended Kth-Best algorithm (Zhang, Lu & Dillon 2006b). Table 3.2 and Table 3.3 list the experiment result for Example 1 and Example 2 respectively. In Table 3.2, ($\triangle x^*$, $\triangle y^*$) represents the average solution difference between the PSO-based algorithm and the classical algorithms for every decision vector. The column of "$\triangle$" sums up the average difference by every decision vector for every c_1, c_2 pair. In the column of "Time", the average running time are listed which is calculated by seconds. The symbols can be explained the same way in Table 3.3 for Example 2.

In Table 3.2, we can see that, there is no obvious diversion among the solutions and the running time is also quite stable. In Table 3.3, where a more complex example is involved, the most optimal solution occurs at the point where $c_1 = 0.5$, $c_2 = 2$ with the least average solution fluctuation and least average computation time. At other points where $c_1 = 0.5$, $c_2 = 1$; $c_1 = 1$, $c_2 = 1$; and $c_1 = 2$, $c_2 = 1$; this PSO-based algorithm runs efficiently and effectively with stable performance.

Above experiments show this PSO-based algorithm can obtain stable solutions which are very close to those from the classical methods. Thus, we can come to the conclusion that the PSO-based algorithm proposed in this chapter can solve FLB problems quite correctly and effectively by exploring veracious solutions.

What we can not ignore is that the computation time of the PSO-based algorithm is still much longer than the classical algorithms. This inefficiency comes from the nature of heuristic strategy which simulates the optimisation process while the classical

Table 3.2: Summary of the running solutions for Example 1

(c_1, c_2)	$(\triangle x^*, \triangle y^*)$	$\triangle$	Time
(0.5, 0.5)	(0.00114, 0)	0.00114	54.6
(0.5, 1)	(0, 0)	0	52.6
(0.5, 1.5)	(0, 0)	0	51
(0.5, 2)	(0, 0)	0	50.8
(1, 0.5)	(0.00008, 0)	0.00008	59
(1, 1)	(0, 0)	0	54
(1, 1.5)	(0, 0)	0	52.8
(1, 2)	(0.00002, 0)	0.00002	51.8
(1.5, 0.5)	(0.0001, 0)	0.0001	56.4
(1.5, 1)	(0, 0)	0	53.8
(1.5, 1.5)	(0, 0)	0	52.6
(1.5, 2)	(0.00006, 0)	0.00006	54.8
(2, 0.5)	(0, 0)	0	57.6
(2, 1)	(0, 0)	0	53.4
(2, 1.5)	(0, 0)	0	52
(2, 2)	(0.00004, 0)	0.00004	50.4

Table 3.3: Summary of the running solutions for Example 2

(c_1, c_2)	$(\triangle x_1^*, \triangle x_2^*, \triangle x_3^*, \triangle y_1^*, \triangle y_2^*)$	$\triangle$	Time
(0.5, 0.5)	(0.13028, 0.04322, 0.0006, 0.09532, 0)	0.26942	113
(0.5, 1)	(0.00402, 0, 0, 0.0008, 0)	0.00482	85.2
(0.5, 1.5)	(0.22934, 0, 0, 0.04672, 0.00214)	0.2782	92.6
(0.5, 2)	(0.0034, 0, 0, 0.00068, 0)	0.00408	84.8
(1, 0.5)	(0.01366, 0, 0, 0.00274, 0)	0.0164	106.4
(1, 1)	(0.00416, 0, 0, 0.00084, 0)	0.005	88.8
(1, 1.5)	(0.26908, 0, 0, 0.05382, 0)	0.3229	88.8
(1, 2)	(0.26908, 0, 0.07884, 0.29356, 0.61704)	1.25852	90
(1.5, 0.5)	(0.26908, 0.13478, 0, 0.38254, 0.27306)	1.05946	103.6
(1.5, 1)	(0.00354, 0, 0, 0.0007, 0)	0.00424	91.2
(1.5, 1.5)	(0.26908, 0, 0, 0.05382, 0)	0.3229	92.6
(1.5, 2)	(0.26908, 0, 0, 0.05744, 0.00702)	0.33354	90
(2, 0.5)	(0.26908, 0, 0, 0.3405, 0.71414)	1.32372	106.2
(2, 1)	(0.00426, 0, 0, 0.00086, 0)	0.00512	92.4
(2,1.5)	(0.26908, 0, 0, 0.13452, 0.20054)	0.60414	87.4
(2, 2)	(0.26908, 0.03682, 0, 0.11272, 0)	0.41862	88

methods use the mathematical properties to directly get the solutions. However, using the mathematical properties sometimes can not reach a solution when these properties can not be satisfied, while heuristic methods are capable of overpassing these complex property verification to generate a reasonable solution at all. In many situations, this reasonable solution is very helpful for a decision maker when making a plan. So the importance of heuristic method should not be ignored as it explores new direction for bi-level optimisation.

3.4 Summary

This chapter studies FLB problems where fuzzy coefficients in their objective functions or constraints are represented in any form of membership functions. By introducing a ranking method based on cut sets, a new concept of optimal solution for FLB problems is defined. A PSO-based algorithm is proposed in this chapter for solving FLB problems. The experiments reveal that the PSO-based algorithm is effective to solve FLB decision problems. This PSO-based algorithm is one of the computation kernels in a fuzzy bi-level decision support system developed to assist decision makers to solve realistic FLB problems. This system will be described in Chapter 8 in detail.

4 Goal Programming for Fuzzy Multi-objective Linear Bi-level Decision Making

This chapter addresses bi-level decision problems featured by multiple objectives, fuzzy coefficients, linear objective functions and constraints. We call them FMOLB decision problems. First, using a fuzzy ranking method, we give a mathematical definition for FMOLB decision problems. Then, based on the definition of a distance measure between two fuzzy vectors using λ-cut, a fuzzy linear bi-level goal (FLBG) model is formatted and related theorems are proved. Next, a λ-cut and goal-programming-based algorithm is presented for solving FMOLB decision problems. Finally, a case study on a newsboy problem is adopted to illustrate the application and executing procedure of the algorithm and experiments are carried out to discuss and analyse the performance of this algorithm.

4.1 Definitions, Models and Theorems

The model of a general bi-level decision problem with multiple objectives for both the leader and follower was given in (Shi & Xia 1997). It is re-formulated in this chapter as:

For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, a multi-objective bi-level (MOB) model is:

$$\min_{x \in X} F(x, y) \tag{4.1a}$$

$$\text{subject to } G(x, y) \leqslant 0 \tag{4.1b}$$

$$\min_{y \in Y} f(x, y) \tag{4.1c}$$

$$\text{subject to } g(x, y) \leqslant 0 \tag{4.1d}$$

where $F : R^n \times R^m \to R^s$, $G : R^n \times R^m \to R^p$, $f : R^n \times R^m \to R^t$, and $g : R^n \times R^m \to R^q$.

Associated with the MOB problem (4.1), some definitions are listed below:

Definition 4.1.1.

- Constraint region of the MOB (4.1):

$$S \triangleq \{(x, y) : x \in X, y \in Y, G(x, y) \leqslant 0, g(x, y) \leqslant 0\}$$

  It refers to all possible combination of choices that the leader and follower may make.

- Projection of S onto the leader's decision space:

$$S(X) \triangleq \{x \in X : \exists y \in Y, G(x, y) \leqslant 0, g(x, y) \leqslant 0\}$$

- Feasible set for the follower $\forall x \in S(X)$:

$$S(x) \triangleq \{y \in Y : (x, y) \in S\}$$

- The follower's rational reaction set for $x \in S(X)$:

$$P(x) \triangleq \{y \in Y : y \in argmin[f(x, \hat{y}) : \hat{y} \in S(x)]\}$$

  where $argmin[f(x, \hat{y}) : \hat{y} \in S(x)] = \{y \in S(x) : f(x, y) \leqslant f(x, \hat{y}), \hat{y} \in S(x)\}$.

  The follower observes the leader's action and reacts by selecting y from his or her feasible set to minimise his or her objective function.

- Inducible region:

$$IR \triangleq \{(x, y) : (x, y) \in S, y \in P(x)\}$$

  which represents the set over which a leader may optimise his or her objectives.

To ensure that (4.1) is well posed, it is assumed that S is non-empty and compact, and that for all decisions taken by the leader, the follower has some room to respond, i.e., $P(x) \neq \emptyset$.

Thus, in terms of the above notations, an MOB problem can be written as:

$$\min\{F(x, y) : (x, y) \in IR\} \tag{4.2}$$

Based on the fuzzy ranking method in Definition 2.6.2, an FMOLB decision problem is defined as:

Definition 4.1.2. For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, $F : X \times Y \to F^s(R)$, and $f : X \times Y \to F^t(R)$,

$$\min_{x \in X} F(x, y) = (\tilde{c}_{11}x + \tilde{d}_{11}y, \ldots, \tilde{c}_{s1}x + \tilde{d}_{s1}y)^T \quad (4.3a)$$

$$\text{subject to } \tilde{A}_1 x + \tilde{B}_1 y \preceq_\alpha \tilde{b}_1 \quad (4.3b)$$

$$\min_{y \in Y} f(x, y) = (\tilde{c}_{12}x + \tilde{d}_{12}y, \ldots, \tilde{c}_{t2}x + \tilde{d}_{t2}y)^T \quad (4.3c)$$

$$\text{subject to } \tilde{A}_2 x + \tilde{B}_2 y \preceq_\alpha \tilde{b}_2 \quad (4.3d)$$

where $\tilde{c}_{h1}$, $\tilde{c}_{i2} \in F^n(R)$, $\tilde{d}_{h1}$, $\tilde{d}_{i2} \in F^m(R)$, $h = 1, 2, \ldots, s$, $i = 1, 2, \ldots, t$, $\tilde{b}_1 \in F^p(R)$, $\tilde{b}_2 \in F^q(R)$, $\tilde{A}_1 = (\tilde{a}_{ij})_{p \times n}$, $\tilde{B}_1 = \left(\tilde{b}_{ij}\right)_{p \times m}$, $\tilde{A}_2 = (\tilde{e}_{ij})_{q \times n}$, $\tilde{B}_2 = (\tilde{s}_{ij})_{q \times m}$, $\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{e}_{ij}, \tilde{s}_{ij} \in F(R)$.

To build an FLBG model, a distance measure between two fuzzy vectors is needed. There are many important measures to compare two fuzzy numbers, such as Hausdorff distance (Chaudhuri & Rosenfeld 1999), Hamming distance (Diamond & Kloeden 1994), Euclidean distance (Diamond & Kloeden 1994), and the maximum distance (Kacprzyk 1997). In this chapter, a certain number of λ-cuts will be used to approximate a fuzzy number. A final solution is considered to be reached when solutions under two adjacent λ-cuts are near enough. To help implement this strategy, a new distance measure between two fuzzy vectors by using $\lambda-$cuts is defined below:

Definition 4.1.3. Let $\tilde{a} = (\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n)$, $\tilde{b} = (\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_n)$ be $n-$dimensional fuzzy vectors, $\Phi = \{\alpha \leqslant \lambda_0 < \lambda_1 < \cdots < \lambda_l \leqslant 1\}$ be a division of $[\alpha, 1]$, the distance between $\tilde{a}$ and $\tilde{b}$ under ϕ is defined as:

$$D(\tilde{a}, \tilde{b}) \triangleq \frac{1}{l+1} \sum_{i=1}^{n} \sum_{j=0}^{l} \{|a^L_{i\lambda_j} - b^L_{i\lambda_j}| + |a^R_{i\lambda_j} - b^R_{i\lambda_j}|\} \quad (4.4)$$

where α is a predefined satisfactory degree.

In this fuzzy distance definition, a satisfactory degree α is used to give more flexibility to compare fuzzy vectors. It is possible that two fuzzy vectors might not be compared by Definition 4.1.3. For example, when we compare two fuzzy vectors $\tilde{a}$ and $\tilde{b}$, if some of the left λ-cuts of $\tilde{a}$ are less than those of $\tilde{b}$, while some right λ-cuts of $\tilde{a}$ are larger than those of $\tilde{b}$, there is no ranking relation between $\tilde{a}$ and $\tilde{b}$.

To solve this problem, we can enhance the aspiration levels of the attributes, i.e., we can adjust the satisfactory degree α to a point where all incomparable parts are discarded. It can be understood as a risk taken by a decision maker who neglects all values with the possibility of occurrence smaller than α. In such a situation, a solution is supposed to be reached under this aspiration level. So, normally, we take the same α for both objectives and constraints in one bi-level problem.

Lemma 4.1.1. *For any n-dimensional fuzzy vectors $\tilde{a}, \tilde{b}, \tilde{c}$, fuzzy distance D defined above satisfies the following properties:*

(1) $D(\tilde{a}, \tilde{b}) = 0$, *if* $\tilde{a}_i = \tilde{b}_i$, $i = 1, 2, \ldots, n$

(2) $D(\tilde{a}, \tilde{b}) = D(\tilde{b}, \tilde{a})$

(3) $D(\tilde{a}, \tilde{b}) \leqslant D(\tilde{a}, \tilde{c}) + D(\tilde{c}, \tilde{b})$. ■

Goals set for the objectives of a leader ($\tilde{g}_L$) and a follower ($\tilde{g}_F$) in (4.3) are defined as:

$$\tilde{g_L} = (\tilde{g}_{L1}, \tilde{g}_{L2}, \ldots, \tilde{g}_{Ls})^T, \tag{4.5a}$$

$$\tilde{g_F} = (\tilde{g}_{F1}, \tilde{g}_{F2}, \ldots, \tilde{g}_{Ft})^T, \tag{4.5b}$$

where $\tilde{g}_{Li}$, $i = 1, \ldots, s$, $\tilde{g}_{Fj}$, $j = 1, \ldots, t$ are fuzzy numbers with membership functions of $\mu_{\tilde{g}_{Li}}, \mu_{\tilde{g}_{Fj}}$.

Our concern is to make the objectives of both a leader and the follower as near to their goals as possible. Using the distance measure defined in (4.4), we format an FLBG problem as:

For $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, $F : X \times Y \to F^s(R)$, and $f : X \times Y \to F^t(R)$,

$$\min_{x \in X} D(F(x, y), \tilde{g_L}) \tag{4.6a}$$

$$\text{subject to } \tilde{A}_1 x + \tilde{B}_1 y \preceq_\alpha \tilde{b}_1 \tag{4.6b}$$

$$\min_{y \in Y} D(f(x, y), \tilde{g_F}) \tag{4.6c}$$

$$\text{subject to } \tilde{A}_2 x + \tilde{B}_2 y \preceq_\alpha \tilde{b}_2 \tag{4.6d}$$

where $\tilde{A}_1 = (\tilde{a}_{ij})_{p \times n}$, $\tilde{B}_1 = \left(\tilde{b}_{ij}\right)_{p \times m}$, $\tilde{A}_2 = (\tilde{e}_{ij})_{q \times n}$, $\tilde{B}_2 = (\tilde{s}_{ij})_{q \times m}$, $\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{e}_{ij}, \tilde{s}_{ij} \in F(R)$, and α is a predefined satisfactory degree.

From Definitions 2.6.2, 4.1.2, and 4.1.3, we transfer problem (4.6) into:

$$\min_{x\in X} \triangleq \tfrac{1}{l+1}\sum_{h=1}^{s}\sum_{j=0}^{l}\{|c_{h1\lambda_j}^{L}x + d_{h1\lambda_j}^{L}y - g_{Lh\lambda_j}^{L}| + |c_{h1\lambda_j}^{R}x + d_{h1\lambda_j}^{R}y - g_{Lh\lambda_j}^{R}|\}, \tag{4.7a}$$

$$\begin{aligned} \text{subject to } & A_{1\lambda_j}^{L}x + B_{1\lambda_j}^{L}y \leqslant b_{1\lambda_j}^{L}, \\ & A_{1\lambda_j}^{R}x + B_{1\lambda_j}^{R}y \leqslant b_{1\lambda_j}^{R}, \\ & j = 0, 1, \ldots, l, \end{aligned} \tag{4.7b}$$

$$\min_{y\in Y} \triangleq \tfrac{1}{l+1}\sum_{i=1}^{t}\sum_{j=0}^{l}\{|c_{i2\lambda_j}^{L}x + d_{i2\lambda_j}^{L}y - g_{Fi\lambda_j}^{L}| + |c_{i2\lambda_j}^{R}x + d_{i2\lambda_j}^{R}y - g_{Fi\lambda_j}^{R}|\}, \tag{4.7c}$$

$$\begin{aligned} \text{subject to } & A_{2\lambda_j}^{L}x + B_{2\lambda_j}^{L}y \leqslant b_{2\lambda_j}^{L}, \\ & A_{2\lambda_j}^{R}x + B_{2\lambda_j}^{R}y \leqslant b_{2\lambda_j}^{R} \\ & j = 0, 1, \ldots, l, \end{aligned} \tag{4.7d}$$

where $\Phi = \{\alpha \leqslant \lambda_0 < \lambda_1 < \cdots < \lambda_l \leqslant 1\}$ is a division of $[\alpha, 1]$.

For a clear understanding of the idea adopted, define:

$$\begin{aligned}
v_{h1}^{L-} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{h1\lambda_j}^{L}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}| - (\sum_{j=0}^{l} c_{h1\lambda_j}^{L}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L})] \\
v_{h1}^{L+} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{h1\lambda_j}^{L}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}| + (\sum_{j=0}^{l} c_{h1\lambda_j}^{L}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L})] \\
v_{h1}^{R-} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{h1\lambda_j}^{R}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}| - (\sum_{j=0}^{l} c_{h1\lambda_j}^{R}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R})] \\
v_{h1}^{R+} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{h1\lambda_j}^{R}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}| + (\sum_{j=0}^{l} c_{h1\lambda_j}^{R}x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R})] \\
h &= 1, 2 \ldots, s, \\
v_{i2}^{L-} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{i2\lambda_j}^{L}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{L}| - (\sum_{j=0}^{l} c_{i2\lambda_j}^{L}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{L})] \\
v_{i2}^{L+} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{i2\lambda_j}^{L}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{L}| + (\sum_{j=0}^{l} c_{i2\lambda_j}^{L}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{L})] \\
v_{i2}^{R-} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{i2\lambda_j}^{R}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{R}| - (\sum_{j=0}^{l} c_{i2\lambda_j}^{R}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{R})] \\
v_{i2}^{R+} &= \tfrac{1}{2}[|\sum_{j=0}^{l} c_{i2\lambda_j}^{R}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{R}| + (\sum_{j=0}^{l} c_{i2\lambda_j}^{R}x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L}y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{R})] \\
i &= 1, 2 \ldots, t,
\end{aligned} \tag{4.8}$$

where v_{h1}^{L-} and v_{h1}^{L+} are deviational variables representing the under-achievement and over-achievement of the h^{th} goal for a leader under the left λ-cut, v_{h1}^{R-} and v_{h1}^{R+} are deviational variables representing the under-achievement and over-achievement of the

h^{th} goal for a leader under the right λ-cut, v_{i2}^{L-}, v_{i2}^{L+}, v_{i2}^{R-} and v_{i2}^{R+} are for a follower respectively.

Associated with the linear bi-level problem (4.7), we now consider the following bi-level problem:

For $(v_{11}^{L-}, v_{11}^{L+}, v_{11}^{R-}, v_{11}^{R+}, \ldots, v_{s1}^{L-}, v_{s1}^{L+}, v_{s1}^{R-}, v_{s1}^{R+}) \in R^{4s}$, $X' \subseteq X \times R^{4s}$, $(v_{12}^{L-}, v_{12}^{L+}, v_{12}^{R-}, v_{12}^{R+}, \ldots, v_{t2}^{L-}, v_{t2}^{L+}, v_{t2}^{R-}, v_{t2}^{R+}) \in R^{4t}$, $Y' \subseteq Y \times R^{4t}$, let $x = (x_1, \cdots, x_n) \in X$, $x' = (x_1, \cdots, x_n, v_{11}^{L-}, v_{11}^{L+}, v_{11}^{R-}, v_{11}^{R+}, \ldots, v_{s1}^{L-}, v_{s1}^{L+}, v_{s1}^{R-}, v_{s1}^{R+}) \in X'$, $y = (y_1, \cdots, y_m) \in Y$, $y' = (y_1, \cdots, y_m, v_{12}^{L-}, v_{12}^{L+}, v_{12}^{R-}, v_{12}^{R+}, \ldots, v_{t2}^{L-}, v_{t2}^{L+}, v_{t2}^{R-}, v_{t2}^{R+}) \in Y'$, and $v_1, v_2, : X' \times Y' \to R$.

$$\min_{x' \in X'} v_1 = \sum_{h=1}^{s} (v_{h1}^{L-} + v_{h1}^{L+} + v_{h1}^{R-} + v_{h1}^{R+}) \tag{4.9a}$$

$$\begin{aligned}
\text{subject to } & \sum_{j=0}^{l} c_{h1\lambda_j}^{L} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y + v_{h1}^{L-} - v_{h1}^{L+} = \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}, \\
& \sum_{j=0}^{l} c_{h1\lambda_j}^{R} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y + v_{h1}^{R-} - v_{h1}^{R+} = \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}, \\
& v_{h1}^{L-}, v_{h1}^{L+}, v_{h1}^{R-}, v_{h1}^{R+} \geqslant 0, \\
& v_{h1}^{L-} \cdot v_{h1}^{L+} = 0, v_{h1}^{R-} \cdot v_{h1}^{R+} = 0 \qquad (4.9\text{b}) \\
& h = 1, 2, \ldots, s, \\
& A_{1}{}_{\lambda_j}^{L} x + B_{1}{}_{\lambda_j}^{L} y \leqslant b_{1}{}_{\lambda_j}^{L}, \\
& A_{1}{}_{\lambda_j}^{R} x + B_{1}{}_{\lambda_j}^{R} y \leqslant b_{1}{}_{\lambda_j}^{R}, \\
& j = 0, 1, \ldots, l,
\end{aligned}$$

$$\min_{y' \in Y'} v_2 = \sum_{i=1}^{t} (v_{i2}^{L-} + v_{i2}^{L+} + v_{i2}^{R-} + v_{i2}^{R+}) \tag{4.9c}$$

$$\begin{aligned}
\text{subject to } & \sum_{j=0}^{l} c_{i2\lambda_j}^{L} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L} y + v_{i2}^{L-} - v_{i2}^{L+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{L}, \\
& \sum_{j=0}^{l} c_{i2\lambda_j}^{R} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{R} y + v_{i2}^{R-} - v_{i2}^{R+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{R}, \\
& v_{i2}^{L-}, v_{i2}^{L+}, v_{i2}^{R-}, v_{i2}^{R+} \geqslant 0, \\
& v_{i2}^{L-} \cdot v_{i2}^{L+} = 0, v_{i2}^{R-} \cdot v_{i2}^{R+} = 0 \qquad (4.9\text{d}) \\
& i = 1, 2, \ldots, t, \\
& A_{2}{}_{\lambda_j}^{L} x + B_{2}{}_{\lambda_j}^{L} y \leqslant b_{2}{}_{\lambda_j}^{L}, \\
& A_{2}{}_{\lambda_j}^{R} x + B_{2}{}_{\lambda_j}^{R} y \leqslant b_{2}{}_{\lambda_j}^{R}, \\
& j = 0, 1, \ldots, l,
\end{aligned}$$

Theorem 4.1.1. *Let* $(x'^{*}, y'^{*}) = (x^*, v_{11}^{L-*}, v_{11}^{L+*}, v_{11}^{R-*}, v_{11}^{R+*}, \ldots, v_{s1}^{L-*}, v_{s1}^{L+*}, v_{s1}^{R-*}, v_{s1}^{R+*}, y^*, v_{12}^{L-*}, v_{12}^{L+*}, v_{12}^{R-*}, v_{12}^{R+*}, \ldots, v_{t2}^{L-*}, v_{t2}^{L+*}, v_{t2}^{R-*}, v_{t2}^{R+*})$ *be the optimal solution to bi-level problem (4.9), then* (x^*, y^*) *is the optimal solution to the bi-level problem defined* *by (4.7).*

Proof. By Definition 4.1.1, let the notations associated with problem (4.7) are denoted by:

$$S = \{(x,y) : A_{k\lambda_j}^{L} x + B_{k\lambda_j}^{L} y \leqslant b_{k\lambda_j}^{L}, \\ A_{k\lambda_j}^{R} x + B_{k\lambda_j}^{R} y \leqslant b_{k\lambda_j}^{R}, k = 1,2, j = 0,1\ldots,l,\} \tag{4.10a}$$

$$S(X) = \{x \in X : \exists y \in Y, A_{k\lambda_j}^{L} x + B_{k\lambda_j}^{L} y \leqslant b_{k\lambda_j}^{L}, \\ A_{k\lambda_j}^{R} x + B_{k\lambda_j}^{R} y \leqslant b_{k\lambda_j}^{R}, k = 1,2, j = 0,1\ldots,l,\} \tag{4.10b}$$

$$S(x) = \{y \in Y : (x,y) \in S\} \tag{4.10c}$$

$$P(x) = \{y \in Y : y \in argmin\ \Psi\} \tag{4.10d}$$

where $\Psi = \frac{1}{l+1}\sum_{i=1}^{t}\sum_{j=0}^{l}\{|c_{i2\lambda_j}^{L} x + d_{i2\lambda_j}^{L}\hat{y} - g_{Fi\lambda_j}^{L}| + |c_{i2\lambda_j}^{R} x + d_{i2\lambda_j}^{R}\hat{y} - g_{Fi\lambda_j}^{R}|, \hat{y} \in S(x)\}$

$$IR = \{(x,y) : (x,y) \in S, y \in P(x)\} \tag{4.10e}$$

Problem (4.7) can be written as

$$\min_{x \in X} \frac{1}{l+1}\sum_{h=1}^{s}\sum_{j=0}^{l}\{|c_{h1\lambda_j}^{L} x + d_{h1\lambda_j}^{L} y - g_{Lh\lambda_j}^{L}| + |c_{h1\lambda_j}^{R} x + d_{h1\lambda_j}^{R} y - g_{Lh\lambda_j}^{R}|\}$$

$$\text{subject to } (x,y) \in IR \tag{4.11}$$

And those of problem (4.9) are denoted by:

$$S' = \{(x',y') : A_{k\lambda_j}^{L} x + B_{k\lambda_j}^{L} y \leqslant b_{k\lambda_j}^{L}, \tag{4.12a}$$

$$A_{k\lambda_j}^{R} x + B_{k\lambda_j}^{R} y \leqslant b_{k\lambda_j}^{R}, k = 1,2, j = 0,1\ldots,l$$

$$\sum_{j=0}^{l} c_{h1\lambda_j}^{L} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y + v_{h1}^{L-} - v_{h1}^{L+} = \sum_{j=0}^{l} g_{Lh\lambda_j}^{L},$$

$$\sum_{j=0}^{l} c_{h1\lambda_j}^{R} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y + v_{h1}^{R-} - v_{h1}^{R+} = \sum_{j=0}^{l} g_{Lh\lambda_j}^{R},$$

$$v_{h1}^{L-}, v_{h1}^{L+}, v_{h1}^{R-}, v_{h1}^{R+} \geqslant 0, v_{h1}^{L-} \cdot v_{h1}^{L+} = 0, v_{h1}^{R-} \cdot v_{h1}^{R+} = 0, h = 1,2,\ldots,s,$$

$$\sum_{j=0}^{l} c_{i2\lambda_j}^{L} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L} y + v_{i2}^{L-} - v_{i2}^{L+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{L},$$

$$\sum_{j=0}^{l} c_{i2\lambda_j}^{R} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{R} y + v_{i2}^{R-} - v_{i2}^{R+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{R},$$

$$v_{i2}^{L-}, v_{i2}^{L+}, v_{i2}^{R-}, v_{i2}^{R+} \geqslant 0, v_{i2}^{L-} \cdot v_{i2}^{L+} = 0, v_{i2}^{R-} \cdot v_{i2}^{R+} = 0, i = 1, 2, \ldots, t, \}$$

$$S(X') = \{x' \in X' : \exists y' \in Y', A_{k\lambda_j}^{L} x + B_{k\lambda_j}^{L} y \leqslant b_{k\lambda_j}^{L}, \tag{4.12b}$$

$$A_{k\lambda_j}^{R} x + B_{k\lambda_j}^{R} y \leqslant b_{k\lambda_j}^{R}, k = 1, 2, j = 0, 1 \ldots, l,$$

$$\sum_{j=0}^{l} c_{h1\lambda_j}^{L} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y + v_{h1}^{L-} - v_{h1}^{L+} = \sum_{j=0}^{l} g_{Lh\lambda_j}^{L},$$

$$\sum_{j=0}^{l} c_{h1\lambda_j}^{R} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y + v_{h1}^{R-} - v_{h1}^{R+} = \sum_{j=0}^{l} g_{Lh\lambda_j}^{R},$$

$$v_{h1}^{L-}, v_{h1}^{L+}, v_{h1}^{R-}, v_{h1}^{R+} \geqslant 0, v_{h1}^{L-} \cdot v_{h1}^{L+} = 0, v_{h1}^{R-} \cdot v_{h1}^{R+} = 0, h = 1, 2, \ldots, s,$$

$$\sum_{j=0}^{l} c_{i2\lambda_j}^{L} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L} y + v_{i2}^{L-} - v_{i2}^{L+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{L},$$

$$\sum_{j=0}^{l} c_{i2\lambda_j}^{R} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{R} y + v_{i2}^{R-} - v_{i2}^{R+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{R},$$

$$v_{i2}^{L-}, v_{i2}^{L+}, v_{i2}^{R-}, v_{i2}^{R+} \geqslant 0, v_{i2}^{L-} \cdot v_{i2}^{L+} = 0, v_{i2}^{R-} \cdot v_{i2}^{R+} = 0, i = 1, 2, \ldots, t, \}$$

$$S(x') = \{y' \in Y' : (x', y') \in S'\} \tag{4.12c}$$

$$P(x') = \{y' \in Y' :$$

$$y' \in argmin[\sum_{i=1}^{t} (\hat{v}_{i2}^{L-} + \hat{v}_{i2}^{L+} + \hat{v}_{i2}^{R-} + \hat{v}_{i2}^{R+}) : \hat{y'} \in S(x')]\} \tag{4.12d}$$

$$IR' = \{(x', y') : (x', y') \in S', y' \in P(x')\} \tag{4.12e}$$

Problem (4.9) can be written as

$$\min_{x' \in X'} \{\sum_{h=1}^{l} (v_{h1}^{L-} + v_{h1}^{L+} + v_{h1}^{R-} + v_{h1}^{R+}) : (x', y') \in IR'\} \tag{4.13}$$

As $(x^{'*}, y^{'*})$ is the optimal solution to problem (4.9), from (4.13), it can be seen that, $\forall (x', y') \in IR'$, we have:

$$\sum_{h=1}^{l} (v_{h1}^{L-} + v_{h1}^{L+} + v_{h1}^{R-} + v_{h1}^{R+}) \geqslant \sum_{h=1}^{l} (v_{h1}^{L-*} + v_{h1}^{L+*} + v_{h1}^{R-*} + v_{h1}^{R+*})$$

As: $\sum\limits_{j=0}^{l} c_{h1\lambda_j}^{L} x + \sum\limits_{j=0}^{l} d_{h1\lambda_j}^{L} y + v_{h1}^{L-} - v_{h1}^{L+} = \sum\limits_{j=0}^{l} g_{Lh\lambda_j}^{L}$ and $v_{h1}^{L-} \cdot v_{h1}^{L+} = 0$, $h =$

$1, 2, \ldots, s$, we have:

$$v_{h1}^{-} + v_{h1}^{+} = |\sum_{j=0}^{l} c_{h1\lambda_j}^{L} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}|,$$

$$v_{h1}^{L-*} + v_{h1}^{L+*} = |\sum_{j=0}^{l} c_{h1\lambda_j}^{L} x^* + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y^* - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}|, h = 1, 2, \ldots, s$$

Similarly, we have:

$$v_{h1}^{R-} + v_{h1}^{R+} = |\sum_{j=0}^{l} c_{h1\lambda_j}^{R} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}|,$$

$$v_{h1}^{R-*} + v_{h1}^{R+*} = |\sum_{j=0}^{l} c_{h1\lambda_j}^{R} x^* + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y^* - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}|, h = 1, 2, \ldots, s$$

So: $\forall (x', y') \in IR'$,

$$|\sum_{j=0}^{l} c_{h1\lambda_j}^{L} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}| + |\sum_{j=0}^{l} c_{h1\lambda_j}^{R} x + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}|$$
$$\geqslant |\sum_{j=0}^{l} c_{h1\lambda_j}^{L} x^* + \sum_{j=0}^{l} d_{h1\lambda_j}^{L} y^* - \sum_{j=0}^{l} g_{Lh\lambda_j}^{L}| + |\sum_{j=0}^{l} c_{h1\lambda_j}^{R} x^* + \sum_{j=0}^{l} d_{h1\lambda_j}^{R} y^* - \sum_{j=0}^{l} g_{Lh\lambda_j}^{R}|,$$
$$h = 1, 2, \ldots, s \tag{4.14}$$

We now prove that the projection of S' onto the $X \times Y$ space, denoted by $S'|_{X,Y}$, is equal to S:

On the one hand, $\forall (x, y) \in S'|_{X,Y}$, from constraints: $A_{k\lambda_j}^{L} x + B_{k\lambda_j}^{L} y \leqslant b_{k\lambda_j}^{L}$, $A_{k\lambda_j}^{R} x + B_{k\lambda_j}^{R} y \leqslant b_{k\lambda_j}^{R}, k = 1, 2, j = 0, 1 \ldots, l$, in S', we have: $(x, y) \in S$, so $S'|_{X,Y} \subseteq S$.

On the other hand, $\forall (x, y) \in S$, by (4.8), we can always find v_{11}^{L-}, v_{11}^{L+}, v_{11}^{R-}, v_{11}^{R+},..., v_{s1}^{L-}, v_{s1}^{L+}, v_{s1}^{R-}, v_{s1}^{R+}, v_{12}^{L-}, v_{12}^{L+}, v_{12}^{R-}, v_{12}^{R+}, ..., v_{t2}^{L-}, v_{t2}^{L+}, v_{t2}^{R-}, v_{t2}^{R+} which satisfies the constraints of (4.9b) and (4.9d). Together with the inequations of $A_{k\lambda_j}^{L} x + B_{k\lambda_j}^{L} y \leqslant b_{k\lambda_j}^{L}$, and $A_{k\lambda_j}^{R} x + B_{k\lambda_j}^{R} y \leqslant b_{k\lambda_j}^{R}, k = 1, 2, j = 0, 1 \ldots, l$, requested by S, we have $(x, v_{11}^{L-}, v_{11}^{L+}, v_{11}^{R-}, v_{11}^{R+}, \ldots, v_{s1}^{L-}, v_{s1}^{L+}, v_{s1}^{R-}, v_{s1}^{R+}, y, v_{12}^{L-}, v_{12}^{L+}, v_{12}^{R-}, v_{12}^{R+}, \ldots, v_{t2}^{L-}, v_{t2}^{L+}, v_{t2}^{R-}, v_{t2}^{R+}) \in S'$, thus $(x, y) \in S'|_{X,Y}$, $S \subseteq S'|_{X,Y}$.

So, we can prove that:

$$S'|_{X,Y} = S \tag{4.15}$$

Similarly, we have

$$S(x)'|_{X,Y} = S(x) \tag{4.16a}$$

$$S(X)'|_{X,Y} = S(X) \tag{4.16b}$$

Also, from:

$$\sum_{j=0}^{l} c_{i2\lambda_j}^{L} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L} y + v_{i2}^{L-} - v_{i2}^{L+} = \sum_{j=0}^{l} g_{Fi\lambda_j}^{L}$$

and

$$v_{i2}^{L-} \cdot v_{i2}^{L+} = 0, i = 1, 2, \ldots, t$$

we have:

$$v_{i2}^{L-} + v_{i2}^{L+} = |\sum_{j=0}^{l} c_{i2\lambda_j}^{L} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{L} y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{L}|, i = 1, 2, \ldots, t, \tag{4.17a}$$

Similarly, we have:

$$v_{i2}^{R-} + v_{i2}^{R+} = |\sum_{j=0}^{l} c_{i2\lambda_j}^{R} x + \sum_{j=0}^{l} d_{i2\lambda_j}^{R} y - \sum_{j=0}^{l} g_{Fi\lambda_j}^{R}|, i = 1, 2, \ldots, t \tag{4.17b}$$

Thus:

$$P(x') = \{y' \in Y' : y' \in argmin\Psi'\} \tag{4.18}$$

where $\Psi' = \sum_{i=1}^{t} \sum_{j=0}^{l} \{|c_{i2\lambda_j}^{L} x + d_{i2\lambda_j}^{L} \hat{y} - g_{Fi\lambda_j}^{L}| + |c_{i2\lambda_j}^{R} x + d_{i2\lambda_j}^{R} \hat{y} - g_{Fi\lambda_j}^{R}|, \hat{y} \in S(x')\}$

From (4.15) and (4.18), we get:

$$P(x')|_{X \times Y} = P(x) \tag{4.19}$$

From (4.10e), (4.12e), (4.15) and (4.19), we get:

$$IR'|_{X \times Y} = IR \tag{4.20}$$

which means, in $X \times Y$ space, the leaders of problem (4.7) and (4.9) have the same optimising space.

Thus, from (4.14) and (4.20), it can be obtained that: $\forall (x, y) \in IR$, we have:

$\frac{1}{l+1} \sum_{h=1}^{s} \sum_{j=0}^{l} \{|c_{h1\lambda_j}^{L} x + d_{h1\lambda_j}^{L} y - g_{Lh\lambda_j}^{L}| + |c_{h1\lambda_j}^{R} x + d_{h1\lambda_j}^{R} y - g_{Lh\lambda_j}^{R}|\}$

$\geqslant \frac{1}{l+1} \sum_{h=1}^{s} \sum_{j=0}^{l} \{|c_{h1\lambda_j}^{L} x^* + d_{h1\lambda_j}^{L} y^* - g_{Lh\lambda_j}^{L}| + |c_{h1\lambda_j}^{R} x^* + d_{h1\lambda_j}^{R} y^* - g_{Lh\lambda_j}^{R}|\}$,

So: (x^*, y^*) is the optimal solution of the problem (4.7). $\square$

Adopting weighting method, (4.9) can be further transferred into (4.21):

$$\min_{x' \in X'} v_1^- + v_1^+ \tag{4.21a}$$

$$\begin{aligned} \text{subject to } & c_1 x + d_1 y \\ & +v_1^- - v_1^+ = \sum_{h=1}^{s} \sum_{j=0}^{l} (g_{Lh\lambda_j}^L + g_{Lh\lambda_j}^R), \\ & v_1^-, v_1^+ \geqslant 0, \\ & v_1^- \cdot v_1^+ = 0 \\ & A_1{}_{\lambda_j}^L x + B_1{}_{\lambda_j}^L y \leqslant b_1{}_{\lambda_j}^L, \\ & A_1{}_{\lambda_j}^R x + B_1{}_{\lambda_j}^R y \leqslant b_1{}_{\lambda_j}^R, \\ & j = 0, 1, \ldots, l, \end{aligned} \tag{4.21b}$$

$$\min_{y' \in Y'} v_2^- + v_2^+ \tag{4.21c}$$

$$\begin{aligned} \text{subject to } & c_2 x + d_2 y = \sum_{i=1}^{t} \sum_{j=0}^{l} (g_{Fi\lambda_j}^L + g_{Fi\lambda_j}^R), \\ & v_2^-, v_2^+ \geqslant 0, \\ & v_2^- \cdot v_2^+ = 0 \\ & A_2{}_{\lambda_j}^L x + B_2{}_{\lambda_j}^L y \leqslant b_2{}_{\lambda_j}^L, \\ & A_2{}_{\lambda_j}^R x + B_2{}_{\lambda_j}^R y \leqslant b_2{}_{\lambda_j}^R, \\ & j = 0, 1, \ldots, l, \end{aligned} \tag{4.21d}$$

where $v_1^- = \sum_{h=1}^{s}(v_{h1}^{L-} + v_{h1}^{R-})$, $v_1^+ = \sum_{h=1}^{s}(v_{h1}^{L+} + v_{h1}^{R+})$, $v_2^- = \sum_{i=1}^{t}(v_{i2}^{L-} + v_{i2}^{R-})$, $v_2^+ = \sum_{i=1}^{t}(v_{i2}^{L+} + v_{i2}^{R+})$, $c_1 = \sum_{h=1}^{s}\sum_{j=0}^{l}(c_{h1\lambda_j}^L + c_{h1\lambda_j}^R)$, $d_1 = \sum_{h=1}^{s}\sum_{j=0}^{l}(d_{h1\lambda_j}^L + d_{h1\lambda_j}^R)$, $c_2 = \sum_{i=1}^{t}\sum_{j=0}^{l}(c_{i2\lambda_j}^L + c_{i2\lambda_j}^R)$, $d_2 = \sum_{i=1}^{t}\sum_{j=0}^{l}(d_{i2\lambda_j}^L + d_{i2\lambda_j}^R)$.

In the above formula, v_1^- and v_1^+ are deviational variables representing the under-achievement and over-achievement of goals for a leader, and v_2^- and v_2^+ are deviational variables representing the under-achievement and over-achievement of goals for a follower respectively.

The nonlinear conditions of $v_1^- \cdot v_1^+ = 0$, and $v_2^- \cdot v_2^+ = 0$ need not be maintained if the Kuhn-Tucker algorithm(Shi *et al.* 2005b) together with the Simplex algorithm are adopted, since only equivalence at an optimum is wanted. Further explanation can be found from (Charnes & Cooper 1961a). Thus, problem (4.21) is further transformed into:

For $(v_1^-, v_1^+) \in R^2$, $\bar{X}' \subseteq X \times R^2$, $(v_2^-, v_2^+) \in R^2$, $\bar{Y}' \subseteq Y \times R^2$, let $x = (x_1, \cdots, x_n) \in X$, $\bar{x}' = (x_1, \cdots, x_n, v_1^-, v_1^+) \in \bar{X}'$, $y = (y_1, \cdots, y_m) \in Y$, $\bar{y}' = (y_1, \cdots, y_m,$

$v_2^-, v_2^+,) \in \bar{Y}'$, and $v_1, v_2 : \bar{X}' \times \bar{Y}' \rightarrow F(R)$.

$$\min_{(x, v_1^-, v_1^+) \in \bar{X}'} v_1 = v_1^- + v_1^+ \tag{4.22a}$$

$$\begin{aligned} \text{subject to } & c_1 x + d_1 y \\ & +v_1^- - v_1^+ = \sum_{h=1}^{s} \sum_{j=0}^{l} (g_{Lh\lambda_j}^L + g_{Lh\lambda_j}^R), \\ & A_{1\lambda_j}^L x + B_{1\lambda_j}^L y \leqslant b_{1\lambda_j}^L, \\ & A_{1\lambda_j}^R x + B_{1\lambda_j}^R y \leqslant b_{1\lambda_j}^R, \\ & j = 0, 1, \ldots, l, \end{aligned} \tag{4.22b}$$

$$\min_{(y, v_2^-, v_2^+) \in \bar{Y}'} v_2 = v_2^- + v_2^+ \tag{4.22c}$$

$$\begin{aligned} \text{subject to } & c_2 x + d_2 y = \sum_{i=1}^{t} \sum_{j=0}^{l} (g_{Fi\lambda_j}^L + g_{Fi\lambda_j}^R), \\ & A_{2\lambda_j}^L x + B_{2\lambda_j}^L y \leqslant b_{2\lambda_j}^L, \\ & A_{2\lambda_j}^R x + B_{2\lambda_j}^R y \leqslant b_{2\lambda_j}^R, \\ & j = 0, 1, \ldots, l, \end{aligned} \tag{4.22d}$$

Problem (4.22) is a standard linear bi-level problem which can be solved by the Kuhn-Tucker algorithm (Shi *et al.* 2005b).

4.2 A λ-cut and Goal-programming-based Algorithm

Based on the analysis above, we illustrate the λ-cut and goal-programming-based algorithm in this section.

First, we obtain relevant parameters including the coefficients that define an FMOLB decision problem, satisfactory degree, and a predefined error (Step 1).

Then, using λ-cuts, we defuzzify the FMOLB decision problem as an MOB decision problem (Step 2).

Afterwards, by introducing the under-achievement auxiliary variables and over-achievement auxiliary variables, we need to solve a classical linear bi-level decision problem under current λ-cuts at this stage (Step 3).

Finally, if the solution difference between current adjacent λ-cuts is close enough (equal to or smaller than the predefined error) (Step 4), the final solution is expected to be reached (Step 7). Otherwise, we double the λ-cuts (Step 5), and do the computation again (Step 6).

the λ-cut and goal-programming-based algorithm is detailed in Algorithm 2.

Algorithm 2: A λ-cut and goal-programming-based algorithm for FMOLB decision problems

Step 1 (Input) Get relevant coefficients which include: Parameters of (4.3); Parameters of (4.5); Satisfactory degree: α; and $\varepsilon > 0$.
Step 2 (Initialise) Let $k = 1$, which is the counter to record current loop. In (4.7), where $\lambda_j \in [\alpha, 1]$, let $\lambda_0 = \alpha$ and $\lambda_1 = 1$ respectively, then each objective will be transferred into two non-fuzzy objective functions, and each fuzzy constraint is converted into four non-fuzzy constraints.
Step 3 (Compute) By introducing auxiliary variables v_1^-, v_1^+, v_2^- and v_2^+, we get the format of (4.22). The solution $(x, v_1^-, v_1^+, y, v_2^-, v_2^+)_2$ of (4.22) is obtained by Kuhn-Tucker approach.
Step 4 (Compare)
if $(k = 1)$ **then**
  $(x, v_1^-, v_1^+, y, v_2^-, v_2^+)_1 = (x, v_1^-, v_1^+, y, v_2^-, v_2^+)_2$; **goto [Step 5]**;
end
if $||(x, v_1^-, v_1^+, y, v_2^-, v_2^+)_2 - (x, v_1^-, v_1^+, y, v_2^-, v_2^+)_1|| < \varepsilon$ **then**
  goto [Step 7] ;
end
Step 5 (Split) Suppose there are $(L + 1)$ nodes λ_j, $(j = 0, 1, \ldots, L)$ in the interval $[\alpha, 1]$, insert L new nodes δ_t $(t = 1, 2 \ldots, L)$ in $[\alpha, 1]$ such that: $\delta_t = (\lambda_{t-1} + \lambda_t)/2$.
Step 6 (Loop) $k = k + 1$; **goto [Step 3]**;
Step 7 (Output) $(x, y)_2$ is obtained as the final solution.

4.3 A Case Study

A classical newsboy problem is to find a newspaper's order quantity for maximising the profit of a newsboy (newspaper retailer) (Ji & Shao 2006). In a real world situation, both a newspaper manufacturer and a retailer have more than one concern. Using an FMOLB model, a newsboy problem is expressed as follows: the leader, a manufacturer controls the decision variable of the wholesale price (x), while the follower, a retailer, decides his or her order quantity (y). The manufacturer has two main objectives: to maximise the net profits, represented by $F_1(x, y)$, and to maximise the newspaper quality, by $F_2(x, y)$ but subject to some constraints, including the requirements of material, marketing cost and labor cost. The retailer also has two objectives to achieve: to minimise his or her purchase cost, represented by $f_1(x, y)$, and to minimise the working hours, by $f_2(x, y)$ under his own constraints. Meanwhile, both the manufacturer and the retailer will set goals $(g_{L1}, g_{L2}, g_{F1}, g_{F1})$ for each of their two objectives.

When modelling this multi-objective bi-level decision problem, the main difficulty is to establish coefficients of the objectives and constraints for both the leader and the follower. We can only estimate some values for material cost, labor cost, etc. according to our experience and previous data. For some items, the values can only be assigned by linguistic terms as about \$1000. This is a common case in any organisational decision practice. By using fuzzy numbers to describe these uncertain values in coefficients, an FMOLB model can be established for this decision problem.

To illustrate the λ-cut and goal-programming-based algorithm, this newsboy problem will be solved step by step:

[Step 1]: (Input the relevant coefficients)

1. Coefficients of (4.3):

The newsboy problem is formatted as:

$$\begin{aligned}
\text{Leader}: &\max_{x \in X} F_1(x,y) = \tilde{6}x + \tilde{3}y \\
&\max_{x \in X} F_2(x,y) = -\tilde{3}x + \tilde{6}y \\
&\text{subject to } -\tilde{1}x + \tilde{3}y \leqslant \tilde{21} \\
\text{Follower}: &\min_{y \in Y} f_1(x,y) = \tilde{4}x + \tilde{3}y \\
&\min_{y \in Y} f_2(x,y) = \tilde{3}x + \tilde{1}y \\
&\text{subject to } -\tilde{1}x - \tilde{3}y \leqslant \tilde{27}
\end{aligned}$$

where $x \in R^1, y \in R^1$, and $X \in R^+, Y \in R^+$.

The membership functions for this FMOLB are as follows:

$$\mu_{\tilde{6}}(x) = \begin{cases} 0 & x < 5 \\ \frac{x^2-25}{11} & 5 \leqslant x < 8 \\ 1 & x = 6 \\ \frac{64-x^2}{28} & 6 < x \leqslant 8 \\ 0 & x > 8 \end{cases}, \qquad \mu_{\tilde{3}}(x) = \begin{cases} 0 & x < 2 \\ \frac{x^2-4}{5} & 2 \leqslant x < 3 \\ 1 & x = 3 \\ \frac{25-x^2}{16} & 3 < x \leqslant 5 \\ 0 & x > 5 \end{cases},$$

$$\mu_{\tilde{-3}}(x) = \begin{cases} 0 & x < -4 \\ \frac{16-x^2}{7} & -4 \leqslant x < -3 \\ 1 & x = -3 \\ \frac{x^2-1}{8} & -3 < x \leqslant -1 \\ 0 & x > -1 \end{cases}, \qquad \mu_{\tilde{4}}(x) = \begin{cases} 0 & x < 3 \\ \frac{x^2-9}{7} & 3 \leqslant x < 4 \\ 1 & x = 4 \\ \frac{36-x^2}{20} & 4 < x \leqslant 6 \\ 0 & x > 6 \end{cases},$$

$$\mu_{\tilde{1}}(x)=\begin{cases}0 & x<0.5\\ \frac{x^2-0.25}{0.75} & 0.5\leqslant x<1\\ 1 & x=1\\ \frac{4-x^2}{3} & 1<x\leqslant 2\\ 0 & x>2\end{cases},\quad \mu_{\tilde{-1}}(x)=\begin{cases}0 & x<-2\\ \frac{4-x^2}{3} & -2\leqslant x<-1\\ 1 & x=-1\\ \frac{x^2-0.25}{0.75} & -1<x\leqslant -0.5\\ 0 & x>-0.5\end{cases},$$

$$\mu_{\tilde{21}}(x)=\begin{cases}0 & x<19\\ \frac{x^2-361}{80} & 19\leqslant x<21\\ 1 & x=21\\ \frac{625-x^2}{184} & 21<x\leqslant 25\\ 0 & x>25\end{cases},\quad \mu_{\tilde{27}}(x)=\begin{cases}0 & x<25\\ \frac{x^2-625}{104} & 25\leqslant x<27\\ 1 & x=27\\ \frac{961-x^2}{232} & 27<x\leqslant 31\\ 0 & x>31\end{cases}.$$

2. Suppose the membership functions of the fuzzy goals set for the leader are:

$$\mu_{\tilde{g}_{L1}}(x)=\begin{cases}0 & x<15\\ \frac{x^2-225}{175} & 15\leqslant x<20\\ 1 & x=20\\ \frac{900-x^2}{500} & 20<x\leqslant 30\\ 0 & x>30\end{cases},\ \mu_{\tilde{g}_{L2}}(x)=\begin{cases}0 & x<4\\ \frac{x^2-16}{48} & 4\leqslant x<8\\ 1 & x=8\\ \frac{225-x^2}{161} & 8<x\leqslant 15\\ 0 & x>15\end{cases}.$$

The membership functions of the fuzzy goals set for the follower are:

$$\mu_{\tilde{g}_{F1}}(x)=\begin{cases}0 & x<10\\ \frac{x^2-100}{225} & 10\leqslant x<15\\ 1 & x=15\\ \frac{400-x^2}{175} & 15<x\leqslant 20\\ 0 & x>20\end{cases},\ \mu_{\tilde{g}_{F2}}(x)=\begin{cases}0 & x<7\\ \frac{x^2-49}{32} & 7\leqslant x<9\\ 1 & x=9\\ \frac{121-x^2}{40} & 9<x\leqslant 11\\ 0 & x>11\end{cases}.$$

3. Satisfactory degree: $\alpha=0.2$

4. $\varepsilon=0.15$

[Step 2]: (Initialise) Let k=1. Associated with this example, the corresponding MOB$_\lambda$ problem is:

$$\begin{aligned}&\min_{x\in X}|\sqrt{11\lambda+25}x+\sqrt{5\lambda+4}y-\sqrt{175\lambda+225}|\\&\qquad+|\sqrt{64-28\lambda}x+25-\sqrt{25-16\lambda}y-\sqrt{900-500\lambda}|\\&\min_{x\in X}|-\sqrt{16-7\lambda}x+\sqrt{11\lambda+25}y-\sqrt{48\lambda+16}|\\&\qquad+|-\sqrt{8\lambda+1}+\sqrt{64-28\lambda}-\sqrt{225-161\lambda}|\end{aligned}$$

$$
\begin{aligned}
&\text{subject to } -\sqrt{4-2\lambda}x+\sqrt{5\lambda+4}y \leqslant \sqrt{80\lambda+361}\\
&\qquad -\sqrt{-0.75\lambda+0.25}x+\sqrt{25-16\lambda}y \leqslant \sqrt{625-184\lambda}\\
&\min_{y\in Y} |\sqrt{7\lambda+9}x+\sqrt{5\lambda+4}y-\sqrt{225\lambda+100}|\\
&\qquad +|\sqrt{36-20\lambda}x+25-\sqrt{25-16\lambda}y-\sqrt{400-175\lambda}|\\
&\min_{y\in Y} |-\sqrt{5\lambda+4}x+\sqrt{0.75\lambda+0.25}y-\sqrt{32\lambda+49}|\\
&+|-\sqrt{25-16\lambda}x+\sqrt{4-3\lambda}y-\sqrt{121-40\lambda}|\\
&\qquad \text{subject to } \sqrt{0.75\lambda+0.25}x+\sqrt{5\lambda+4}y \leqslant \sqrt{104\lambda+625}\\
&\qquad\qquad \sqrt{4-3\lambda}x+\sqrt{25-16\lambda}y \leqslant \sqrt{901-232\lambda}
\end{aligned}
$$

where $\lambda \in [0.2, 1]$.

Referring to the algorithm, only $\lambda_0 = 0.2$ and $\lambda_1 = 1$ are considered initially. Thus four non-fuzzy objective functions and four non-fuzzy constraints for the leader and follower are generated respectively:

$$
\begin{aligned}
&\min_{x\in X} \tfrac{1}{4}\{|\sqrt{27.2}x+\sqrt{5}y-\sqrt{260}|+|6x+3y-20|\\
&\qquad +|\sqrt{58.4}x+\sqrt{21.8}y-20\sqrt{2}|+|6x+3y-20|\\
&\qquad +|-\sqrt{14.6}x+\sqrt{27.2}y-\sqrt{25.6}|+|-3x+6y-8|\\
&\qquad +|-\sqrt{2.6}+\sqrt{58.4}y-\sqrt{192.8}|+|-3x+6y-8|\}\\
&\text{subject to } -\sqrt{3.4}x+\sqrt{5}y \leqslant \sqrt{377}\\
&\qquad -x+3y \leqslant 21\\
&\qquad -\sqrt{0.4}+\sqrt{5}y \leqslant \sqrt{645.8}\\
&\qquad -x+3y \leqslant 21\\
&\qquad \min_{y\in Y} \tfrac{1}{4}\{|3x+2y-12.04|+|4x+3y-19.1|\\
&\qquad\qquad +|6x-5y-7.4|+|4x-3y-10.63|\\
&\qquad\qquad +|-2x+0.5y-18.03|+|-3x+y-15|\\
&\qquad\qquad +|-5x+2y-9|+|-3x+y-9|\}\\
&\qquad \text{subject to } \sqrt{0.4}x+\sqrt{5}y \leqslant \sqrt{645.8}\\
&\qquad\qquad x+3y \leqslant 27\\
&\qquad\qquad \sqrt{3.4}x+\sqrt{21.8}y \leqslant \sqrt{914.6}\\
&\qquad\qquad x+3y \leqslant 27
\end{aligned}
$$

[Step 3]: (Compute)

By introducing auxiliary variables $v_1^-, v_1^+, v_2^-, v_2^+$, we have:

$$
\begin{aligned}
&\min_{(x, v_1^-, v_1^+) \in \bar{X}'} v_1^- + v_1^+ \\
&\text{subject to } 3.083x + 20.076y + v_1^- - v_1^+ = 54.73, \\
&\qquad -1.8x + 2.2y \leqslant 19.4 \\
&\qquad -x + 3y \leqslant 21 \\
&\qquad -0.6x + 4.7y \leqslant 24.3 \\
&\qquad -x + 3y \leqslant 21 \\
&\qquad \min_{(y, v_2^-, v_2^+) \in \bar{Y}'} v_2^- + v_2^+ \\
&\qquad \text{subject to } 16.498x + 8.205y + v_2^- - v_2^+ = 51.337, \\
&\qquad\qquad 0.6x + 2.2y \leqslant 25.4 \\
&\qquad\qquad x + 3y \leqslant 7 \\
&\qquad\qquad 1.8x + 4.7y \leqslant 30.2 \\
&\qquad\qquad x + 3y \leqslant 27
\end{aligned}
$$

Using Branch-and-bound algorithm (Bard & Moore 1990), the current solution is (1.901,0,0, 2.434,0,0).

[Step 4]: (Compare) Because k=1, goto **[Step 5]**.

[Step 5]: (Split) By inserting a new node $\lambda_1 = (0.2 + 1)/2 = 0.6$, there are a total three nodes of $\lambda_0 = 0.2, \lambda_1 = 0.6$ and $\lambda_2 = 1$. Then a total six non-fuzzy objective functions for the leader and follower, together with six non-fuzzy constraints for the leader and follower respectively, are generated.

[Step 6]: (Loop) k=1+1=2, goto **[Step 3]**, and a current solution of (2.011,0,0, 2.356,0,0) is obtained. As $|2.011 - 1.901| + |2.356 - 2.434| = 0.188 > \varepsilon = 0.15$, the algorithm continues until the solution of (1.957,0,0, 2.388,0,0) is obtained. The computing results are listed in Table 4.1.

Table 4.1: Summary of the running solutions

k	x	y	$v_{1\lambda}^+$	$v_{1\lambda}^-$	$v_{2\lambda}^+$	$v_{2\lambda}^-$
1	1.901	2.434	0	0	0	0
2	2.011	2.356	0	0	0	0
3	1.872	2.446	0	0	0	0
4	1.957	2.388	0	0	0	0

[Step 7]: (Output) As $|1.957 - 1.872| + |2.388 - 2.2.446| = 0.14 < \varepsilon = 0.15$, $(x^*, y^*) = (1.957, 2.388)$ is the final solution of this FMOLB problem. The objectives for the leader and follower under $(x^*, y^*) = (1.957, 2.388)$ are:

$$\begin{cases} F_1(x^*, y^*) = F(1.957, 2.388) = 1.957\tilde{c_{11}} + 2.388\tilde{d_{11}} \\ F_2(x^*, y^*) = F(1.957, 2.388) = 1.957\tilde{c_{12}} + 2.388\tilde{d_{12}} \\ f_1(x^*, y^*) = F(1.957, 2.388) = 1.957\tilde{c_{21}} + 2.388\tilde{d_{21}} \\ f_2(x^*, y^*) = F(1.957, 2.388) = 1.957\tilde{c_{22}} + 2.388\tilde{d_{22}} \end{cases},$$

Under this solution, the membership functions for the leader's objectives are shown in Figure 4.1 and the membership functions for the follower's objectives are shown in Figure 4.2.

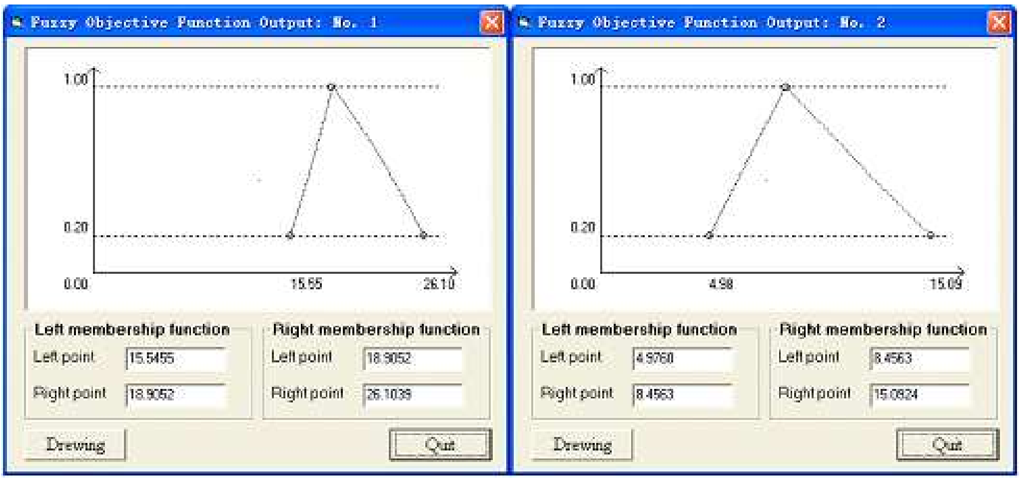

Figure 4.1: Membership functions of $F_1(x^*, y^*)$ and $F_2(x^*, y^*)$

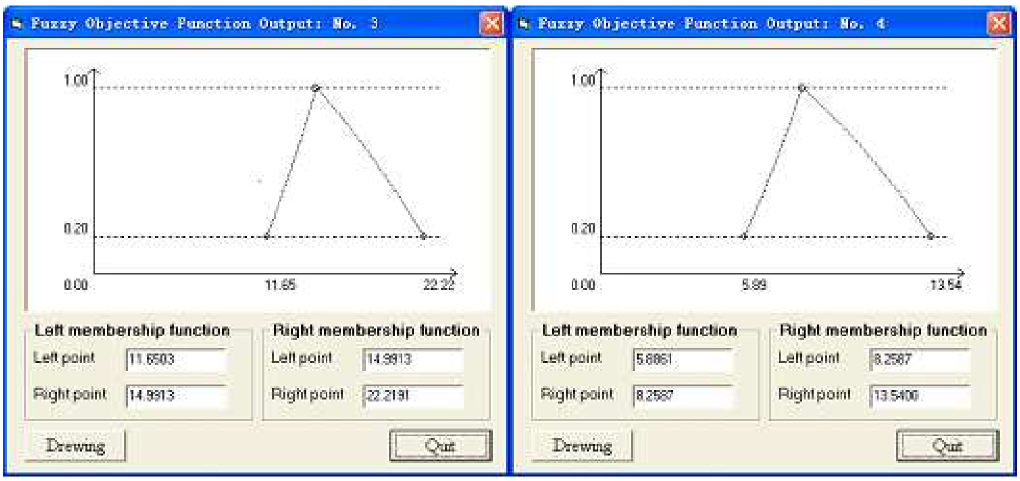

Figure 4.2: Membership functions of $f_1(x^*, y^*)$ and $f_2(x^*, y^*)$

These fuzzy values shown in Figure 4.1 and Figure 4.2 describe the achievements of every objective under the solutions. From Figure 4.1 we can see that if the manufacturer chooses his or her decision variable as 1.957, the most possible net profit will be 18.9025, which is very close to the goal set for this objective. The other objective values can be interpreted the same way.

4.4 Experiments and Evaluation

The algorithm proposed in this chapter was implemented by Visual Basic 6.0, and run on a desktop computer with CPU Pentium 4 2.8GHz, RAM 1G, Windows XP. To test the performance of the proposed algorithm, the following experiments are carried out.

(1) To test the efficiency of the proposed algorithm, we employ ten numerical examples and enlarge the problem scales by changing the numbers of decision variables, objective functions and constraints for both leaders and followers from two to ten simultaneously. For each of these examples, the final solution has been obtained within five seconds.

(2) To test the performance of the fuzzy distance measure in Definition 4.1.3, we adjust the satisfactory degree values from 0 to 0.5 on the ten numerical examples again. At the same time, we change some of the fuzzy coefficients in the constraints by moving the points whose membership values equal 0 by 10% from the left and right respectively. Experiments reveal that, when a satisfactory degree is set as 0, the average solution will change by about 6% if some of the constraint coefficients are moved as discussed above. When we increase satisfactory degrees, the average solution change decreases. At the point where satisfactory degrees are equal to 0.5, the average solution change is 0.

From Experiment (1), we can see that this algorithm is quite efficient. The reason is the fact that final solutions can be reached by solving corresponding linear bi-level programming problems, which can be handled by the Kuhn-Tucker and the Simplex algorithms.

From Experiment (2), we can see that if we change some coefficients of fuzzy numbers within a small range, solutions will be less sensitive to this change under a higher satisfactory degree. The reason is that, when the satisfactory degree is set to 0, every λ-cut of fuzzy coefficients from 0 to 1 will be considered. Thus, the decision maker can certainly be influenced by minor information.

For a decision making process involved with fuzzy coefficients, decision makers may sometimes make small adjustment on the uncertain information about the preference or circumstances. If the change occurs to the minor information, i.e. with smaller satisfactory degrees, there should normally be no tremendous change to the final solution. For example, when estimating future profit, the manufacturer may adjust the possibility of five thousand dollars' profit from 2% to 3%, while the possibility of one hundred thousand dollars' profit remains 100%. In such a situation, there should be no outstanding change for his or her final decision on the device investment. Therefore, to increase the satisfactory degrees is an acceptable strategy for a feasible solution.

From the above analysis, the advantages and disadvantages of the algorithm proposed in this chapter are as follows:

(1) This algorithm is quite efficient, as it adopts strategies to transform a non-linear bi-level problem into a linear problem;

(2) When pursuing optimality, the negative effect from conflicting objectives can be avoided and a leader can finally reach his or her satisfactory solution by setting goals for the objectives;

(3) The information of the original fuzzy numbers are considered adequately by using a certain number of λ-cuts to approximate the final precise solution;

(4) In some situations, this algorithm might suffer from expensive calculation, as the size of λ-cuts will increase exponentially with respect to iteration counts.

4.5 Summary

This chapter studies FMOLB decision problems by λ-cut and goal programming. After formulating an FMOLB decision problem, we have proved that the solutions can be obtained by solving the corresponding linear bi-level decision problem which can be handled easily by Kuhn-Tuchker and Simplex algorithms. Therefore, it is possible for the algorithm developed in this research to deal with FMOLB decision problems stably and effectively. Based on the theoretical proof, a λ-cut and goal-programming-based algorithm is proposed for FMOLB decision problems, and a newsboy problem is presented to further explain the idea of this algorithm.

This λ-cut and goal-programming-based algorithm is one of the computation kernels in a fuzzy bi-level decision support system developed to assist decision makers to solve realistic FMOLB problems. This system will be described in Chapter 8 in detail.

5 Cutset Strategy and PSO for Fuzzy Linear Multi-follower Multi-objective Bi-level Decision Making

This chapter focuses on linear bi-level decision problems that have multiple followers, multiple objectives, and fuzzy coefficients in the objectives and/or constraints of the leaders and/or the followers. We call this kinds of problem fuzzy multi-follower multi-objective linear bi-level (FMMLB) decision problems.

In this chapter, a framework, which is to define FMMLB decision problems by different cooperation in objectives, constraints, and decision variables among followers, is presented. Focusing on FMMLB decision problems defined in this framework, three algorithms, i.e. a Branch-and-Bound-based algorithm, a Kth-Best-based algorithm, and a PSO-based algorithm are developed. Experiments are then carried to compare these algorithms, and algorithm choosing is discussed.

5.1 Models

According to eight different cooperation situations among followers (Lu *et al.* 2006), we extend the models to fuzzy situations and give their corresponding mathematic models as below.

Model 1. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are shared decision variables $y_j (j = 1, 2, \dots, K)$, the same objective functions $f(x, y)$ and the same constraint functions among them, is defined as follows:

Let $x \in X \subseteq R^n$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1, 2, \dots, K$, $y = (y_1, \dots, y_K) \in Y = (Y_1, \dots, Y_K)$, $F(x, y) : X \times Y \to F^s(R)$, $f(x, y) : X \times Y \to F^t(R)$, it consists of finding a solution to the upper level problem:

$$\min_{x \in X} F(x,y) = \left(\tilde{c}_1^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j, \ldots, \tilde{c}_s^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j \right)^T \tag{5.1a}$$

$$\text{subject to } \tilde{A}^{(1)} x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j \preceq \tilde{b}^{(1)} \tag{5.1b}$$

where $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p \times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p \times m_j}$, $i = 1, 2, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j \in Y_j} f(x,y) = \left(\tilde{c}_1^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{1k}^{(2)} y_k, \ldots, \tilde{c}_t^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{tk}^{(2)} y_k \right)^T \tag{5.2a}$$

$$\text{subject to } \tilde{A}^{(2)} x + \sum_{k=1}^{K} \tilde{B}_k^{(2)} y_k \preceq \tilde{b}^{(2)} \tag{5.2b}$$

where $\tilde{c}_i^{(2)} \in F^n(R)$, $\tilde{b}^{(2)} \in F^q(R)$, $\tilde{A}^{(2)} \in M(F(R))_{q \times n}$, $\tilde{B}_k^{(2)} \in M(F(R))_{q \times m_k}$, $\tilde{d}_{ik}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $k = 1, 2, \ldots, K$.

Model 2. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are shared decision variables $y_j (j = 1, 2, \ldots, K)$, the same objective functions $f(x,y)$ but different constraint functions among them, is defined as follows:

For $x \in X \subseteq R^n$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1, 2, \ldots, K$, $y = (y_1, \ldots, y_K) \in Y = (Y_1, \ldots, Y_K)$, $F(x,y) : X \times Y \to F^s(R)$, $f(x,y) : X \times Y \to F^t(R)$, it consists of finding a solution to the upper level problem:

$$\min_{x \in X} F(x,y) = \left(\tilde{c}_1^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j, \ldots, \tilde{c}_s^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j \right)^T \tag{5.3a}$$

$$\text{subject to } \tilde{A}^{(1)} x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j \preceq \tilde{b}^{(1)} \tag{5.3b}$$

where $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p \times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p \times m_j}$,$i = 1, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j \in Y_j, j=1,\ldots,K} f(x,y) = \left(\tilde{c}_1^{(2)}x + \sum_{k=1}^{K} \tilde{d}_{1k}^{(2)} y_k, \ldots, \tilde{c}_t^{(2)}x + \sum_{k=1}^{K} \tilde{d}_{tk}^{(2)} y_k,\right)^T \tag{5.4a}$$

$$\text{subject to } \tilde{A}_j^{(2)}x + \sum_{k=1}^{K} \tilde{B}_{jk}^{(2)} y_k \preceq \tilde{b}_j^{(2)} \tag{5.4b}$$

where $\tilde{c}_i^{(2)} \in F^n(R)$, $\tilde{b}_j^{(2)} \in F^q(R)$, $\tilde{A}_j^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_{jk}^{(2)} \in M(F^{m_k}(R))_{q\times m_k}$, $\tilde{d}_{jk}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots, K$.

Model 3. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are shared decision variables $y_j (j = 1, 2, \ldots, K)$ and constraint functions but different objective functions $f_i(x, y)$ among them, is defined as follows:

For $x \in X \subseteq R^n$, $y_j \in Y_j \subseteq R^{m_j}$ $(j = 1, 2, \ldots, K)$, $y = (y_1, \ldots, y_K) \in Y = (Y_1, \ldots, Y_K)$, $F(x, y) : X \times Y \to F^s(R)$, $f_j(x, y) : X \times Y \to F^t(R)$ $(j = 1, 2, \ldots, K)$, it consists of finding a solution to the upper level problem:

$$\min_{x\in X} F(x,y) = \left(\tilde{c}_1^{(1)}x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j, \ldots, \tilde{c}_s^{(1)}x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j\right)^T \tag{5.5a}$$

$$\text{subject to } \tilde{A}^{(1)}x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j \preceq \tilde{b}^{(1)} \tag{5.5b}$$

where $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p\times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$,$i = 1, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j\in Y_j} f_j(x,y) = \left(\tilde{c}_{j1}^{(2)}x + \sum_{k=1}^{K} \tilde{d}_{j1k}^{(2)} y_k, \ldots, \tilde{c}_{jt}^{(2)}x + \sum_{k=1}^{K} \tilde{d}_{jtk}^{(2)} y_k\right)^T \tag{5.6a}$$

$$\text{subject to } \tilde{A}^{(2)}x + \sum_{k=1}^{K} \tilde{B}_k^{(2)} y_k \preceq \tilde{b}^{(2)} \tag{5.6b}$$

where $\tilde{c}_{ji}^{(2)} \in F^n(R)$, $\tilde{b}^{(2)} \in F^q(R)$, $\tilde{A}^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_k^{(2)} \in M(F(R))_{q\times m_k}$, $\tilde{d}_{jik}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots, K$.

Model 4. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are shared decision variables $y_j (j = 1, 2, \ldots, K)$ but different objective functions $f_i(x, y)$ and constraint functions among them, is defined as follows:

For $x \in X \subseteq R^n$, $y_j \in Y_j \subseteq R^{m_j}$ $(j = 1, 2, \ldots, K)$, $y = (y_1, \ldots, y_K) \in Y = (Y_1, \ldots, Y_K)$, $F(x, y) : X \times Y \to F^s(R)$, $f_j(x, y) : X \times Y \to F^t(R)$ $(j = 1, 2, \ldots, K)$, it consists of finding a solution to the upper level problem:

$$\min_{x \in X} F(x, y) = \left(\tilde{c}_1^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j, \ldots, \tilde{c}_s^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j \right)^T \tag{5.7a}$$

$$\text{subject to } \tilde{A}^{(1)} x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j \preceq \tilde{b}^{(1)} \tag{5.7b}$$

where $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p \times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p \times m_j}$,$i = 1, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j \in Y_j} f_j(x, y) = \left(\tilde{c}_{j1}^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{j1k}^{(2)} y_k, \ldots, \tilde{c}_{jt}^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{jtk}^{(2)} y_k \right)^T \tag{5.8a}$$

$$\text{subject to } \tilde{A}_j^{(2)} x + \sum_{k=1}^{K} \tilde{B}_{jk}^{(2)} y_k \preceq \tilde{b}_j^{(2)} \tag{5.8b}$$

where $\tilde{c}_{ji}^{(2)} \in F^n(R)$, $\tilde{b}_j^{(2)} \in F^q(R)$, $\tilde{A}_j^{(2)} \in M(F(R))_{q \times n}$, $\tilde{B}_{jk}^{(2)} \in M(F^{m_k}(R))_{q \times m_k}$, $\tilde{d}_{jik}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots K$.

Model 5. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are shared objective functions, shared constraint functions and partial shared decision variables among the followers, is defined as follows:

For $x \in X \subseteq R^n$, $z \in Z \subseteq R^h$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1, 2, \ldots, K$, $y = (y_1, \ldots, y_K) \in Y = (Y_1, \ldots, Y_K)$, $(y, z) = (y_1, \ldots, y_K, z) \in Y \times Z$, $F(x, y, z) : X \times Y \times Z \to F^s(R)$, $f(x, y, z) : X \times Y \times Z \to F^t(R)$, and $j = 1, 2, \ldots, K$, it consists of finding a solution to the upper level problem:

$$\min_{x \in X} F(x, y, z) = \left(\tilde{c}_1^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j + \tilde{d}_1^{(1)} z, \ldots, \tilde{c}_s^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j + \tilde{d}_s^{(1)} z \right)^T \tag{5.9a}$$

$$\text{subject to } \tilde{A}^{(1)} x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j + \tilde{B}^{(1)} z \preceq \tilde{b}^{(1)} \tag{5.9b}$$

where $\tilde{b}^{(1)} \in F^p(R)$, $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{d}_i^{(1)} \in F^h(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p \times n}$,

$\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$, $\tilde{B}^{(1)} \in M(F(R))_{p\times h}$, $i = 1,\ldots,s$, $j = 1,2\ldots,K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j\in Y_j, z\in Z} f(x,y,z) = \left(\tilde{c}_1^{(2)}x + \sum_{k=1}^{K}\tilde{d}_{1k}^{(2)}y_k + \tilde{d}_1^{(2)}z, \ldots, \tilde{c}_t^{(2)}x + \sum_{k=1}^{K}\tilde{d}_{tk}^{(2)}y_k + \tilde{d}_t^{(2)}z\right)^T \tag{5.10a}$$

$$\text{subject to } \tilde{A}^{(2)}x + \sum_{k=1}^{K}\tilde{B}_k^{(2)}y_k + \tilde{B}^{(2)}z \preceq \tilde{b}^{(2)} \tag{5.10b}$$

where $\tilde{c}_i^{(2)} \in F^n(R)$, $\tilde{b}^{(2)} \in F^q(R)$, $\tilde{A}^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_k^{(2)} \in M(F(R))_{q\times m_k}$, $\tilde{B}^{(2)} \in M(F(R))_{q\times h}$, $\tilde{d}_{jk}^{(2)} \in F^{m_k}(R)$, $\tilde{d}_i^{(2)} \in F^h(R)$, $i = 1,2,\ldots,t$, $j = 1,2,\ldots,K$.

Model 6. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are shared objective functions and partially shared decision variables but different constraint functions among followers, is defined as follows:

For $x \in X \subseteq R^n$, $z \in Z \subseteq R^h$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1,2,\ldots,K$, $y = (y_1,\ldots,y_K) \in Y = (Y_1,\ldots,Y_K)$, $(y,z) = (y_1,\ldots,y_K,z) \in Y\times Z$, $F(x,y,z) : X\times Y\times Z \to F^s(R)$, $f(x,y,z) : X\times Y\times Z \to F^t(R)$, and $j = 1,2,\ldots,K$, it consists of finding a solution to the upper level problem:

$$\min_{x\in X} F(x,y,z) = \left(\tilde{c}_1^{(1)}x + \sum_{j=1}^{K}\tilde{d}_{1j}^{(1)}y_j + \tilde{d}_1^{(1)}z, \ldots, \tilde{c}_s^{(1)}x + \sum_{j=1}^{K}\tilde{d}_{sj}^{(1)}y_j + \tilde{d}_s^{(1)}z\right)^T \tag{5.11a}$$

$$\text{subject to } \tilde{A}^{(1)}x + \sum_{j=1}^{K}\tilde{B}_j^{(1)}y_j + \tilde{B}^{(1)}z \preceq \tilde{b}^{(1)} \tag{5.11b}$$

where $\tilde{b}^{(1)} \in F^p(R)$, $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{d}_i^{(1)} \in F^h(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p\times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$, $\tilde{B}^{(1)} \in M(F(R))_{p\times h}$, $i = 1,\ldots,s$, $j = 1,2\ldots,K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j\in Y_j, z\in Z} f(x,y,z) = \left(\tilde{c}_1^{(2)}x + \sum_{k=1}^{K}\tilde{d}_{1k}^{(2)}y_k + \tilde{d}_1^{(2)}z, \ldots, \tilde{c}_t^{(2)}x + \sum_{k=1}^{K}\tilde{d}_{tk}^{(2)}y_k + \tilde{d}_t^{(2)}z\right)^T \tag{5.12a}$$

$$\text{subject to } \tilde{A}_j^{(2)}x + \sum_{k=1}^{K}\tilde{B}_{jk}^{(2)}y_k + \tilde{B}_j^{(2)}z \preceq \tilde{b}_j^{(2)} \tag{5.12b}$$

where $\tilde{c}_i^{(2)} \in F^n(R)$, $\tilde{b}^{(2)} \in F^q(R)$, $\tilde{A}_j^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_{jk}^{(2)} \in M(F(R))_{q\times m_k}$, $\tilde{B}_j^{(2)} \in M(F(R))_{q\times h}$, $\tilde{d}_{jk}^{(2)} \in F^{m_k}(R)$, $\tilde{d}_i^{(2)} \in F^h(R)$, $i = 1, 2, \ldots, t$, $j = 1, 2, \ldots, K$.

Model 7. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are partially shared decision variables and shared constraint functions but different objective functions among them, is defined as follows:

For $x \in X \subseteq R^n$, $z \in Z \subseteq R^h$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1, 2, \ldots, K$, $y = (y_1, \ldots, y_K) \in Y = (Y_1, \ldots, Y_K)$, $(y, z) = (y_1, \ldots, y_K, z) \in Y \times Z$, $F(x, y, z) : X \times Y \times Z \to F^s(R)$, $f(x, y, z) : X \times Y \times Z \to F^t(R)$, and $j = 1, 2, \ldots, K$, it consists of finding a solution to the upper level problem:

$$\min_{x\in X} F(x, y, z) = \left(\tilde{c}_1^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j + \tilde{d}_1^{(1)} z, \ldots, \tilde{c}_s^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j + \tilde{d}_s^{(1)} z \right)^T \tag{5.13a}$$

$$\text{subject to } \tilde{A}^{(1)} x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j + \tilde{B}^{(1)} z \preceq \tilde{b}^{(1)} \tag{5.13b}$$

where $\tilde{b}^{(1)} \in F^p(R)$, $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{d}_i^{(1)} \in F^h(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p\times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$, $\tilde{B}^{(1)} \in M(F(R))_{p\times h}$, $i = 1, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j\in Y_j, z\in Z} f_j(x, y, z) = \left(\tilde{c}_{j1}^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{j1k}^{(2)} y_k + \tilde{d}_{j1}^{(2)} z, \ldots, \tilde{c}_t^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{jtk}^{(2)} y_k + \tilde{d}_{jt}^{(2)} z \right)^T \tag{5.14a}$$

$$\text{subject to } \tilde{A}^{(2)} x + \sum_{k=1}^{K} \tilde{B}_k^{(2)} y_k + \tilde{B}^{(2)} z \preceq \tilde{b}^{(2)} \tag{5.14b}$$

where $\tilde{c}_{ji}^{(2)} \in F^n(R)$, $\tilde{b}^{(2)} \in F^q(R)$, $\tilde{A}^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_k^{(2)} \in M(F(R))_{q\times m_k}$, $\tilde{B}^{(2)} \in M(F(R))_{q\times h}$, $\tilde{d}_{ji}^{(2)} \in F^h(R)$, $\tilde{d}_{jik}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $j = 1, 2, \ldots, K$.

Model 8. An FMMLB problem, in which $K \geqslant 2$ followers are involved and there are partially shared decision variables but different objective and constraint functions among them, is defined as follows:

For $x \in X \subseteq R^n$, $z \in Z \subseteq R^h$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1, 2, \ldots, K$, $y = (y_1, \ldots, y_K) \in Y = (Y_1, \ldots, Y_K)$, $(y, z) = (y_1, \ldots, y_K, z) \in Y \times Z$, $F(x, y, z) : X \times Y \times Z \to F^s(R)$, $f(x, y, z) : X \times Y \times Z \to F^t(R)$, and $j = 1, 2, \ldots, K$, it consists of finding a solution to the upper level problem:

$$\min_{x\in X} F(x,y,z) = \left(\tilde{c}_1^{(1)}x + \sum_{j=1}^{K}\tilde{d}_{1j}^{(1)}y_j + \tilde{d}_1^{(1)}z, \ldots, \tilde{c}_s^{(1)}x + \sum_{j=1}^{K}\tilde{d}_{sj}^{(1)}y_j + \tilde{d}_s^{(1)}z\right)^T \tag{5.15a}$$

$$\text{subject to } \tilde{A}^{(1)}x + \sum_{j=1}^{K}\tilde{B}_j^{(1)}y_j + \tilde{B}^{(1)}z \preceq \tilde{b}^{(1)} \tag{5.15b}$$

where $\tilde{b}^{(1)} \in F^p(R)$, $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{d}_i^{(1)} \in F^h(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p\times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$, $\tilde{B}^{(1)} \in M(F(R))_{p\times h}$, $i = 1,\ldots,s$, $j = 1,2\ldots,K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j\in Y_j, z\in Z} f_j(x,y,z) = \left(\tilde{c}_{j1}^{(2)}x + \sum_{k=1}^{K}\tilde{d}_{j1k}^{(2)}y_k + \tilde{d}_{j1}^{(2)}z, \ldots, \tilde{c}_t^{(2)}x + \sum_{k=1}^{K}\tilde{d}_{jtk}^{(2)}y_k + \tilde{d}_{jt}^{(2)}z\right)^T \tag{5.16a}$$

$$\text{subject to } \tilde{A}_j^{(2)}x + \sum_{k=1}^{K}\tilde{B}_{jk}^{(2)}y_k + \tilde{B}_j^{(2)}z \preceq \tilde{b}_j^{(2)} \tag{5.16b}$$

where $\tilde{c}_{ji}^{(2)} \in F^n(R)$, $\tilde{b}_j^{(2)} \in F^q(R)$, $\tilde{A}_j^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_{jk}^{(2)} \in M(F(R))_{q\times m_k}$, $\tilde{B}_j^{(2)} \in M(F(R))_{q\times h}$, $\tilde{d}_{ji}^{(2)} \in F^h(R)$, $\tilde{d}_{jik}^{(2)} \in F^{m_k}(R)$, $i = 1,2,\ldots,t$, $k = 1,2,\ldots,K$, $j = 1,2,\ldots,K$.

By analysing above eight models and using a weighting method, we can get a general model (Model G) for FMMLB decision problems:

Definition 5.1.1. For $x \in X \subseteq R^n$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1,2,\ldots,K$, $y = (y_1,\ldots,y_K) \in Y = (Y_1,\ldots,Y_K)$, $F(x,y) : X \times Y \to F^s(R)$, $f(x,y) : X \times Y \to F^t(R)$, it consists of finding a solution to the upper level problem:

$$\min_{x\in X} F(x,y) = \left(\tilde{c}_1^{(1)}x + \sum_{j=1}^{K}\tilde{d}_{1j}^{(1)}y_j, \ldots, \tilde{c}_s^{(1)}x + \sum_{j=1}^{K}\tilde{d}_{sj}^{(1)}y_j\right)^T \tag{5.17a}$$

$$\text{subject to } \tilde{A}^{(1)}x + \sum_{j=1}^{K}\tilde{B}_j^{(1)}y_j \preceq \tilde{b}^{(1)} \tag{5.17b}$$

where $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p\times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$,$i =$

$1, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j \in Y_j, j=1,\ldots,K} f(x,y) = \left(\tilde{c}_1^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{1k}^{(2)} y_k, \ldots, \tilde{c}_t^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{tk}^{(2)} y_k, \right)^T \tag{5.18a}$$

$$\text{subject to } \tilde{A}_j^{(2)} x + \sum_{k=1}^{K} \tilde{B}_{jk}^{(2)} y_k \preceq \tilde{b}_j^{(2)} \tag{5.18b}$$

where $\tilde{c}_i^{(2)} \in F^n(R)$, $\tilde{b}_j^{(2)} \in F^q(R)$, $\tilde{A}_j^{(2)} \in M(F(R))_{q\times n}$, $\tilde{B}_{jk}^{(2)} \in M(F^{m_k}(R))_{q\times m_k}$, $\tilde{d}_{jk}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots, K$.

Therefore, we have found that:

(1) Model 1 is a special issue of Model G when $\tilde{A}_j^{(2)} = \tilde{A}^{(2)}$, $\tilde{B}_{jk}^{(2)} = \tilde{B}_k^{(2)}$, $\tilde{b}_j^{(2)} = \tilde{b}^{(2)}$, $j = 1, 2, \ldots, K$.

(2) Model 2 has the same format with Model G.

(3) For Model 3, we know that it is a multi-objective programming for followers and for $j = 1, 2, \ldots, K$, $\tilde{A}_j^{(2)} = \tilde{A}^{(2)}$, $\tilde{B}_{jk}^{(2)} = \tilde{B}_k^{(2)}$, $\tilde{b}_j^{(2)} = \tilde{b}^{(2)}$. We can transform it to Model G by using a weighting method, i.e., $f(x,y), \tilde{c}_r^{(2)}, \tilde{d}_{rk}^{(2)}$ in Model G is equal to $f_j(x,y)$, $\tilde{c}_{jr}^{(2)}$,$\tilde{d}_{jrk}^{(2)}$ in Model 3, $r = 1, 2, \ldots, t$, respectively.

(4) For Model 4, We can transform it to Model G by using a weighting method, i.e. $f(x,y), \tilde{c}_r^{(2)}, \tilde{d}_{rk}^{(2)}$ in Model G is equal to $f_j(x,y)$, $\tilde{c}_{jr}^{(2)}$,$\tilde{d}_{jrk}^{(2)}$ in Model 3, $r = 1, 2, \ldots, t$, respectively.

(5) Models 5, 6 and 7 are special cases of Model 8, respectively.

(6) Model 8 is an FMMLB decision problem in which K followers share variable z. By using a weighting method, we can obtain:

For $x \in X \subseteq R^n$, $z \in Z \subseteq R^h$, $y_j \in Y_j \subseteq R^{m_j}$, $j = 1, 2, \ldots, K$, $y = (y_1, \ldots, y_K) \in Y = ((Y_1, Z), \ldots, (Y_K, Z))$, $F(x,y,z) : X \times Y \times Z \to F^s(R)$, $f(x,y,z) : X \times Y \times Z \to F^t(R)$, and $j = 1, 2, \ldots, K$, it consists of finding a solution to the upper level problem:

$$\min_{x \in X} F(x,y,z) = \left(\tilde{c}_1^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{1j}^{(1)} y_j + \tilde{d}_1^{(1)} z, \ldots, \tilde{c}_s^{(1)} x + \sum_{j=1}^{K} \tilde{d}_{sj}^{(1)} y_j + \tilde{d}_s^{(1)} z \right)^T \tag{5.19a}$$

$$\text{subject to } \tilde{A}^{(1)} x + \sum_{j=1}^{K} \tilde{B}_j^{(1)} y_j + \tilde{B}^{(1)} z \preceq \tilde{b}^{(1)} \tag{5.19b}$$

where $\tilde{b}^{(1)} \in F^p(R)$, $\tilde{c}_i^{(1)} \in F^n(R)$, $\tilde{d}_{ij}^{(1)} \in F^{m_j}(R)$, $\tilde{d}_i^{(1)} \in F^h(R)$, $\tilde{A}^{(1)} \in M(F(R))_{p\times n}$, $\tilde{B}_j^{(1)} \in M(F(R))_{p\times m_j}$, $\tilde{B}^{(1)} \in M(F(R))_{p\times h}$, $i = 1, \ldots, s$, $j = 1, 2 \ldots, K$ and y_j, for each value of x, is the solution of the lower level problem:

$$\min_{y_j \in Y_j, z \in Z} f_j(x,y,z) = \left(\tilde{c}_{j1}^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{j1k}^{(2)} y_k + \tilde{d}_{j1}^{(2)} z, \ldots, \tilde{c}_t^{(2)} x + \sum_{k=1}^{K} \tilde{d}_{jtk}^{(2)} y_k + \tilde{d}_{jt}^{(2)} z \right)^T \tag{5.20a}$$

$$\text{subject to } \tilde{A}_j^{(2)} x + \sum_{k=1}^{K} \tilde{B}_{jk}^{(2)} y_k + \tilde{B}_j^{(2)} z \preceq \tilde{b}_j^{(2)} \tag{5.20b}$$

where $\tilde{c}_{ji}^{(2)} \in F^n(R), \tilde{b}_j^{(2)} \in F^q(R), \tilde{A}_j^{(2)} \in M(F(R))_{q\times n}, \tilde{B}_{jk}^{(2)} \in M(F(R))_{q\times m_k}$, $\tilde{B}_j^{(2)} \in M(F(R))_{q\times h}$, $\tilde{d}_{ji}^{(2)} \in F^h(R)$, $\tilde{d}_{jik}^{(2)} \in F^{m_k}(R)$, $i = 1, 2, \ldots, t$, $k = 1, 2, \ldots, K$, $j = 1, 2, \ldots, K$.

Model 8 has the same solution as Model G because it can be transformed into Model G.

(7) We only need to develop algorithms to solve Model G. Through some transformation as discussed above, all these eight kinds FMMLB decision problems can be solved then.

5.2 Algorithms

5.2.1 An Approximation Branch-and-Bound-based Algorithm

We first present related theorems.

Theorem 5.2.1. *(Zhang, Lu & Dillon 2007d) For $x \in X \subseteq R^n$, $y_i \in Y_i \subseteq R^{m_i}$, $i = 1, 2, \ldots, K$, if all the fuzzy coefficients in the fuzzy bi-level problem problem defined by*

Definition 5.1.1 have membership functions :

$$\mu_{\tilde{z}}(t) = \begin{cases} 0 & t < \alpha^L_{\lambda_0} \\ \frac{\lambda_1 - \lambda_0}{\alpha^L_{\lambda_1} - \alpha^L_{\lambda_0}} \left(t - \alpha^L_{\lambda_0}\right) + \lambda_0 & \alpha^L_{\lambda_0} \leqq t < \alpha^L_{\lambda_1} \\ \frac{\lambda_2 - \lambda_1}{\alpha^L_{\lambda_2} - \alpha^L_{\lambda_1}} \left(t - \alpha^L_{\lambda_1}\right) + \lambda_1 & \alpha^L_{\lambda_1} \leqq t < \alpha^L_{\lambda_2} \\ \cdots & \cdots \\ \lambda & \alpha^L_{\lambda_l} \leqq t < \alpha^R_{\lambda_l} \\ \frac{\lambda_l - \lambda_{l-1}}{\alpha^R_{\lambda_{l-1}} - \alpha^R_{\lambda_l}} \left(-t + \alpha^R_{\lambda_{l-1}}\right) + \lambda_{l-1} & \alpha^R_{\lambda_l} \leqq t < \alpha^R_{\lambda_{l-1}} \\ \cdots & \cdots \\ \frac{\lambda_0 - \lambda_1}{\alpha^R_{\lambda_1} - \alpha^R_{\lambda_0}} \left(-t + \alpha^R_{\lambda_0}\right) + \lambda_0 & \alpha^R_{\lambda_1} \leqq t \leqq \alpha^R_{\lambda_0} \\ 0 & \alpha^R_{\lambda_0} < t \end{cases}, \tag{5.21}$$

where $\tilde{z}$ denotes any fuzzy coefficients in model defined by Definition 5.1.1, then, it is the solution of the problem defined by Definition 5.1.1 that $(x^*, y^*) \in R^n \times R^m$ satisfying

$$\begin{aligned}
&\min_{x \in X} F(x, y_1, y_2, \ldots, y_K)^L_{\lambda_j} = \tilde{c}^L_{\lambda_j} x + \textstyle\sum_{s=1}^K \tilde{d}_{s\lambda_j}^{\ L} y_s \\
&\min_{x \in X} F(x, y_1, y_2, \ldots, y_K)^R_{\lambda_j} = \tilde{c}^R_{\lambda_j} x + \textstyle\sum_{s=1}^K \tilde{d}_{s\lambda_j}^{\ R} y_s \\
&\textit{subject to } \tilde{A}^L_{\lambda_j} x + \textstyle\sum_{t=1}^K \tilde{B}_{t\lambda_j}^{\ L} y_t \leqslant \tilde{b}^L_{\lambda_j}, \\
&\qquad \tilde{A}^R_{\lambda_j} x + \textstyle\sum_{t=1}^K \tilde{B}_{t\lambda_j}^{\ R} y_t \leqslant \tilde{b}^R_{\lambda_j}, \\
&\qquad \min_{y_i \in Y_i, i=1,\ldots,K} f(x, y_1, y_2, \ldots, y_K)^L_{\lambda_j} = \textstyle\sum_{i=1}^K \tilde{c}_{i\lambda_j}^{\ L} x + \sum_{i=1}^K \tilde{e}_{i\lambda_j}^{\ L} y_i \\
&\qquad \min_{y_i \in Y_i, i=1,\ldots,K} f(x, y_1, y_2, \ldots, y_K)^R_{\lambda_j} = \textstyle\sum_{i=1}^K \tilde{c}_{i\lambda_j}^{\ R} x + \sum_{i=1}^K \tilde{e}_{i\lambda_j}^{\ R} y_i \\
&\qquad \textit{subject to } \tilde{A}_{i\lambda_j}^{\ L} x + \tilde{C}_{i\lambda_j}^{\ L} y_i \leqslant \tilde{b}_{i\lambda_j}^{\ L}, \\
&\qquad \tilde{A}_{i\lambda_j}^{\ R} x + \tilde{C}_{i\lambda_j}^{\ R} y_i \leqslant \tilde{b}_{i\lambda_j}^{\ R}, \\
&\qquad i = 1, 2, \ldots, K, \\
&\qquad j = 0, 1, 2, \ldots, l.
\end{aligned} \tag{5.22}$$

where λ_j represents a certain cut set, $\lambda_j \in [0, 1]$.

Problem (5.22) can be further transferred into the following linear bi-level problem by a weighting method:

$$
\begin{aligned}
&\min_{x\in X} F(x,y_1,y_2,\ldots,y_K)_{\lambda_j} = \tilde{c}^L_{\lambda_j}x + \textstyle\sum_{s=1}^K \tilde{d}^{\,L}_{s\lambda_j}y_s + \tilde{c}^R_{\lambda_j}x + \textstyle\sum_{s=1}^K \tilde{d}^{\,R}_{s\lambda_j}y_s\\
&\text{subject to } \tilde{A}^L_{\lambda_j}x + \textstyle\sum_{t=1}^K \tilde{B}^{\,L}_{t\lambda_j}y_t \leqslant \tilde{b}^L_{\lambda_j},\\
&\qquad \tilde{A}^R_{\lambda_j}x + \textstyle\sum_{t=1}^K \tilde{B}^{\,R}_{t\lambda_j}y_t \leqslant \tilde{b}^R_{\lambda_j},\\
&\qquad \min_{y_i\in Y_i, i=1,\ldots,K} f(x,y_1,y_2,\ldots,y_K)_{\lambda_j} = \textstyle\sum_{i=1}^K \tilde{c}^{\,L}_{i\lambda_j}x + \textstyle\sum_{i=1}^K \tilde{e}^{\,L}_{i\lambda_j}y_i\\
&\qquad\qquad + \textstyle\sum_{i=1}^K \tilde{c}^{\,R}_{i\lambda_j}x + \textstyle\sum_{i=1}^K \tilde{e}^{\,R}_{i\lambda_j}y_i\\
&\qquad \text{subject to } \tilde{A}^{\,L}_{i\lambda_j}x + \tilde{C}^{\,L}_{i\lambda_j}y_i \leqslant \tilde{b}^{\,L}_{i\lambda_j},\\
&\qquad\qquad \tilde{A}^{\,R}_{i\lambda_j}x + \tilde{C}^{\,R}_{i\lambda_j}y_i \leqslant \tilde{b}^{\,R}_{i\lambda_j},\\
&\qquad\qquad i = 1,2,\ldots,K,\\
&\qquad\qquad j = 0,1,2,\ldots,l.
\end{aligned}
\tag{5.23}
$$

Theorem 5.2.2. *(Zhang et al. 2007d) For $x \in X \subseteq R^n$, $y_i \in Y_i \subseteq R^{m_i}$, $i = 1,2,\ldots,K$, a necessary and sufficient condition that x^*, y^* is a solution of problem defined by Definition 5.1.1 is that there exist (row) vectors u^*, v^*, z^* such that $(x^*, y^*, u^*, v^*, z^*)$ is a solution of :*

$$\min_{x\in X} F(x,y_1,y_2,\ldots,y_K) = \sum_{j=1}^{l}(\tilde{c}^L_{\lambda_j}x + \sum_{s=1}^{K}\tilde{d}^{\,L}_{s\lambda_j}y_s + \tilde{c}^R_{\lambda_j}x + \sum_{s=1}^{K}\tilde{d}^{\,R}_{s\lambda_j}y_s) \tag{5.24a}$$

$$
\begin{aligned}
\textit{subject to } &\tilde{A}^L_{\lambda_j}x + \sum_{t=1}^{K}\tilde{B}^{\,L}_{t\lambda_j}y_t \leqslant \tilde{b}^L_{\lambda_j},\\
&\tilde{A}^R_{\lambda_j}x + \sum_{t=1}^{K}\tilde{B}^{\,R}_{t\lambda_j}y_t \leqslant \tilde{b}^R_{\lambda_j}, \qquad (5.24b)\\
&\tilde{A}^{\,L}_{i\lambda_j}x + \tilde{C}^{\,L}_{i\lambda_j}y_i \leqslant \tilde{b}^{\,L}_{i\lambda_j},\\
&\tilde{A}^{\,R}_{i\lambda_j}x + \tilde{C}^{\,R}_{i\lambda_j}y_i \leqslant \tilde{b}^{\,R}_{i\lambda_j},\\
&i = 1,2,\ldots,K, j = 0,1,2,\ldots,l,\\
&u(\sum_{j=1}^{l}\sum_{t=1}^{K}(\tilde{B}^{\,L}_{t\lambda_j} + \tilde{B}^{\,R}_{t\lambda_j})) + v(\sum_{j=1}^{l}\sum_{i=1}^{K}(\tilde{C}^{\,L}_{i\lambda_j} + \tilde{C}^{\,R}_{i\lambda_j})) - z \qquad (5.24c)\\
&\qquad = -\sum_{j=1}^{l}\sum_{i=1}^{K}(\tilde{e}^{\,L}_{i\lambda_j} + \tilde{e}^{\,R}_{i\lambda_j}),\\
&u(\sum_{j=1}^{l}(\tilde{b}^L_{\lambda_j} + \tilde{b}^R_{\lambda_j} - \tilde{A}^L_{\lambda_j}x - \sum_{s=1}^{K}\tilde{B}^{\,L}_{s\lambda_j}y_t - \tilde{A}^R_{\lambda_j}x - \sum_{s=1}^{K}\tilde{B}^{\,R}_{s\lambda_j}y_t)) \qquad (5.24d)
\end{aligned}
$$

$$+ v(\sum_{j=1}^{l}\sum_{i=1}^{K}(\tilde{b}^{L}_{i\lambda_j} + \tilde{b}^{R}_{i\lambda_j}) - \tilde{A}^{L}_{i\lambda_j}x - \tilde{C}^{L}_{i\lambda_j}y_i - \tilde{A}^{R}_{i\lambda_j}x - \tilde{C}^{R}_{i\lambda_j}y_i) + z\sum_{i=1}^{k} y_i = 0.$$

Based on the extended Branch-and-Bound algorithm (Lu *et al.* 2007b), we extend it to fuzzy situations and propose an approximation Branch-and-Bound-based algorithm for solving FMMLB decision problems in this section.

We first write all the inequalities (except of the leader's variables) of (5.23) as $g_i(x, y) \geqslant 0$, $i = 1, \ldots, p + q + m$, and note that complementary slackness simply means $u_i g_i(x, y) = 0$ ($i = 1, \ldots, p+q+m$). We suppress the complementary term and solve the resulted linear sub-problem. At each time of iteration the condition (5.24d) is checked. If it is satisfied, the corresponding point is in the inducible region and hence a potential solution to (5.23). Otherwise, a Branch-and-Bound scheme is used to implicitly examine all combinations of the complementarities slackness.

Now, we give some notations for describing the details of the approximation Branch-and-Bound-based algorithm.

Let $W = \{1, \ldots, p + q + m\}$ be the index set for the terms in (5.24d), $\bar{F}$ be the incumbent upper bound on the objective function of the leader. At the $k-$th level of a search tree we define a subset of indices $W_k \subseteq W$, and a path P_k corresponding to an assignment of either $u_i = 0$ or $g_i = 0$ for $i \in W_k$. Now let

$$\begin{aligned} S_k^+ &= \{i : i \in W_k, u_i = 0\} \\ S_k^- &= \{i : i \in W_k, g_i = 0\} \\ S_k^0 &= \{i : i \notin W_k\}. \end{aligned}$$

For $i \in S_k^0$, the variables u_i or g_i are free to assume any nonnegative value in the solution of (5.24) with (5.24d) omitted, so complementary slackness will not necessarily be satisfied.

By using these notations we give all steps of the approximation Branch-and-Bound-based algorithm in Algorithm 3.

We give some explanations for these steps and their working process as follows.

After initialisation, Step 7 will find a new point which is potentially bi-level feasible. If no solution exists, or the solution does not offer an improvement over the incumbent (Step 8), the algorithm goes to Step 11 and backtracks.

Step 9 checks the value of $u_i^k g_i(x^k, y^k)$to determine if the complementary slackness conditions are satisfied. In practice, if $\left|u_i^k g_i\right| < 10^{-6}$ it is considered to be zero. Confirmation indicates that a feasible solution of a bi-level program has been found

Algorithm 3: An approximation Branch-and-Bound-based algorithm for FMMLB decision problems

Step 1: Given two sets of weights w_{j1} $(j = 1, 2, \ldots, s)$ and w_{j2} $(j = 1, 2, \ldots, t)$ to the objectives of the leader and the follower respectively, and let $\sum_{j=1}^{s} w_{j1} = 1$ and $\sum_{j=1}^{t} w_{j2} = 1$.
Step 2: The problem defined by Definition 5.1.1 is transformed to problem (5.23).
Step 3: Set $l = 1$, a range of errors $\epsilon > 0$, using the extended Branch-and-Bound algorithm (Shi *et al.* 2006) to solve problem (5.23) under current $\lambda-$ cut.
Step 4: Decompose interval $[0, 1]$ into 2^{l-1} equal sub-intervals with $(2^{l-1} + 1)$ nodes λ_i $(i = 0, \cdots, 2^{l-1})$ arranged in the order of $0 = \lambda_0 < \cdots < \lambda_{2^{l-1}} = 1$.
Step 5: Transform problem (5.23) to problem (5.24) by using Theorem 5.2.1 and a weighting method (Bialas & Karwan 1978).
Step 6: (Initialisation) Set $k = 0$, $S_k^+ = \phi$, $S_k^- = \phi$, $S_k^0 = \{1, \ldots, p + q + m\}$, and $\bar{F} = \infty$.
Step 7: (Iteration k) Set $u_i = 0$ for $i \in S_k^+$ and $g_i = 0$ for $i \in S_k^-$. It first attempts to solve (5.24) without (5.24d). If the resultant problem is infeasible, go to Step 11; otherwise, put $k \leftarrow k + 1$ and label the solution as (x^k, y^k, u^k).
Step 8: (Fathoming) If $F(x^k, y^k) \geqslant \bar{F}$, then go to Step 11.
Step 9: (Branching) If $u_i^k g_i(x^k, y^k) = 0$, $i = 1, \ldots, p + q + m$, then go to Step 10. Otherwise select i for which $u_i^k g_i(x^k, y^k) \neq 0$ is the largest and label it i_1. Put $S_k^+ \leftarrow S_k^+ \cup \{i_1\}$, $S_k^0 \leftarrow S_k^0 \setminus \{i_1\}$, $S_k^- \leftarrow S_k^-$, append i_1 to P_k, and go to Step 7.
Step 10: (Updating) Let $\bar{F} \leftarrow F(x^k, y^k)$.
Step 11: (Backtracking) If no live node exists, go to Step 12. Otherwise branch to the newest live vertex and update S_k^+, S_k^-, S_k^0 and P_k as discussed below. Go back to Step 7.
Step 12: (Termination) If $\bar{F} = \infty$, there is not feasible solution to the current problem. Otherwise, declare the feasible point associated with $\bar{F}$ which is the optimal solution $(x, y)_{2^l}$ to the current problem.
Step 13: $l = l + 1$, repeat Step 4 to Step 12.
Step 14: If $\|(x, y)_{2^{l+1}} - (x, y)_{2^l}\| < \varepsilon$, then the solution (x^*, y^*) of the current problem is $(x, y)_{2^{l+1}}$, otherwise, go back to Step 13.
Step 15: Show the result. Terminate.

and at Step 10 the upper bound on the leader's objective function is updated. Alternatively, if the complementary slackness conditions are not satisfied, the term with the largest product is used at Step 9 to provide a branching variable. Branching is always completed on the Kuhn-Tucker multiplier (Bard 1998).

At Step 11, the backtracking operation is performed. Note that a live node is one associated with a sub-problem that has not yet been fathomed at either Step 7 due to infeasibility or at Step 8 due to bounding, and whose solution violates at least one complementary slackness condition. To facilitate book keeping, the path P_k in the Branch-and-Bound tree is represented by a vector, its dimension is the current depth of the tree. The order of the components of P_k is determined by their level in the tree. Indices only appear in P_k if they are in either S_k^+ or S_k^- with the entries underlined if they are in S_k^-. Because the algorithm always branches on a Kuhn-Tucker multiplier first, backtracking is accomplished by finding the rightmost non-underlined component if P_k, underlining it, and erasing all entries to the right. The erased entries are deleted from S_k^- and added to S_k^0.

We apply the proposed approximation Branch-and-Bound-based algorithm to solve a simple FMMLB decision problem to illustrate how the algorithm is used.

In this example, the leader has two objectives F_1 and F_2. There are two followers, who share decision variables and constraints but have one individual objective ($f1_1$ for the first follower and $f2_1$ for the second follower). We can see that this FMMLB decision problem exactly falls into the category of Model 3.

Example 5.2.1. Consider the following FMMLB decision problem with $x \in R^1$, $y \in R^1$, and $X = \{x \geqslant 0\}$, $Y = \{y \geqslant 0\}$,

$$
\begin{aligned}
&\min_{x\in X} F_1(x,y) = -\tilde{1}x + \tilde{2}y \\
&\min_{x\in X} F_2(x,y) = \tilde{2}x - \tilde{4}y \\
&\text{subject to } -\tilde{1}x + \tilde{3}y \leqslant \tilde{4} \\
&\qquad \min_{y\in Y} f1_1(x,y) = -\tilde{1}x + \tilde{2}y \\
&\qquad \min_{y\in Y} f2_1(x,y) = \tilde{2}x - \tilde{1}y \\
&\qquad \text{subject to } \tilde{1}x - \tilde{1}y \leqslant \tilde{0} \\
&\qquad\qquad\qquad -\tilde{1}x - \tilde{1}y \leqslant \tilde{0}
\end{aligned}
$$

where

$$
\mu_{\tilde{1}}(t) = \begin{cases} 0, & t < 0, \\ t^2, & 0 \leqq t < 1, \\ 2-t, & 1 \leqq t < 2, \\ 0, & 2 \leqq t. \end{cases}
\qquad
\mu_{\tilde{2}}(t) = \begin{cases} 0, & t < 1, \\ t-1, & 1 \leqq t < 2, \\ 3-t, & 2 \leqq t < 3, \\ 0, & 3 \leqq t. \end{cases}
$$

$$
\mu_{\tilde{3}}(t) = \begin{cases} 0, & t < 2, \\ t-2, & 2 \leqq t < 3, \\ 4-t, & 3 \leqq t < 4, \\ 0, & 4 \leqq t. \end{cases}
\qquad
\mu_{\tilde{4}}(t) = \begin{cases} 0, & t < 3, \\ t-3, & 3 \leqq t < 4, \\ 5-t, & 4 \leqq t < 5, \\ 0, & 5 \leqq t. \end{cases}
$$

$$\mu_{\tilde{0}}(t) = \begin{cases} 0, & t < -1, \\ t+1, & -1 \leqq t < 0, \\ 1-t^2, & 0 \leqq t < 1, \\ 0, & 1 \leqq t. \end{cases}$$

We now solve this problem by using the proposed approximation Branch-and-Bound-based algorithm.

Step 1. Set weights (0.5, 0.5) for the two fuzzy objectives of the leader and of the follower respectively.

Step 2. The FMMLB decision problem is transformed to the following MOLB decision problem by using Theorem 5.2.1.

$$\begin{aligned}
&\min_{x\in X}(F_1(x,y))^L_\lambda = (-\tilde{1})^L_\lambda x + \tilde{2}^L_\lambda y, \quad \lambda \in [0,1] \\
&\min_{x\in X}(F_1(x,y))^R_\lambda = (-\tilde{1})^R_\lambda x + \tilde{2}^R_\lambda y, \quad \lambda \in [0,1] \\
&\min_{x\in X}(F_2(x,y))^L_\lambda = \tilde{2}^L_\lambda x + (-\tilde{4})^L_\lambda y, \quad \lambda \in [0,1] \\
&\min_{x\in X}(F_2(x,y))^R_\lambda = \tilde{2}^R_\lambda x + (-\tilde{4})^R_\lambda y, \quad \lambda \in [0,1] \\
&\text{subject to } (-\tilde{1})^L_\lambda x + \tilde{3}^L_\lambda y \leqq \tilde{4}^L_\lambda, (-\tilde{1})^R_\lambda x + \tilde{3}^R_\lambda y \leqq \tilde{4}^R_\lambda, \lambda \in [0,1] \\
&\qquad \min_{y\in Y}(f1_1(x,y))^L_\lambda = \tilde{2}^L_\lambda x + (-\tilde{1})^L_\lambda y, \quad \lambda \in [0,1] \\
&\qquad \min_{y\in Y}(f1_1(x,y))^R_\lambda = \tilde{2}^R_\lambda x + (-\tilde{1})^R_\lambda y, \quad \lambda \in [0,1] \\
&\qquad \min_{y\in Y}(f2_1(x,y))^L_\lambda = (-\tilde{1})^L_\lambda x + \tilde{2}^L_\lambda y, \quad \lambda \in [0,1] \\
&\qquad \min_{y\in Y}(f2_1(x,y))^R_\lambda = (-\tilde{1})^R_\lambda x + \tilde{2}^R_\lambda y, \quad \lambda \in [0,1] \\
&\qquad \text{subject to } \tilde{1}^L_\lambda x + (-\tilde{1})^L_\lambda y \leqq \tilde{0}^L_\lambda, \tilde{1}^R_\lambda x + (-\tilde{1})^R_\lambda y \leqq \tilde{0}^R_\lambda, \quad \lambda \in [0,1] \\
&\qquad\qquad (-\tilde{1})^L_\lambda x + (-\tilde{1})^L_\lambda y \leqq \tilde{0}^L_\lambda,\ (-\tilde{1})^R_\lambda x + (-\tilde{1})^R_\lambda y \leqq \tilde{0}^R_\lambda, \quad \lambda \in [0,1]
\end{aligned}$$

Step 3. Let $l = 1$ and $\varepsilon = 10^{-6}$.

Step 4. Decompose interval $[0,1]$ into 2^{l-1} equal sub-intervals with $(2^{l-1}+1)$ nodes λ_i, $(i = 0, \cdots, 2^{l-1})$ which is arranged in the order of $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{2^{l-1}} = 1$. We now need to solve the following MOLB decision problem

$$\begin{aligned}
&\min_{x\in X}(F_1(x,y))^{L(R)}_1 = -1x + 2y \\
&\min_{x\in X}(F_1(x,y))^L_0 = -2x + y
\end{aligned}$$

$$
\begin{aligned}
&\min_{x\in X}(F_1(x,y))_0^R = 0x + 3y\\
&\min_{x\in X}(F_2(x,y))_1^{L(R)} = 2x - 4y\\
&\min_{x\in X}(F_2(x,y))_0^L = 1x - 5y\\
&\min_{x\in X}(F_2(x,y))_0^R = 3x - 3y\\
&\text{subject to } -1x + 3y \leqslant 4\\
&\qquad\qquad -2x + 2y \leqslant 3\\
&\qquad\qquad 0x + 4y \leqslant 5\\
&\qquad\qquad \min_{y\in Y}(f1_1(x,y))_1^{L(R)} = 2x - 1y\\
&\qquad\qquad \min_{y\in Y}(f1_1(x,y))_0^L = 1x - 2y\\
&\qquad\qquad \min_{y\in Y}(f1_1(x,y))_0^R = 3x - 0y\\
&\qquad\qquad \min_{y\in Y}(f2_1(x,y))_1^{L(R)} = -1x + 2y\\
&\qquad\qquad \min_{y\in Y}(f2_1(x,y))_0^L = -2x + 1y\\
&\qquad\qquad \min_{y\in Y}(f2_1(x,y))_0^L = 0x + 3y\\
&\qquad\qquad \text{subject to } 1x - 1y \leqslant 0\\
&\qquad\qquad\qquad\qquad 0x - 2y \leqslant -1\\
&\qquad\qquad\qquad\qquad 2x - 0y \leqslant 1\\
&\qquad\qquad\qquad\qquad -1x - 1y \leqslant 0\\
&\qquad\qquad\qquad\qquad -2x - 2y \leqslant -1.
\end{aligned}
$$

Step 5. We transform this MOLB problem to a linear bi-level problem by using a weighting method, and we have:

$$
\begin{aligned}
&\min_{x\in X} F(x,y) = 3x - 6y\\
&\text{subject to } -1x + 3y \leqslant 4\\
&\qquad\qquad -2x + 2y \leqslant 3\\
&\qquad\qquad 0x + 4y \leqslant 5
\end{aligned}
$$

$$
\begin{aligned}
&\min_{y \in Y} f(x,y) = 3x + 3y \\
&\text{subject to } 1x - 1y \leqslant 0 \\
&\qquad 0x - 2y \leqslant -1 \\
&\qquad 2x - 0y \leqslant 1 \\
&\qquad -1x - 1y \leqslant 0 \\
&\qquad -2x - 2y \leqslant -1.
\end{aligned}
$$

Step 6-12. According to the proposed approximation Branch-and-Bound-based algorithm, let us rewrite it as follows:

$$
\begin{aligned}
g_1(x,y) &= 4 - (-1x + 3y) \geqslant 0 \\
g_2(x,y) &= 3 - (-2x + 2y) \geqslant 0 \\
g_3(x,y) &= 5 - (0x + 4y) \geqslant 0 \\
g_4(x,y) &= -(1x - 1y) \geqslant 0 \\
g_5(x,y) &= -1 - (0x - 2y) \geqslant 0 \\
g_6(x,y) &= 1 - (2x - 0y) \geqslant 0 \\
g_7(x,y) &= 1x + 1y \geqslant 0 \\
g_8(x,y) &= -1 - (-2x - 2y) \geqslant 0 \\
g_9(x,y) &= y \geqslant 0
\end{aligned}
$$

and we also have:

$$
\begin{aligned}
&\min_{x \in X} F(x,y) = 3x - 6y \\
&\text{subject to } -1x + 3y \leqslant 4 \\
&\qquad -2x + 2y \leqslant 3 \\
&\qquad 0x + 4y \leqslant 5 \\
&\qquad 1x - 1y \leqslant 0 \\
&\qquad 0x - 2y \leqslant -1 \\
&\qquad 2x - 0y \leqslant 1
\end{aligned}
$$

$$
\begin{aligned}
& -1x - 1y \leqslant 0 \\
& -2x - 2y \leqslant -1 \\
& 3u_1 + 2u_2 + 4u_3 - u_4 - 2u_5 - 0u_6 - u_7 - 2u_8 - u_9 = -3 \\
& \sum_{i=1}^{9} u_i g_i(x, y) = 0 \\
& x \geqslant 0, y \geqslant 0, u_1 \geqslant 0, \ldots, u_9 \geqslant 0.
\end{aligned}
$$

Finally, we get the following linear programming problem with one check condition.

$$
\begin{aligned}
\min_{x \in X} \; & F(x, y) = 3x - 6y \\
\text{subject to } & -1x + 3y \leqslant 4 \\
& -2x + 2y \leqslant 3 \\
& 0x + 4y \leqslant 5 \\
& 1x - 1y \leqslant 0 \\
& 0x - 2y \leqslant -1 \\
& 2x - 0y \leqslant 1 \\
& -1x - 1y \leqslant 0 \\
& -2x - 2y \leqslant -1 \\
& 3u_1 + 2u_2 + 4u_3 - u_4 - 2u_5 - 0u_6 - u_7 - 2u_8 - u_9 = -3 \\
& x \geqslant 0, y \geqslant 0, u_1 \geqslant 0, \ldots, u_9 \geqslant 0.
\end{aligned}
$$

At each time of iteration, the following condition is checked.

$$\sum_{i=1}^{9} u_i g_i(x, y) = 0.$$

More specifically, after initialising the data, the algorithm finds a feasible solution to the Kuhn-Tucker representation with the complementary slackness conditions omitted and proceeds to Step 9. The current point, $x^1 = 0$, $y^1 = 1.25$, $u^1 = (0, 0, 0, 3, 0, 0, 0, 0, 0)$, with $F(x^1, y^1) = -7.5$, is not satisfied complementarities so a branching variable is selected (u_4) and the index sets are updated, giving $S_1^+ = \{4\}$, $S_1^- = \phi$, $S_1^0 = \{1, 2, 3, 5, 6, 7, 8, 9\}$ and $P_1 = \{4\}$.

In the next four iterations, the algorithm branches on u_5, u_7, u_8 and u_9, respectively.

Now, five levels down in the Branch-and-Bound search tree (Figure 5.1), the current sub-problem at Step 7 turns out to be infeasible so the algorithm goes to Step 11 and backtracks. The index sets are $S_5^+ = \{4, 5, 7, 8\}$, $S_5^- = \{9\}$, $S_5^0 = \{1, 2, 3, 6\}$ and $P_5 = \{4, 5, 7, 8, \underline{9}\}$.

So we go to Step 7, and the algorithm turns out to be infeasible. Thus the algorithm goes to Step 11 and backtracks. The index sets are $S_6^+ = \{4, 5, 7\}$, $S_6^- = \{8\}$, $S_6^0 = \{1, 2, 3, 6, 9\}$ and $P_6 = \{4, 5, 7, \underline{8}\}$.

Go to Step 7 again, and the algorithm turns out to be infeasible, so the algorithm goes to Step 11 and backtracks. The index sets are now $S_7^+ = \{4, 5\}$, $S_7^0 = \{1, 2, 3, 6, 8, 9\}$ and $P_7 = \{4, 5, \underline{7}\}$.

Go to Step 7 again, and the algorithm turns out to be infeasible, so the algorithm goes to Step 11 and backtracks. The index sets are $S_8^+ = \{4\}$, $S_8^- = \{5\}$, $S_8^0 = \{1, 2, 3, 6, 7, 8, 9\}$ and $P_8 = \{4, \underline{5}\}$.

Go to Step 7, a feasible solution is found. It passes the test at Step 8 and satisfies the complementary slackness conditions at Step 9. Continuing at Step 8, $\bar{F} = -3$. The algorithm backtracks at Step 11 and updates the sets, $S_9^+ = \phi$, $S_9^- = \{4\}$, $S_9^0 = \{1, 2, 3, 5, 6, 7, 8, 9\}$ and $P_9 = \{\underline{4}\}$. Returning to Step 7, another feasible solution is found, but at Step 8, the value of the leader's objective function is greater than the incumbent upper bound, so it goes to Step 11 and backtracks. However, no live vertices exist. We have found an optimal solution, occurring at the point $(x^*, y^*) = (0, 0.5)$, $(u^*) = (0, 0, 0, 3, 0, 0, 0, 0, 0)$ with $F^* = -3$ and $f^* = 1.5$. The Branch-and-Bound tree is shown in Figure 5.1.

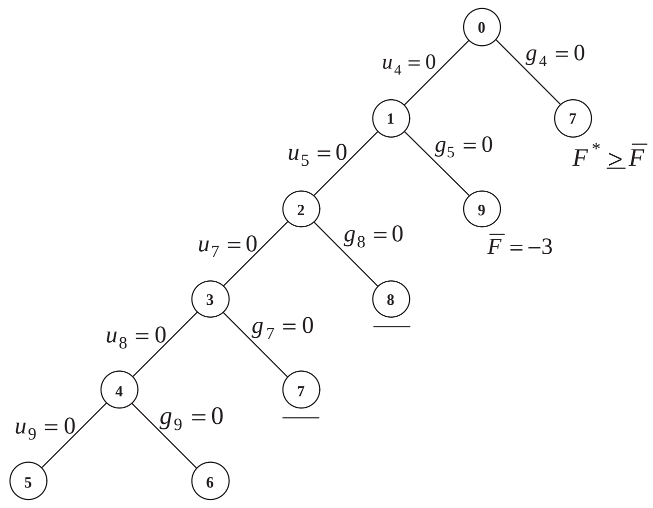

Figure 5.1: A Branch-and-Bound tree

By examining above procedure, we found that the optimal solution occurs at the point $(x^*, y^*) = (0, 0.5)$ with

$$\begin{aligned}
&\min_{x\in X}(F_1(x,y))_1^{L(R)} = 1\\
&\min_{x\in X}(F_1(x,y))_0^{L} = 0.5\\
&\min_{x\in X}(F_1(x,y))_0^{R} = 1.5\\
&\min_{x\in X}(F_2(x,y))_1^{L(R)} = -2\\
&\min_{x\in X}(F_2(x,y))_0^{L} = -2.5\\
&\min_{x\in X}(F_2(x,y))_0^{R} = -1.5\\
&\min_{y\in Y}(f_1(x,y))_1^{L(R)} = -0.5\\
&\min_{y\in Y}(f_1(x,y))_0^{L} = -1\\
&\min_{y\in Y}(f_1(x,y))_0^{R} = 0\\
&\min_{y\in Y}(f_2(x,y))_1^{L(R)} = 1\\
&\min_{y\in Y}(f_2(x,y))_0^{L} = 0.5\\
&\min_{y\in Y}(f_2(x,y))_0^{R} = 1.5
\end{aligned}$$

Step 13. When $l = 2$, we solve the following MOLB decision problem by Step 4

$$\begin{aligned}
&\min_{x\in X}(F_1(x,y))_1^{L(R)} = -1x + 2y\\
&\min_{x\in X}(F_1(x,y))_{\frac{1}{2}}^{L} = -\frac{3}{2}x + \frac{3}{2}y\\
&\min_{x\in X}(F_1(x,y))_0^{L} = -2x + 1y\\
&\min_{x\in X}(F_1(x,y))_{\frac{1}{2}}^{R} = -\frac{\sqrt{2}}{2}x + \frac{5}{2}y\\
&\min_{x\in X}(F_1(x,y))_0^{R} = 0x + 3y\\
&\min_{x\in X}(F_2(x,y))_1^{L(R)} = 2x - 4y\\
&\min_{x\in X}(F_2(x,y))_{\frac{1}{2}}^{L} = \frac{3}{2}x - \frac{9}{2}y\\
&\min_{x\in X}(F_2(x,y))_0^{L} = 1x - 5y
\end{aligned}$$

$$
\begin{aligned}
&\min_{x\in X}(F_2(x,y))^L_{\frac{1}{2}} = \frac{5}{2}x - \frac{7}{2}y\\
&\min_{x\in X}(F_2(x,y))^R_0 = 3x - 3y\\
&\text{subject to } -1x + 3y \leqslant 4\\
&\qquad -\frac{3}{2}x + \frac{5}{2}y \leqslant \frac{7}{2}\\
&\qquad -2x + 2y \leqslant 3\\
&\qquad -\frac{\sqrt{2}}{2}x + \frac{7}{2}y \leqslant \frac{9}{2}\\
&\qquad 0x + 4y \leqslant 5\\
&\qquad \min_{y\in Y}(f_1(x,y))^{L(R)}_1 = 2x - 1y\\
&\qquad \min_{y\in Y}(f_1(x,y))^L_{\frac{1}{2}} = \frac{3}{2}x - \frac{3}{2}y\\
&\qquad \min_{y\in Y}(f_1(x,y))^L_0 = 1x - 2y\\
&\qquad \min_{y\in Y}(f_1(x,y))^R_{\frac{1}{2}} = \frac{5}{2}x - \frac{\sqrt{2}}{2}y\\
&\qquad \min_{y\in Y}(f_1(x,y))^R_0 = 3x - 0y\\
&\qquad \min_{y\in Y}(f_2(x,y))^{L(R)}_1 = -1x + 2y\\
&\qquad \min_{y\in Y}(f_2(x,y))^L_{\frac{1}{2}} = -\frac{3}{2}x + \frac{3}{2}y\\
&\qquad \min_{y\in Y}(f_2(x,y))^L_0 = -2x + 1y\\
&\qquad \min_{y\in Y}(f_2(x,y))^R_{\frac{1}{2}} = -\frac{\sqrt{2}}{2}x + \frac{5}{2}y\\
&\qquad \min_{y\in Y}(f_2(x,y))^R_0 = 0x + 3y\\
&\qquad \text{subject to } 1x - 1y \leqslant 0\\
&\qquad\qquad \frac{\sqrt{2}}{2}x - \frac{3}{2}y \leqslant -\frac{1}{2}\\
&\qquad\qquad 0x - 2y \leqslant -1\\
&\qquad\qquad \frac{3}{2}x - \frac{\sqrt{2}}{2}y \leqslant \frac{\sqrt{2}}{2}\\
&\qquad\qquad 2x - 0y \leqslant 1
\end{aligned}
$$

$$
\begin{aligned}
&-\frac{3}{2}x-\frac{3}{2}y\leqslant -\frac{1}{2}\\
&-1x-1y\leqslant 0\\
&-\frac{\sqrt{2}}{2}x-\frac{\sqrt{2}}{2}y\leqslant \frac{\sqrt{2}}{2}\\
&-2x-2y\leqslant -1.
\end{aligned}
$$

By Step 5 to Step 12, we have

$$
\begin{aligned}
&\min_{x\in X} F(x,y)=\left(3+\frac{5-\sqrt{2}}{2}\right)x-10y\\
&\text{subject to}-1x+3y\leqslant 4\\
&\qquad -\frac{3}{2}x+\frac{5}{2}y\leqslant \frac{7}{2}\\
&\qquad -2x+2y\leqslant 3\\
&\qquad -\frac{\sqrt{2}}{2}x+\frac{7}{2}y\leqslant \frac{9}{2}\\
&\qquad 4y\leqslant 5\\
&\qquad \min_{y\in Y} f(x,y)=\left(\frac{5-\sqrt{2}}{2}+3\right)x+\left(\frac{5-\sqrt{2}}{2}+3\right)y\\
&\qquad \text{subject to } 1x-1y\leqslant 0\\
&\qquad\qquad \frac{\sqrt{2}}{2}x-\frac{3}{2}y\leqslant -\frac{1}{2}\\
&\qquad\qquad 0x-2y\leqslant -1\\
&\qquad\qquad \frac{3}{2}x-\frac{\sqrt{2}}{2}y\leqslant \frac{\sqrt{2}}{2}\\
&\qquad\qquad 2x-0y\leqslant 1\\
&\qquad\qquad -\frac{3}{2}x-\frac{3}{2}y\leqslant -\frac{1}{2}\\
&\qquad\qquad -1x-1y\leqslant 0\\
&\qquad\qquad -\frac{\sqrt{2}}{2}x-\frac{\sqrt{2}}{2}y\leqslant \frac{\sqrt{2}}{2}\\
&\qquad\qquad -2x-2y\leqslant -1
\end{aligned}
$$

The optimal solution occurs at the point $(x^*, y^*) = (0, 0.5)$ with

$$
\begin{aligned}
&\min_{x \in X}(F_1(x,y))_1^{L(R)} = 1\\
&\min_{x \in X}(F_1(x,y))_{\frac{1}{2}}^{L} = 0.75\\
&\min_{x \in X}(F_1(x,y))_0^{L} = 0.5\\
&\min_{x \in X}(F_1(x,y))_{\frac{1}{2}}^{R} = 1.25\\
&\min_{x \in X}(F_1(x,y))_0^{R} = 1.5\\
&\min_{x \in X}(F_2(x,y))_1^{L(R)} = -2\\
&\min_{x \in X}(F_2(x,y))_{\frac{1}{2}}^{L} = -2.25\\
&\min_{x \in X}(F_2(x,y))_0^{L} = -2.5\\
&\min_{x \in X}(F_2(x,y))_{\frac{1}{2}}^{L} = -1.75\\
&\min_{x \in X}(F_2(x,y))_0^{R} = -1.5\\
&\min_{y \in Y}(f_1(x,y))_1^{L(R)} = -0.5\\
&\min_{y \in Y}(f_1(x,y))_{\frac{1}{2}}^{L} = -0.75\\
&\min_{y \in Y}(f_1(x,y))_0^{L} = -1\\
&\min_{y \in Y}(f_1(x,y))_{\frac{1}{2}}^{R} = -\sqrt{2}/4\\
&\min_{y \in Y}(f_1(x,y))_0^{R} = 0\\
&\min_{y \in Y}(f_2(x,y))_1^{L(R)} = 1\\
&\min_{y \in Y}(f_2(x,y))_{\frac{1}{2}}^{L} = 0.75\\
&\min_{y \in Y}(f_2(x,y))_0^{L} = 0.5\\
&\min_{y \in Y}(f_2(x,y))_{\frac{1}{2}}^{R} = 1.25\\
&\min_{y \in Y}(f_2(x,y))_0^{R} = 1.5.
\end{aligned}
$$

Step 14. When $(x, y) = (0, 0.5)$, we have $\|(x, y)_{2^2} - (x, y)_{2^1}\| = 0 < \varepsilon$.

Step 15. The solution of the problem is $(x, y) = (0, 0.5)$ such that

$$\min_{x \in X} F_1(x, y) = 0.5 \times \tilde{2}$$
$$\min_{x \in X} F_2(x, y) = -0.5 \times \tilde{4}$$
$$\min_{y \in Y} f1_1(x, y) = 0.5 \times \tilde{2}$$
$$\min_{y \in Y} f2_1(x, y) = -0.5 \times \tilde{1}.$$

This example shows how the approximation Branch-and-Bound-based algorithm is used to solve an FMMLB decision problem.

5.2.2 An Approximation Kth-Best-based Algorithm

We first present related definitions and theorems.

For the problem defined by Definition 5.1.1, we give the following definition to provide a solution concept for the related MOLB decision problems defined by (5.22).

Definition 5.2.1. (1) Constraint region of the MOLB decision problem (5.22):

$$\begin{aligned} S \triangleq \{&(x, y) : x \in X, y \in Y, \tilde{A}^L_{\lambda_j} x + \sum_{t=1}^{K} \tilde{B}^L_{t\lambda_j} y_t \leqslant \tilde{b}^L_{\lambda_j}, \tilde{A}^R_{\lambda_j} x + \sum_{t=1}^{K} \tilde{B}^R_{t\lambda_j} y_t \leqslant \tilde{b}^R_{\lambda_j}, \\ &\tilde{A}^L_{i\lambda_j} x + \tilde{C}^L_{i\lambda_j} y_i \leqslant \tilde{b}^L_{i\lambda_j}, \tilde{A}^R_{i\lambda_j} x + \tilde{C}^R_{i\lambda_j} y_i \leqslant \tilde{b}^R_{i\lambda_j}, i = 1, \ldots, K, j = 0, 1, \ldots, l.\} \end{aligned}$$

It refers to all possible combination of choices that a leader and followers may make.

(2) Projection of S onto the leader's decision space:

$$\begin{aligned} S(X) \triangleq \{&x \in X : \exists y \in Y, \tilde{A}^L_{\lambda_j} x + \sum_{t=1}^{K} \tilde{B}^L_{t\lambda_j} y_t \leqslant \tilde{b}^L_{\lambda_j}, \tilde{A}^R_{\lambda_j} x + \sum_{t=1}^{K} \tilde{B}^R_{t\lambda_j} y_t \leqslant \tilde{b}^R_{\lambda_j}, \\ &\tilde{A}^L_{i\lambda_j} x + \tilde{C}^L_{i\lambda_j} y_i \leqslant \tilde{b}^L_{i\lambda_j}, \tilde{A}^R_{i\lambda_j} x + \tilde{C}^R_{i\lambda_j} y_i \leqslant \tilde{b}^R_{i\lambda_j}, \\ &i = 1, \ldots, K, j = 0, 1, \ldots, l.\} \end{aligned}$$

(3) Feasible set for the followers $\forall x \in S(X)$:

$$S(x) \triangleq \{y \in Y : \tilde{A}^L_{\lambda_j} x + \sum_{t=1}^{K} \tilde{B}^L_{t\lambda_j} y_t \leqslant \tilde{b}^L_{\lambda_j}, \tilde{A}^R_{\lambda_j} x + \sum_{t=1}^{K} \tilde{B}^R_{t\lambda_j} y_t \leqslant \tilde{b}^R_{\lambda_j},$$
$$\tilde{A}^L_{i\lambda_j} x + \tilde{C}^L_{i\lambda_j} y_i \leqslant \tilde{b}^L_{i\lambda_j}, \tilde{A}^R_{i\lambda_j} x + \tilde{C}^R_{i\lambda_j} y_i \leqslant \tilde{b}^R_{i\lambda_j},$$
$$i = 1, \ldots, K, j = 0, 1, \ldots, l.\}$$

(4) The followers' rational reaction set for $x \in S(x)$:

$$P(x) \triangleq \{y \in Y : y \in argmin[f(x, \hat{y}) : \hat{y} \in S(x)]\}$$

where $argmin[f(x, \hat{y}) : \hat{y} \in S(x)] = \{y \in S(x) : f(x, y) \leqslant f(x, \hat{y}), \hat{y} \in S(x)\}$.

The followers observe the leader's action and reacts by selecting y from his or her feasible set to minimise his or her objective function.

(5) Inducible region:

$$IR \triangleq \{(x, y) : (x, y) \in S, y \in P(x)\}$$

which represents the set over which a leader may optimise his or her objectives.

Theorem 5.2.3. *(Zhang et al. 2006b) The inducible region defined by (5) can be written equivalently as a piecewise linear equality constraint comprised of supporting hyper planes of S.*

Corollary 5.2.1. *(Zhang et al. 2006b) The MOLB problem (5.22) is equivalent to minimising over a feasible region comprised of a piecewise linear equality constraint.*

Corollary 5.2.2. *(Zhang et al. 2006b) A solution for problem (5.22) occurs at a vertex of IR.*

Theorem 5.2.4. *(Zhang & Lu 2007a) The solution $(x^\star, y^\star)$ of problem (5.22) occurs at a vertex of S.*

Corollary 5.2.3. *(Zhang & Lu 2007a) If x is an extreme point of IR, it is an extreme point of S.*

Theorem 5.2.4 and Corollary 5.2.2 have provided theoretical foundation for the fuzzy kth-best approach. It means that by searching extreme points on the constraint region S, we can efficiently find an optimal solution for an FMOLB problem. According to the objective function of the upper level, we order all the extreme points on S in descending order, and select the first extreme point to check if it is on the inducible region IR. If yes, the current extreme point is the optimal solution. Otherwise, continue the process.

More specifically, let $(x_{[1]}, y_{[1]}), \ldots, (x_{[N]}, y_{[N]})$ denote the N ordered extreme points to the linear programming problem:

$$\min \left\{ \sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left({c_{j1}}_{\alpha_i}^{L} x + {d_{j1}}_{\alpha_i}^{L} y \right) + \sum_{i=0}^{n} \left({c_{j1}}_{\alpha_i}^{R} x + {d_{j1}}_{\alpha_i}^{R} y \right) \right) : (x, y) \in S \right\} \tag{5.25}$$

such that

$$\begin{aligned} &\sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left({c_{j1}}_{\alpha_i}^{L} x_{[i]} + {d_{j1}}_{\alpha_i}^{L} y_{[i]} \right) + \sum_{i=0}^{n} \left({c_{j1}}_{\alpha_i}^{R} x_{[i]} + {d_{j1}}_{\alpha_i}^{R} y_{[i]} \right) \right) \leqslant \\ &\quad \sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left({c_{j1}}_{\alpha_i}^{L} x_{[i+1]} + {d_{j1}}_{\alpha_i}^{L} y_{[i+1]} \right) + \sum_{i=0}^{n} \left({c_{j1}}_{\alpha_i}^{R} x_{[i+1]} + {d_{j1}}_{\alpha_i}^{R} y_{[i+1]} \right) \right), \\ &\qquad i = 1, \ldots, N. \end{aligned} \tag{5.26}$$

Let $\bar{y}$ denote the optimal solution to the following problem

$$\min(f(x_{[i]}, y) : y \in S(x_{[i]})) \tag{5.27}$$

We only need to find the smallest i under which $y_[i] = \bar{y}$.

Let us write (5.27) as follows

$$\begin{aligned} &\min f(x, y) \\ &\text{subject to } y \in S(x) \\ &x = x_{[i]} \end{aligned} \tag{5.28}$$

Based on the definitions and theorems presented above and the extended Kth-Best algorithm (Shi *et al.* 2005a), we extend it to fuzzy situations and propose an approximation kth-best-based algorithm for solving FMMLB decision problems in Algorithm 4.

We apply the proposed approximation kth-best-based algorithm to solve a simple

Algorithm 4: An approximation kth-best-based algorithm for FMMLB decision problems

Step 1: Given two sets of weights w_{j1} $(j = 1, \ldots, s)$ and w_{j2} $(j = 1, \ldots, t)$ to the objectives of the leader and the followers respectively, let $\sum_{j=1}^{s} w_{j1} = 1$ and $\sum_{j=1}^{t} w_{j2} = 1$.
Step 2: The problem defined by Definition 5.1.1 is transformed to Problem (5.23).
Step 3: Set $l = 1$, a range of errors $\epsilon > 0$, to solve Problem (5.23) by following steps.
step 3.1: Put $i = 1$. Solve Problem (5.23) with the Simplex method to obtain the optimal solution $(x_{[1]}, y_{[1]})$. Let $W = \{(x_{[1]}, y_{[1]})\}$, $T = \emptyset$. Go to Step 2.
Step 3.2: Solve Problem (5.28) with the bounded simplex method. Let $\bar{y}$ denote the optimal solution to (5.28). If $\bar{y} = y_{[i]}$, stop; $(x_{[i]}, y_{[i]}$ is the global optimum to Problem (5.23). Otherwise, go to Step 3.
Step 3.3: Let $W_{[i]}$ denote the set of adjacent extreme points of $x_{[i]}, y_{[i]}$ such that $(x, y) \in W_{[i]}$ implies $\sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left(c_{j1}{}^{L}_{\alpha_i} x_{[i]} \ + d_{j1}{}^{L}_{\alpha_i} y_{[i]} \right) + \sum_{i=0}^{n} \left(c_{j1}{}^{R}_{\alpha_i} x_{[i]} + d_{j1}{}^{R}_{\alpha_i} y_{[i]} \right) \right) \leqslant \sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left(c_{j1}{}^{L}_{\alpha_i} x_{[i+1]} + d_{j1}{}^{L}_{\alpha_i} y_{[i+1]} \right) + \sum_{i=0}^{n} \left(c_{j1}{}^{R}_{\alpha_i} x_{[i+1]} + d_{j1}{}^{R}_{\alpha_i} y_{[i+1]} \right) \right)$.
Let $T = T \cup \{(x_{[i]}, y_{[i]}\}$ and $W = (W \cup W_{[i]})\, T$. Go to Step 4.
Step 3.4: Set $i = i + 1$ and choose $x_{[i]}, y_{[i]}$ so that $\sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left(c_{j1}{}^{L}_{\alpha_i} x_{[i]} + d_{j1}{}^{L}_{\alpha_i} y_{[i]} \right) + \ \sum_{i=0}^{n} \left(c_{j1}{}^{R}_{\alpha_i} x_{[i]} + d_{j1}{}^{R}_{\alpha_i} y_{[i]} \right) \right) = \min\{\sum_{j=1}^{s} w_{j1} \left(\sum_{i=0}^{n} \left(c_{j1}{}^{L}_{\alpha_i} x + d_{j1}{}^{L}_{\alpha_i} y \right) + \sum_{i=0}^{n} \left(c_{j1}{}^{R}_{\alpha_i} x + d_{j1}{}^{R}_{\alpha_i} y \right) \right) : (x, y) \in W\}$. Go back to Step 2.
Step 4: Decompose interval $[0, 1]$ into 2^{l-1} equal sub-intervals with $(2^{l-1} + 1)$ nodes λ_i $(i = 0, \cdots, 2^{l-1})$ arranged in the order of $0 = \lambda_0 < \cdots < \lambda_{2^{l-1}} = 1$.
Step 5: If $\|(x, y)_{2^{l+1}} - (x, y)_{2^l}\| < \varepsilon$, then the solution (x^*, y^*) of the problem is $(x, y)_{2^{l+1}}$, otherwise, go back to Step 3.1.
Step 6: Show the results, terminates.

FMMLB decision problem to illustrate how the algorithm is used.

In this example, the leader has one objective F. There are two followers, who share decision variables and one objective f but have one individual constraints. We can see that this FMMLB decision problem exactly falls into the category of Model 2.

Example 5.2.2. Consider the following FMMLB decision problem with $x \in R^1$, $y \in R^1$, and $X = \{x \geqslant 0\}$, $Y = \{y \geqslant 0\}$:

$$\min_{x \in X} F(x, y) = \tilde{1}x - \tilde{2}y$$
$$\text{subject to } -\tilde{1}x + \tilde{3}y \leqslant \tilde{4}$$

$$\begin{aligned}
&\min_{y\in Y} f(x,y) = \tilde{1}x + \tilde{1}y \\
&\text{subject to } \tilde{1}x - \tilde{1}y \leqslant \tilde{0} \\
&\qquad\qquad - \tilde{1}x - \tilde{1}y \leqslant \tilde{0}
\end{aligned}$$

where

$$\mu_{\tilde{1}}(t) = \begin{cases} 0, & t<0, \\ t^2, & 0 \leqq t < 1, \\ 2-t, & 1 \leqq t < 2, \\ 0, & 2 \leqq t. \end{cases} \quad \mu_{\tilde{2}}(t) = \begin{cases} 0, & t<1, \\ t-1, & 1 \leqq t < 2, \\ 3-t, & 2 \leqq t < 3, \\ 0, & 3 \leqq t. \end{cases}$$

$$\mu_{\tilde{3}}(t) = \begin{cases} 0, & t<2, \\ t-2, & 2 \leqq t < 3, \\ 4-t, & 3 \leqq t < 4, \\ 0, & 4 \leqq t. \end{cases} \quad \mu_{\tilde{4}}(t) = \begin{cases} 0, & t<3, \\ t-3, & 3 \leqq t < 4, \\ 5-t, & 4 \leqq t < 5, \\ 0, & 5 \leqq t. \end{cases}$$

$$\mu_{\tilde{0}}(t) = \begin{cases} 0, & t<-1, \\ t+1, & -1 \leqq t < 0, \\ 1-t^2, & 0 \leqq t < 1, \\ 0, & 1 \leqq t. \end{cases}$$

We now solve this problem by using the proposed approximation kth-best-based algorithm.

Step 1. The original problem is transferred to the following problem by using Theorem 5.2.1.

$$\begin{aligned}
&\min_{x\in X}(F(x,y))_c = 1x - 2y \\
&\min_{x\in X}(F(x,y))_0^L = 0x - 3y \\
&\min_{x\in X}(F(x,y))_0^R = 2x - 1y \\
&\text{subject to } -1x + 3y \leqslant 4 \\
&\qquad\qquad -2x + 2y \leqslant 3 \\
&\qquad\qquad 0x + 4y \leqslant 5 \\
&\qquad\qquad \min_{y\in Y}(f(x,y))_c = 1x + 1y
\end{aligned}$$

$$
\begin{aligned}
&\min_{y\in Y}(f(x,y))_0^L = 0x + 0y\\
&\min_{y\in Y}(f(x,y))_0^R = 2x + 2y\\
&\text{subject to } 1x - 1y \leqslant 0\\
&\qquad\qquad 0x - 2y \leqslant -1\\
&\qquad\qquad 2x - 0y \leqslant 1\\
&\qquad\qquad -1x - 1y \leqslant 0\\
&\qquad\qquad 0x - 0y \leqslant 0\\
&\qquad\qquad -2x - 2y \leqslant -1
\end{aligned}
$$

Step 2. This problem is then transferred to the following linear bi-level problem by using a weighting method.

$$
\begin{aligned}
&\min_{x\in X} F(x,y) = 3x - 6y\\
&\text{subject to } -1x + 3y \leqslant 4\\
&\qquad -2x + 2y \leqslant 3\\
&\qquad 0x + 4y \leqslant 5\\
&\qquad \min_{y\in Y} f(x,y) = 3x + 3y\\
&\qquad \text{subject to } 1x - 1y \leqslant 0\\
&\qquad\qquad 0x - 2y \leqslant -1\\
&\qquad\qquad 2x - 0y \leqslant 1\\
&\qquad\qquad -1x - 1y \leqslant 0\\
&\qquad\qquad 0x - 0y \leqslant 0\\
&\qquad\qquad -2x - 2y \leqslant -1
\end{aligned}
$$

Step 3. According to the extended Kth-Best approach proposed in (Shi *et al.* 2005a), we obtain the following problem:

$$
\begin{aligned}
&\min_{x\in X} F(x,y) = 3x - 6y\\
&\text{subject to } -1x + 3y \leqslant 4\\
&\qquad -2x + 2y \leqslant 3
\end{aligned}
$$

$$
\begin{aligned}
&0x + 4y \leqslant 5 \\
&1x - 1y \leqslant 0 \\
&0x - 2y \leqslant -1 \\
&2x - 0y \leqslant 1 \\
&-1x - 1y \leqslant 0 \\
&0x - 0y \leqslant 0 \\
&-2x - 2y \leqslant -1
\end{aligned}
$$

Let $i = 1$, and solve the above problem with the Simplex method to obtain the optimal solution $(x_{[1]}, y_{[1]}) = (0, 1.25)$. Let $W = \{0, 1.25\}$ and $T = \emptyset$. Go to Step 2.

Loop 1:

By (5.28), we have:

$$
\begin{aligned}
&\min_{x \in X} f(x, y) = 3x + 3y \\
&\text{subject to } -1x + 3y \leqslant 4 \\
&\qquad -2x + 2y \leqslant 3 \\
&\qquad 0x + 4y \leqslant 5 \\
&\qquad 1x - 1y \leqslant 0 \\
&\qquad 0x - 2y \leqslant -1 \\
&\qquad 2x - 0y \leqslant 1 \\
&\qquad -1x - 1y \leqslant 0 \\
&\qquad 0x - 0y \leqslant 0 \\
&\qquad -2x - 2y \leqslant -1 \\
&\qquad x = 0 \\
&\qquad y \leqslant 0
\end{aligned}
$$

Using the bounded Simplex method, we have $\bar{y} = 0.5$. Because $\bar{y} \neq y_{[i]}$, we go to Step 3. We have $W_{[i]} = \{(0.5, 1.25), (0, 0.5), (0, 1.25)\}$, $T = \{(0, 1.25)\}$ and $W = \{(0, 0.5), (0.5, 1.25)\}$, then go to Step 4. Update $i = 2$, and choose $(x_{[i]}, y_{[i]}) = (0.5, 1.25)$, then go to Step 2.

Loop 2:

By (5.28), we have:

$$\begin{aligned}
&\min_{x\in X} f(x,y) = 3x + 3y \\
&\text{subject to } -1x + 3y \leqslant 4 \\
&\quad -2x + 2y \leqslant 3 \\
&\quad 0x + 4y \leqslant 5 \\
&\quad 1x - 1y \leqslant 0 \\
&\quad 0x - 2y \leqslant -1 \\
&\quad 2x - 0y \leqslant 1 \\
&\quad -1x - 1y \leqslant 0 \\
&\quad 0x - 0y \leqslant 0 \\
&\quad -2x - 2y \leqslant -1 \\
&\quad x = 0.5 \\
&\quad y \leqslant 0
\end{aligned}$$

Using the bounded Simplex method, we have $\bar{y} = 0.5$. Because $\bar{y} \neq y_{[i]}$, we go to Step 3. We have $W_{[i]} = \{(0.5, 1.25), (0, 0.5), (0, 1.25)\}$, $T = \{(0, 1.25), (0.5, 1.25)\}$ and $W = \{(0, 0.5), (0.5, 0.5)\}$, then go to Step 4. Update $i = 3$, and choose $(x_{[i]}, y_{[i]}) = (0, 0.5)$, then go to Step 2.

Loop 3:

By (5.28), we have:

$$\begin{aligned}
&\min_{x\in X} f(x,y) = 3x + 3y \\
&\text{subject to } -1x + 3y \leqslant 4 \\
&\quad -2x + 2y \leqslant 3 \\
&\quad 0x + 4y \leqslant 5 \\
&\quad 1x - 1y \leqslant 0 \\
&\quad 0x - 2y \leqslant -1 \\
&\quad 2x - 0y \leqslant 1 \\
&\quad -1x - 1y \leqslant 0
\end{aligned}$$

$$
\begin{aligned}
&0x - 0y \leqslant 0 \\
&-2x - 2y \leqslant -1 \\
&x = 0 \\
&y \leqslant 0
\end{aligned}
$$

Using the bounded Simplex method, we have $\bar{y} = 0.5$. Because $\bar{y} = y_{[i]}$, we go to Step 3. We stop here. $(x_{[i]}, y_{[i]}) = (0, 0.5)$ is the global solution to this example, By examining above procedure, we found that the optimal solution occurs at the point $(x^*, y^*) = (0, 0.5)$.

Step 4. The result is

$$
\begin{aligned}
&\min_{x \in X}(F(x,y))_c = -1 \\
&\min_{x \in X}(F(x,y))_0^L = -1.5 \\
&\min_{x \in X}(F(x,y))_0^R = -0.5 \\
&\min_{y \in Y}(f(x,y))_c = 0.5 \\
&\min_{y \in Y}(f(x,y))_0^L = 0 \\
&\min_{y \in Y}(f(x,y))_0^R = 1
\end{aligned}
$$

Consequently, under this solution, we have the objective functions for both the leader and the follower as follows:

$$
\begin{aligned}
&\min_{x \in X} F(x,y) = \tilde{1}x - \tilde{2}y = \tilde{c} \\
&\min_{y \in Y} f(x,y) = \tilde{1}x + \tilde{1}y = \tilde{d} \\
&x = 0, y = 0.5
\end{aligned}
$$

where

$$
\mu_{\tilde{c}}(t) = \begin{cases} 0, & t < -1.5, \\ \frac{t+1.5}{0.5}, & -1.5 \leqq t < -1, \\ \frac{-0.5-t}{0.5}, & -1 \leqq t < -0.5, \\ 0, & -0.5 \leqq t. \end{cases} \qquad \mu_{\tilde{d}}(t) = \begin{cases} 0, & t < 0, \\ \frac{t}{0.5}, & 0 \leqq t < 0.5, \\ \frac{1-t}{0.5}, & 0.5 \leqq t < 1, \\ 0, & 1 \leqq t. \end{cases}
$$

5.2.3 A PSO-based Algorithm

In this section, we apply the PSO technique on FMMLB decision problems and develop an algorithm accordingly. The detailed algorithm is specified in Algorithm 5. The notations used in subsequent paragraphs are explained in Table 5.1.

Table 5.1: The explanation of some notations for Algorithm 5

N	the number of candidate solutions (particles) by the leader within its swarm;
M	the number of candidate solutions (particles) by the followers within its swarm;
m	$=\sum_{i=1}^{K} m_i$, the total number of decision variables from followers;
x_i	$= (x_{i1}, x_{i2}, \dots, x_{in})^T$, $i = 1, \dots, N$, the i^{th} candidate solution for the leader;
v_i	$= (v_{i1}, v_{i2}, \dots, v_{in})^T$, $i = 1, \dots, N$, the velocity of x_i;
y_i	$=(y_{i1}, y_{i2}, \dots, y_{im})^T$, the followers' choice for each x_i from the leader;
y_{ij}	$= (y_{ij1}, y_{ij2}, \dots, y_{ijm})^T$, $j = 1, \dots, M$, the j^{th} candidate solution by the followers for the choice x_i from the leader;
v_{ij}	$= (v_{ij1}, ..., v_{ijm})^T$, $j = 1, \dots, M$, the velocity of y_{ij};
p_i	$= (p_{i1}, p_{i2}, \dots, p_{in})^T$, the best previously visited position of x_i;
p_{ij}	$= (p_{ij1}, p_{ij2}, \dots, p_{ijm})^T$, the best previously visited position of y_{ij};
y_{pi}	$= (y_{pi1}, y_{pi2}, \dots, y_{pim})^T$, the response from the followers for the choice p_i from the leader;
CS	$= (CS_1, CS_2, \dots, CS_n)$, the recording vector to record if x_i is within constraint area;
g	the index of the best particle for the leader in the swarm;
k_l	current iteration number for the upper-level problem;
k_f	current iteration number for the lower-level problem;
$MaxK_l$	the predefined max iteration number for k_l;
$MaxK_f$	the predefined max iteration number for k_f.

Figure 5.2 shows the outline of this PSO-based algorithm. It first samples the leader-controlled variables to get some candidate choices for a leader. Then, we use PSO method together with the stretching technology (Parsopoulos & Vrahatis 2002) to get followers' response for every leader's choice. Thus a pool of candidate solutions for both the leader and the followers is formed. By pushing every solution pair moving towards current best ones, the whole solution pool is updated. Once a solution is reached for the leader, we use the stretching technology (Parsopoulos & Vrahatis 2002) to escape the local optimisation. We repeat this procedure by a pre-defined count and

reach a final solution.

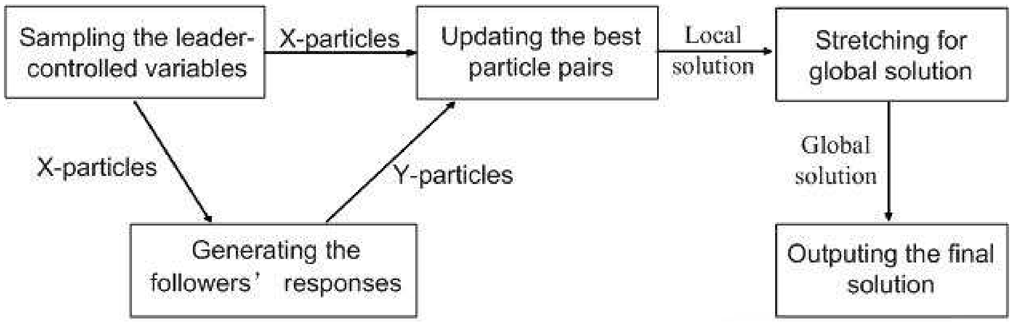

Figure 5.2: The outline of Algorithm 5

First we initiate a swarm comprised by the leader-controlled variables (X_particles). For each particle (x_i) in the swarm, we fix x_i and pass it to the followers as a constant. Then the optimal response from the followers can be generated by solving the following single level optimisation problem:

$$\begin{array}{l} \min\limits_{y_i \in Y_i} f(x, y_1, y_2, \ldots, y_K) = \tilde{a}x + \sum_{i=1}^{k} \tilde{e}_i y_i \\ \text{subject to } \tilde{D}_i x + \tilde{C}_i y_i \preceq_\alpha \tilde{d}_i \\ i = 1, 2, \ldots, K \end{array} \tag{5.29}$$

To solve Problem (5.29), we also need to generate a population (Y_particles), each of which has a velocity. Both the Y_particles and the corresponding velocities are random number distributed among a pre-defined range. The followers thus have many candidate solutions of $(x_i, y_{ij}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, M)$. From every particle pair (x_i, y_{ij}), a bunch of the followers' objective values can be generated, which are inevitably fuzzy numbers. These fuzzy objective values will be evaluated by comparing any two of them using Definition 2.6.2. By this ranking method, we can select the previously visited best positions for all y_particles and the best one among y_particles. Then the stretching technology will be used to erase local solutions. Having current best positions, we adjust the velocities which are redirected towards these best positions. Then every y_particle will be moved by its corresponding velocity. Specifically, we use the following equations to update the position (y_{ij}) and velocity (v_{ij}) for each in Y_particles:

$$\begin{array}{l} v_{ij}^{k_f+1} = wv_{ij}^{k_f} + cr_1^{k_f}(p_{ij} - y_{ij}^{k_f}) + cr_2^{k_f}(y_i^{k_f} - y_{ij}^{k_f}) \\ y_{ij}^{k_f+1} = y_{ij}^{k_f} + v_{ij}^{k_f+1} \end{array} \tag{5.30}$$

Here, k_f is to record current loop. Once k_f is larger than some predefined value, y_i

Algorithm 5: A PSO-based algorithm for FMMLB decision problems

Input: Parameters of the problem defined by Equation (5.17) and Equation (5.18)
Output: (x_g, y_g)
Sampling:
$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T$;
$v_i = (v_{i1}, v_{i2}, ...v_{in})^T$, $i = 1, \ldots, N$;
Generating the responses from the follower:
foreach x_i **do**
  $k_f = 1$;
  $p_{ij} = (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T = (0, 0, \ldots, 0)^T$;
  Sampling:
  $y_{ij} = (y_{ij1}, y_{ij2}, \ldots, y_{ijm})^T$;
  $v_{ij} = (v_{ij1}, v_{ij2}, ...v_{ijm})^T$, $j = 1, \ldots, M$;
1 CS_i = false;
  if $\tilde{A}_1 x + \tilde{B}_1 y \preceq \tilde{b}_1$ ***and*** $\tilde{A}_2 x_i + \tilde{B}_2 y \preceq \tilde{b}_2$ **then**
    CS_i = true;
  end
  if *(*$p_{ij} = (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T = (0, 0, \ldots, 0)^T$*)* ***or*** *(*$f(x_i, y_{ij}) \preceq f(x_i, p_{ij})$*)* **then**
    $p_{ij} = (p_{ij1}, p_{ij2}, \ldots, p_{ijm})^T = (y_{ij1}, y_{ij2}, \ldots, y_{ijm})^T$;
  end
  Stretching for global solution for follower by (2.8);
  Searching the best response y_i from p_{ij}, $j = 1, 2, \ldots, M$;
  Updating velocities and positions using Equation (5.30);
  $k_f = k_f + 1$;
  if $k_f \geqslant$ *MaxK* **then**
    Goto 2;
  else
    Goto 1;
  end
end
2 **if** CS_i = *true* **then**
  if *(*$p_i = (p_{i1}, \ldots, p_{in})^T = (0, \ldots, 0)^T$*)* ***or*** *(*$F(x_i, y_i) \leqslant F(p_i, y_{pi})$*)* **then**
    $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})^T = (x_{i1}, x_{i2}, \ldots, x_{in})^T$;
    $y_{pi} = (y_{pi1}, y_{pi2}, \ldots, y_{pim})^T = (y_{i1}, y_{i2}, \ldots, y_{im})^T$;
  end
end
Stretching for global solution for leader by (2.8);
Searching (x_g, y_g) from p_i, and y_{pi}, $i = 1, \ldots, N$;
Updating x_i and v_i using Equation (5.31);
$k_l = k_l + 1$;
if $k_l \geqslant$ *MaxK* **then**
  stop
else
  Goto 2;
end

will be sent to the leader as the followers' response for x_i.

Having obtained the responses of $y_i, i = 1, 2, \ldots, K$, from the followers, the leader's objective values for each x_i can be calculated. We then use the fuzzy ranking method defined by Definition 2.6.2 again to compare these objective values and select the best position for each x_i and the best one among them, which can make the most optimal objective values. After using the stretching technology on current found best ones, the PSO technique is applied for every particle pair (x_i, y_i) to update the position (x_i) and velocity (v_i) of every leader's particle:

$$\begin{aligned} v_i^{k_l+1} &= wv_i^{k_l} + cr_1^{k_l}(p_i - x_i^{k_l}) + cr_2^{k_l}(x_g^{k_l} - x_i^{k_l}) \\ x_i^{k_l+1} &= x_i^{k_l} + v_i^{k_l+1} \end{aligned} \tag{5.31}$$

Once the iteration times k_l is large enough, current best particle pair (x_g, y_g) will be outputted as the final solution.

5.3 Experiments

In this section, we employ one numerical example to test the performance of the three FMMLB algorithms developed in this chapter. Based on the experiments, we then discuss the choice for these algorithms.

Suppose the example is as:

$$\begin{aligned} \text{Leader :} \quad & \max_{x \in X} F(x, y_1, y_2) = -\tilde{3}x + \tilde{3}y_1 + \tilde{21}y_2 \\ & \text{subject to } -\tilde{3}x + \tilde{6}y_1 + \tilde{4}y_2 \preceq_\alpha \tilde{3} \\ \text{Follower 1 :} \quad & \min_{y_1 \in Y_1} f(x, y_2) = \tilde{6}x + \tilde{4}y_1 \\ \text{Follower 2 :} \quad & \min_{y_2 \in Y_2} f(x, y_2) = \tilde{3}x + \tilde{1}y_2 \\ & \text{subject to } \tilde{3}x + \tilde{3}y_1 \preceq_\alpha \tilde{21} \\ & \qquad -\tilde{1}x + \tilde{1}y_2 \preceq_\alpha -\tilde{1} \end{aligned}$$

where $X = \{x \geqslant 0\}$, $Y_1 = \{y_1 \geqslant 0\}$, and $Y_2 = \{y_2 \geqslant 0\}$.

The membership functions for this FMMLB problem are as follows:

$$\mu_{\tilde{-3}}(x)=\begin{cases}0 & x<-4\\ (16-x^2)/7 & -4\leqq x<-3\\ 1 & x=-3\\ (x^2-1)/8 & -3<x\leqslant -1\\ 0 & x>-1\end{cases},\ \mu_{\tilde{3}}(x)=\begin{cases}0 & x<2\\ (x^2-4)/5 & 2\leqq x<3\\ 1 & x=3\\ (25-x^2)/16 & 3<x\leqslant 5\\ 0 & x>5\end{cases}.$$

$$\mu_{\tilde{21}}(x)=\begin{cases}0 & x<19\\ (x^2-361)/80 & 19\leqq x<21\\ 1 & x=21\\ (625-x^2)/184 & 21<x\leqslant 25\\ 0 & x>25\end{cases},\ \mu_{\tilde{6}}(x)=\begin{cases}0 & x<5\\ (x^2-25)/11 & 5\leqq x<6\\ 1 & x=6\\ (64-x^2)/28 & 6<x\leqslant 8\\ 0 & x>8\end{cases}.$$

$$\mu_{\tilde{4}}(x)=\begin{cases}0 & x<3\\ (x^2-9)/7 & 3\leqq x<4\\ 1 & x=4\\ (36-x^2)/20 & 4<x\leqslant 6\\ 0 & x>6\end{cases},\ \mu_{\tilde{1}}(x)=\begin{cases}0 & x<0.5\\ (x^2-0.25)/0.75 & 0.5\leqq x<1\\ 1 & x=1\\ (4-x^2)/3 & 1<x\leqslant 2\\ 0 & x>2\end{cases}.$$

$$\mu_{\tilde{-1}}(x)=\begin{cases}0 & x<-2\\ (4-x^2)/3 & -2\leqq x<-1\\ 1 & x=-1\\ (x^2-0.25)/0.75 & -1<x\leqslant -0.5\\ 0 & x>-0.5\end{cases}.$$

This example was run by the approximation Branch-and-Bound-based algorithm, the approximation Kth-Best-based algorithm, and the PSO-based algorithm proposed in this chapter, which were implemented by Visual Basic 6.0, and tested on a desktop computer with CPU Pentium 4 2.8GHz, RAM 1G, Windows XP.

In the experiments, we compare the solutions obtained from the approximation Branch-and-Bound-based algorithm, the approximation Kth-Best-based algorithm, and the PSO-based algorithm. For the PSO-based algorithm, the inertia weight w is initially set as 1.2, and is gradually declined towards 0, and the population size is set as 20. Now we adjust the parameters of c_1 and c_2 from 0.5 to 2 respectively. Under every pair of specific c_1 and c_2, this example was run in the PSO-based algorithm by five times, and different solutions have been obtained. To evaluate the performance, Table 5.2 lists the experiment result, where $(\triangle x^*, \triangle y_1^*, \triangle y_2^*)$ represents the average solution difference between the PSO-based algorithm and the classical algorithms, i.e. the approximation Branch-and-Bound-based algorithm and the approximation Kth-Best-based algorithm, for every decision vector. The column of "$\triangle$" sums up the average difference by every decision vector for every c_1, c_2 pair. In the column of "Time", the average running time is listed which is calculated by seconds.

Table 5.2: Summary of the running solutions

(c_1, c_2)	$(\triangle x^*, \triangle y_1^*, \triangle y_2^*)$	$\triangle$	Time
(0.5, 0.5)	(0.68834, 0.04396, 0.17334)	0.90564	46.2
(0.5, 1)	(0.68276, 0.0276, 0.17334)	0.8837	42.4
(0.5, 1.5)	(0.68526, 0.10666, 0.17334)	0.96526	41
(0.5, 2)	(0.67506, 0.04536, 0.17334)	0.89374	40.6
(1, 0.5)	(0.69212, 0.01128, 0.17334)	0.87674	40.2
(1, 1)	(0.69074, 0.04364, 0.17334)	0.97074	38.4
(1, 1.5)	(0.68626, 0.04422, 0.17334)	0.90382	39
(1, 2)	(0.69294 , 0.10666 , 0.17334)	0.97294	38.6
(1.5, 0.5)	(0.67422 , 0.04572 , 0.17334)	0.89328	39
(1.5, 1)	(0.69262 , 0.10666 , 0.17334)	0.97262	39.8
(1.5, 1.5)	(0.66826 , 0.04642 , 0.17334)	0.88802	39
(1.5, 2)	(0.6918 , 0.04352 , 0.17334)	0.90866	39.6
(2, 0.5)	(0.6926 , 0.0434 , 0.17334)	0.90934	39
(2, 1)	(0.6869 , 0.04412 , 0.17334)	0.90436	38.4
(2, 1.5)	(0.68622 , 0.0861 , 0.17334)	0.94566	38.6
(2, 2)	(0.68336 , 0.04458 , 0.17334)	0.90128	39.6

It can be seen from Table 5.2 that the approximation Branch-and-Bound-based algorithm and the approximation Kth-Best-based algorithm can reach exactly the same result, which can demonstrate that in a certain degree these two algorithms are effective to solve FMMLB decision problems.

Meanwhile, we can see the solutions obtained from the PSO-based algorithm are quite close from those from the approximation Branch-and-Bound-based algorithm and the approximation Kth-Best-based algorithm, and do not fluctuate much with the parameters' change. These illustrate that the performance of the PSO-based algorithm is quite stable.

From the fact that the approximation Branch-and-Bound-based algorithm and the approximation Kth-Best-based algorithm can reach exactly the same result which is very close to the result from the PSO-based algorithm, we can come to the conclusion that these three algorithms are valid to solve FMMLB decision problems.

What we can not ignore is that the computation time of the PSO-based algorithm is still much longer than the classical algorithms. This inefficiency comes from the nature of heuristic strategy which simulates the optimisation process while the classical methods use the mathematical properties to directly reach the solution. However, using the mathematical properties sometimes can not reach a solution when these properties can not be satisfied, while heuristic methods are capable of overpassing these complex property verification to generate a reasonable solution at all. In many situations, this reasonable solution is very helpful for a decision maker when making a plan.

5.4 Summary

This chapter addresses FMMLB decision problems. First we propose a model framework to define FMMLB problems by different cooperation in objectives, constraints, and decision variables among followers. Then using cutset strategy and PSO, three algorithms, i.e. an approximation Branch-and-Bound-based algorithm, an approximation Kth-Best-based algorithm, and a PSO-based algorithm are developed to solve FMMLB decision problems. Experiments are carried and comparisons are made among these algorithms. Based on the experiments, we discuss the choice among these algorithms.

6 Cutset Strategy for Fuzzy Linear Goal Bi-level Decision Making

Goal programming requests a decision maker to set a goal for the objective that he/she wishes to attain. A preferred solution is then defined to minimise the deviation from the goal. Therefore goal programming seems to yield a satisfactory solution rather than an optimal one. In fuzzy linear bi-level decision problems, when both a leader and a follower set goals for their objectives respectively, the problem becomes a fuzzy linear bi-level goal (FLBG) decision problem, which is addressed in this Chapter. After presenting a FLBG model, a $\lambda-$cut-based algorithm is developed. A numerical example is then employed to demonstrate the model and the proposed algorithm.

6.1 A Model

Definition 6.1.1. A $\lambda-$cut-based FLBG model is defined as:

$$\begin{aligned}
&\min_{x\in X} |c^L_{1\lambda_j}x + d^L_{1\lambda_j}y - g^L_{L\lambda_j}|,\\
&\min_{x\in X} |c^R_{1\lambda_j}x + d^R_{1\lambda_j}y - g^R_{L\lambda_j}|,
\end{aligned} \tag{6.1a}$$

$$\begin{aligned}
\text{subject to } &A_1{}^L_{\lambda_j}x + B_1{}^L_{\lambda_j}y \leqq b_1{}^L_{\lambda_j},\\
&A_1{}^R_{\lambda_j}x + B_1{}^R_{\lambda_j}y \leqq b_1{}^R_{\lambda_j},
\end{aligned} \tag{6.1b}$$

$$\begin{aligned}
&\min_{y\in Y} |c^L_{2\lambda_j}x + d^L_{2\lambda_j}y - g^L_{F\lambda_j}|,\\
&\min_{y\in Y} |c^R_{2\lambda_j}x + d^R_{2\lambda_j}y - g^R_{F\lambda_j}|,
\end{aligned} \tag{6.1c}$$

$$\begin{aligned}
\text{subject to } &A_2{}^L_{\lambda_j}x + B_2{}^L_{\lambda_j}y \leqq b_2{}^L_{\lambda_j},\\
&A_2{}^R_{\lambda_j}x + B_2{}^R_{\lambda_j}y \leqq b_2{}^R_{\lambda_j}\\
&j = 0, 1, 2, ..., l,
\end{aligned} \tag{6.1d}$$

where $\tilde{c}_1$, $\tilde{c}_2 \in F^n(R)$, $\tilde{d}_1$, $\tilde{d}_2 \in F^m(R)$, $\tilde{b}_1 \in F^p(R)$, $\tilde{b}_2 \in F^*(R^q)$, $\tilde{A}_1 = (\tilde{a}_{ij})_{p\times n}$, $\tilde{B}_1 = \left(\tilde{b}_{ij}\right)_{p\times m}$, $\tilde{A}_2 = (\tilde{e}_{ij})_{q\times n}$, $\tilde{B}_2 = (\tilde{s}_{ij})_{q\times m}$, $\tilde{a}_i$, $\tilde{b}_i$,$\tilde{d}_i$, $\tilde{a}_{ij}$, $\tilde{b}_{ij}$, $\tilde{e}_{ij}$, $\tilde{s}_{ij} \in F(R)$.

This model uses $\lambda-$cut to defuzzify and describe a FLBG decision problem. Based on this model, a $\lambda-$cut-based algorithm will be presented in next section.

6.2 A $\lambda-$cut-based Algorithm

For a clear understanding of the idea adopted, we define:

$$\begin{aligned}
v_{1\lambda_j}^{L-} &= \tfrac{1}{2}[|c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y - g_{L\lambda_j}^{L}| - (c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y - g_{L\lambda_j}^{L})] \\
v_{1\lambda_j}^{L+} &= \tfrac{1}{2}[|c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y - g_{L\lambda_j}^{L}| + (c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y - g_{L\lambda_j}^{L})] \\
v_{1\lambda_j}^{R-} &= \tfrac{1}{2}[|c_{1\lambda_j}^{R}x + d_{1\lambda_j}^{R}y - g_{L\lambda_j}^{R}| - (c_{1\lambda_j}^{R}x + d_{1\lambda_j}^{R}y - g_{L\lambda_j}^{R})] \\
v_{1\lambda_j}^{R+} &= \tfrac{1}{2}[|c_{1\lambda_j}^{R}x + d_{1\lambda_j}^{R}y - g_{L\lambda_j}^{R}| + (c_{1\lambda_j}^{R}x + d_{1\lambda_j}^{R}y - g_{L\lambda_j}^{R})] \\
v_{2\lambda_j}^{L-} &= \tfrac{1}{2}[|c_{2\lambda_j}^{L}x + d_{2\lambda_j}^{L}y - g_{F\lambda_j}^{L}| - (c_{2\lambda_j}^{L}x + d_{2\lambda_j}^{L}y - g_{F\lambda_j}^{L})] \\
v_{2\lambda_j}^{L+} &= \tfrac{1}{2}[|c_{2\lambda_j}^{L}x + d_{2\lambda_j}^{L}y - g_{F\lambda_j}^{L}| + (c_{2\lambda_j}^{L}x + d_{2\lambda_j}^{L}y - g_{F\lambda_j}^{L})] \\
v_{2\lambda_j}^{R-} &= \tfrac{1}{2}[|c_{2\lambda_j}^{R}x + d_{2\lambda_j}^{R}y - g_{F\lambda_j}^{R}| - (c_{2\lambda_j}^{R}x + d_{2\lambda_j}^{R}y - g_{F\lambda_j}^{R})] \\
v_{2\lambda_j}^{R+} &= \tfrac{1}{2}[|c_{2\lambda_j}^{R}x + d_{2\lambda_j}^{R}y - g_{F\lambda_j}^{R}| + (c_{2\lambda_j}^{R}x + d_{2\lambda_j}^{R}y - g_{F\lambda_j}^{R})]
\end{aligned} \tag{6.2}$$

Associated with FLBG problems defined by (6.1), we now consider the following bi-level decision problem:

For $(v_{1\lambda_j}^{L-}, v_{1\lambda_j}^{L+}, v_{1\lambda_j}^{R-}, v_{1\lambda_j}^{R+}) \in R^4$, $X' \subseteq X \times R^4$, $(v_{2\lambda_j}^{L-}, v_{2\lambda_j}^{L+}, v_{2\lambda_j}^{R-}, v_{2\lambda_j}^{R+}) \in R^4$, $Y' \subseteq Y \times R^4$, let $x = (x_1, \cdots, x_n) \in X$, $x' = (x_1, \cdots, x_n, v_{1\lambda_j}^{L-}, v_{1\lambda_j}^{L+}, v_{1\lambda_j}^{R-}, v_{1\lambda_j}^{R+}) \in X'$, $y = (y_1, \cdots, y_m) \in Y$, $y' = (y_1, \cdots, y_m, v_{2\lambda_j}^{L-}, v_{2\lambda_j}^{L+}, v_{2\lambda_j}^{R-}, v_{2\lambda_j}^{R+}) \in Y'$, and $v_{1\lambda_j}^{L}, v_{1\lambda_j}^{R}, v_{2\lambda_j}^{L}, v_{2\lambda_j}^{R} : X' \times Y' \to F(R)$.

$$\begin{aligned}
&\min_{x' \in X'} v_{1\lambda_j}^{L} = v_{1\lambda_j}^{L-} + v_{1\lambda_j}^{L+} \\
&\min_{x' \in X'} v_{1\lambda_j}^{R} = v_{1\lambda_j}^{R-} + v_{1\lambda_j}^{R+}
\end{aligned} \tag{6.3a}$$

$$\begin{aligned}
\text{subject to } & c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y + v_{1\lambda_j}^{L-} - v_{1\lambda_j}^{L+} = g_{L\lambda_j}^{L}, \\
& c_{1\lambda_j}^{R}x + d_{1\lambda_j}^{R}y + v_{1\lambda_j}^{R-} - v_{1\lambda_j}^{R+} = g_{L\lambda_j}^{R}, \\
& v_{1\lambda_j}^{L-}, v_{1\lambda_j}^{L+}, v_{1\lambda_j}^{R-}, v_{1\lambda_j}^{R+} \geqslant 0, \\
& v_{1\lambda_j}^{L-} \cdot v_{1\lambda_j}^{L+} = 0, \\
& v_{1\lambda_j}^{R-} \cdot v_{1\lambda_j}^{R+} = 0, \\
& {A_1}_{\lambda_j}^{L}x + {B_1}_{\lambda_j}^{L}y \leqq {b_1}_{\lambda_j}^{L}, \\
& {A_1}_{\lambda_j}^{R}x + {B_1}_{\lambda_j}^{R}y \leqq {b_1}_{\lambda_j}^{R},
\end{aligned} \tag{6.3b}$$

$$\begin{aligned}
&\min_{y' \in Y'} v^L_{2\lambda_j} = v^{L-}_{2\lambda_j} + v^{L+}_{2\lambda_j} \\
&\min_{y' \in Y'} v^R_{2\lambda_j} = v^{R-}_{2\lambda_j} + v^{R+}_{2\lambda_j}
\end{aligned} \tag{6.3c}$$

$$\begin{aligned}
\text{subject to } & c^L_{2\lambda_j} x + d^L_{2\lambda_j} y + v^{L-}_{2\lambda_j} - v^{L+}_{2\lambda_j} = g^L_{F\lambda_j}, \\
& c^R_{2\lambda_j} x + d^R_{2\lambda_j} y + v^{R-}_{2\lambda_j} - v^{R+}_{2\lambda_j} = g^R_{F\lambda_j}, \\
& v^{L-}_{2\lambda_j}, v^{L+}_{2\lambda_j}, v^{R-}_{2\lambda_j}, v^{R+}_{2\lambda_j} \geqslant 0, \\
& v^{L-}_{2\lambda_j} \cdot v^{L+}_{2\lambda_j} = 0, \\
& v^{R-}_{2\lambda_j} \cdot v^{R+}_{2\lambda_j} = 0, \\
& A_{2}{}^L_{\lambda_j} x + B_{2}{}^L_{\lambda_j} y \leqq b_{2}{}^L_{\lambda_j}, \\
& A_{2}{}^R_{\lambda_j} x + B_{2}{}^R_{\lambda_j} y \leqq b_{2}{}^R_{\lambda_j} \\
& j = 0, 1, 2, ..., l
\end{aligned} \tag{6.3d}$$

Theorem 6.2.1. *Let* $(x'^*, y'^*) = (x^*, v^{L-*}_{1\lambda_j}, v^{L+*}_{1\lambda_j}, v^{R-*}_{1\lambda_j}, v^{R+*}_{1\lambda_j}, y^*, v^{L-*}_{2\lambda_j}, v^{L+*}_{2\lambda_j}, v^{R-*}_{2\lambda_j}, v^{R+*}_{2\lambda_j})$ *be the optimal solution to bi-level decision problem (6.3), then* (x^*, y^*) *is the optimal solution to the bi-level decision problem defined by (6.1).*

Proof. By Definition 4.1.1, let the notations associated with Problem (6.1) are denoted by:

$$\begin{aligned}
S = \{(x,y) : \; & A_{i}{}^L_{\lambda_j} x + B_{i}{}^L_{\lambda_j} y \leqq b_{i}{}^L_{\lambda_j}, A_{i}{}^R_{\lambda_j} x + B_{i}{}^R_{\lambda_j} y \leqq b_{i}{}^R_{\lambda_j}, \\
& i = 1, 2, j = 0, 1, 2, ..., l\}
\end{aligned} \tag{6.4a}$$

$$\begin{aligned}
S(X) = \{x \in X : \; & \exists y \in Y, A_{i}{}^L_{\lambda_j} x + B_{i}{}^L_{\lambda_j} y \leqq b_{i}{}^L_{\lambda_j}, A_{i}{}^R_{\lambda_j} x + B_{i}{}^R_{\lambda_j} y \leqq b_{i}{}^R_{\lambda_j}, \\
& i = 1, 2, j = 0, 1, 2, ..., l\}
\end{aligned} \tag{6.4b}$$

$$S(x) = \{y \in Y : (x,y) \in S\} \tag{6.4c}$$

$$\begin{aligned}
P(x) = \{y \in Y : y \in argmin[& |c^L_{2\lambda_j} x + d^L_{2\lambda_j} \hat{y} - g^L_{F\lambda_j}|, \\
& |c^R_{2\lambda_j} x + d^R_{2\lambda_j} \hat{y} - g^R_{F\lambda_j}| : \hat{y} \in S(x)]\}
\end{aligned} \tag{6.4d}$$

$$IR = \{(x,y) : (x,y) \in S, y \in P(x)\} \tag{6.4e}$$

Problem (6.1) can be written as

$$\min\{|c^L_{1\lambda_j} x + d^L_{1\lambda_j} y - g^L_{L\lambda_j}|, |c^R_{1\lambda_j} x + d^R_{1\lambda_j} y - g^R_{L\lambda_j}| : (x,y) \in IR\}\} \tag{6.5}$$

and those of problem (6.3) are denoted by:

$$\begin{aligned} S' =\{&(x',y'): A_{i\lambda_j}^{L}x + B_{i\lambda_j}^{L}y \leqq b_{i\lambda_j}^{L}, A_{i\lambda_j}^{R}x + B_{i\lambda_j}^{R}y \leqq b_{i\lambda_j}^{R}, \\ & v_{i\lambda_j}^{L-}\cdot v_{i\lambda_j}^{L+} = 0, v_{i\lambda_j}^{R-}\cdot v_{i\lambda_j}^{R+} = 0, i = 1,2, \\ & c_{1\lambda_j}^{L}x + d_{i\lambda_j}^{L}y + v_{1\lambda_j}^{L-} - v_{1\lambda_j}^{L+} = g_{L\lambda_j}^{L}, \\ & c_{1\lambda_j}^{R}x + d_{i\lambda_j}^{R}y + v_{1\lambda_j}^{R-} - v_{1\lambda_j}^{R+} = g_{L\lambda_j}^{R}, \\ & c_{2\lambda_j}^{L}x + d_{2\lambda_j}^{L}y + v_{2\lambda_j}^{L-} - v_{2\lambda_j}^{L+} = g_{F\lambda_j}^{L}, \\ & c_{2\lambda_j}^{R}x + d_{2\lambda_j}^{R}y + v_{2\lambda_j}^{R-} - v_{2\lambda_j}^{R+} = g_{F\lambda_j}^{R}, \\ & j = 0,1,\ldots,l\} \end{aligned} \tag{6.6a}$$

$$\begin{aligned} S(X') =\{&x' \in X' : \exists y' \in Y', A_{i\lambda_j}^{L}x + B_{i\lambda_j}^{L}y \leqq b_{i\lambda_j}^{L}, A_{i\lambda_j}^{R}x + B_{i\lambda_j}^{R}y \leqq b_{i\lambda_j}^{R}, \\ & v_{i\lambda_j}^{L-}\cdot v_{i\lambda_j}^{L+} = 0, v_{i\lambda_j}^{R-}\cdot v_{i\lambda_j}^{R+} = 0, i = 1,2, \\ & c_{1\lambda_j}^{L}x + d_{i\lambda_j}^{L}y + v_{1\lambda_j}^{L-} - v_{1\lambda_j}^{L+} = g_{L\lambda_j}^{L}, \\ & c_{1\lambda_j}^{R}x + d_{i\lambda_j}^{R}y + v_{1\lambda_j}^{R-} - v_{1\lambda_j}^{R+} = g_{L\lambda_j}^{R}, \\ & c_{2\lambda_j}^{L}x + d_{2\lambda_j}^{L}y + v_{2\lambda_j}^{L-} - v_{2\lambda_j}^{L+} = g_{F\lambda_j}^{L}, \\ & c_{2\lambda_j}^{R}x + d_{2\lambda_j}^{R}y + v_{2\lambda_j}^{R-} - v_{2\lambda_j}^{R+} = g_{F\lambda_j}^{R}, \\ & j = 0,1,...l\} \end{aligned} \tag{6.6b}$$

$$S(x') =\{y' \in Y' : (x',y') \in S'\} \tag{6.6c}$$

$$P(x') =\{y' \in Y' : y' \in argmin[\hat{v}_{2\lambda_j}^{L-} + \hat{v}_{2\lambda_j}^{L+}, \hat{v}_{2\lambda_j}^{R-} + \hat{v}_{2\lambda_j}^{R+} : \hat{y'} \in S(x')]\} \tag{6.6d}$$

$$IR' =\{(x',y') : (x',y') \in S', y' \in P(x')\} \tag{6.6e}$$

Problem (6.3) can be written as

$$min\{v_{1\lambda_j}^{L-} + v_{1\lambda_j}^{L+}, v_{1\lambda_j}^{R-} + v_{1\lambda_j}^{R+} : (x',y') \in IR'\} \tag{6.7}$$

As $(x^{'*}, y^{'*})$ is the optimal solution to Problem (6.3), from (6.7), it can be obtained that, $\forall (x',y') \in IR'$, we have: $v_{1\lambda_j}^{L-} + v_{1\lambda_j}^{L+} \geqslant v_{1\lambda_j}^{L-*} + v_{1\lambda_j}^{L+*}$, and $v_{1\lambda_j}^{R-} + v_{1\lambda_j}^{R+} \geqslant v_{1\lambda_j}^{R-*} + v_{1\lambda_j}^{R+*}$.

As $c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y + v_{1\lambda_j}^{L-} - v_{1\lambda_j}^{L+} = g_{L\lambda_j}^{L}$ and $v_{1\lambda_j}^{L-}\cdot v_{1\lambda_j}^{L+} = 0$, we have

$$v_{1\lambda_j}^{L-} + v_{1\lambda_j}^{L+} = |c_{1\lambda_j}^{L}x + d_{1\lambda_j}^{L}y - g_{L\lambda_j}^{L}|,$$

and

$$v_{1\lambda_j}^{L-*} + v_{1\lambda_j}^{L+*} = |c_{1\lambda_j}^{L}x^* + d_{1\lambda_j}^{L}y^* - g_{L\lambda_j}^{L}|.$$

So:

$$|c^L_{1\lambda_j}x + d^L_{1\lambda_j}y - g^L_{L\lambda_j}| \geqslant |c^L_{1\lambda_j}x^* + d^L_{1\lambda_j}y^* - g^L_{L\lambda_j}| \tag{6.8a}$$

Similarly, we can get that:

$$|c^R_{1\lambda_j}x + d^R_{1\lambda_j}y - g^R_{L\lambda_j}| \geqslant |c^R_{1\lambda_j}x^* + d^R_{1\lambda_j}y^* - g^R_{L\lambda_j}| \tag{6.9a}$$

Now we prove that the projection of S' onto the $X \times Y$ space, denoted by $S'|_{X,Y}$, is equal to S:

On the one hand, $\forall(x, y) \in S'|_{X,Y}$, from constraints: $A^L_{i\lambda_j}x+B^L_{i\lambda_j}y \leqq b^L_{i\lambda_j}$, $A^R_{i\lambda_j}x+B^R_{i\lambda_j}y \leqq b^R_{i\lambda_j}$, $i = 1, 2$ in S', we have: $(x, y) \in S$, so $S'|_{X,Y} \subseteq S$

On the other hand, $\forall(x, y) \in S$, by (6.2), we can always find such $v^{L-}_{i\lambda_j}$, $v^{L+}_{i\lambda_j}$, $v^{R-}_{i\lambda_j}$, $v^{R+}_{i\lambda_j}$, $i = 1, 2$, which make constraints: $v^{R-}_{i\lambda_j} \cdot v^{R+}_{i\lambda_j} = 0$, $i = 1, 2$, $c^L_{1\lambda_j}x + d^L_{i\lambda_j}y + v^{L-}_{1\lambda_j} - v^{L+}_{1\lambda_j} = g^L_{L\lambda_j}$, $c^R_{1\lambda_j}x + d^R_{i\lambda_j}y + v^{R-}_{1\lambda_j} - v^{R+}_{1\lambda_j} = g^R_{L\lambda_j}$, $c^L_{2\lambda_j}x + d^L_{2\lambda_j}y + v^{L-}_{2\lambda_j} - v^{L+}_{2\lambda_j} = g^L_{F\lambda_j}$, and $c^R_{2\lambda_j}x + d^R_{2\lambda_j}y + v^{R-}_{2\lambda_j} - v^{R+}_{2\lambda_j} = g^R_{F\lambda_j}$ satisfied. Together with the inequations of $A^L_{i\lambda_j}x + B^L_{i\lambda_j}y \leqq b^L_{i\lambda_j}$, and $A^R_{i\lambda_j}x + B^R_{i\lambda_j}y \leqq b^R_{i\lambda_j}, i = 1, 2$ requested by S, we have $(x, v^{L-}_{1\lambda_j}, v^{L+}_{1\lambda_j}, y, v^{R-}_{2\lambda_j}, v^{R+}_{2\lambda_j}) \in S'$, thus $(x, y) \in S'|_{X,Y}$, $S \subseteq S'|_{X,Y}$.

So, we can prove that

$$S'|_{X,Y} = S \tag{6.10}$$

Similarly, we have

$$S(x)'|_{X,Y} = S(x) \tag{6.11a}$$

$$S(X)'|_{X,Y} = S(X) \tag{6.11b}$$

Also, from $c^L_{2\lambda_j}x + d^L_{2\lambda_j}y + v^{L-}_{2\lambda_j} - v^{L+}_{2\lambda_j} = g^L_{F\lambda_j}$ and $v^{L-}_{2\lambda_j} \cdot v^{L+}_{2\lambda_j} = 0$, we have:

$$v^{L-}_{2\lambda_j} + v^{L+}_{2\lambda_j} = |c^L_{2\lambda_j}x + d^L_{2\lambda_j}y - g^L_{F\lambda_j}| \tag{6.12}$$

Similarly, we have:

$$v^{R-}_{2\lambda_j} + v^{R+}_{2\lambda_j} = |c^R_{2\lambda_j}x + d^R_{2\lambda_j}y - g^R_{F\lambda_j}| \tag{6.13}$$

Thus:

$$\begin{aligned} P(x') = &\{y' \in Y' : y' \in argmin[|c^L_{2\lambda_j}x + d^L_{2\lambda_j}\hat{y} - g^L_{F\lambda_j}|, \\ &|c^R_{2\lambda_j}x + d^R_{2\lambda_j}\hat{y} - g^R_{F\lambda_j}| : \hat{y}' \in S(x')]\} \end{aligned} \tag{6.14}$$

From (6.10) and (6.14), we have:

$$P(x')|_{X\times Y} = P(x) \tag{6.15}$$

From (6.4e), (6.6e), (6.10) and (6.15), we have:

$$IR'|_{X\times Y} = IR \tag{6.16}$$

which means, the leaders of (6.1) and (6.3) share the same optimising space in $X \times Y$ space.

Thus, from (6.8) and (6.16) and the discussions above, we have: $\forall(x, y) \in IR$

$$|c^L_{1\lambda_j}x + d^L_{1\lambda_j}y - g^L_{L\lambda_j}| \geqslant |c^L_{1\lambda_j}x^* + d^L_{1\lambda_j}y^* - g^L_{L\lambda_j}|,$$
$$|c^R_{1\lambda_j}x + d^R_{1\lambda_j}y - g^R_{L\lambda_j}| \geqslant |c^R_{1\lambda_j}x^* + d^R_{1\lambda_j}y^* - g^R_{L\lambda_j}|$$

So, (x^*, y^*) is the optimal solution of Problem (6.1). □

By adopting a weighting method, (6.3) can be further transferred into (6.17):

$$\min_{x'\in X'} v^{L-}_{1\lambda_j} + v^{L+}_{1\lambda_j} + v^{R-}_{1\lambda_j} + v^{R+}_{1\lambda_j} \tag{6.17a}$$

$$\begin{aligned}
\text{subject to } & c^L_{1\lambda_j}x + d^L_{1\lambda_j}y + v^{L-}_{1\lambda_j} - v^{L+}_{1\lambda_j} = g^L_{L\lambda_j},\\
& c^R_{1\lambda_j}x + d^R_{1\lambda_j}y + v^{R-}_{1\lambda_j} - v^{R+}_{1\lambda_j} = g^R_{L\lambda_j},\\
& v^{L-}_{1\lambda_j}, v^{L+}_{1\lambda_j}, v^{R-}_{1\lambda_j}, v^{R+}_{1\lambda_j} \geqslant 0,\\
& v^{L-}_{1\lambda_j} \cdot v^{L+}_{1\lambda_j} = 0, && (6.17\text{b})\\
& v^{R-}_{1\lambda_j} \cdot v^{R+}_{1\lambda_j} = 0,\\
& A_1{}^L_{\lambda_j}x + B_1{}^L_{\lambda_j}y \leqq b_1{}^L_{\lambda_j},\\
& A_1{}^R_{\lambda_j}x + B_1{}^R_{\lambda_j}y \leqq b_1{}^R_{\lambda_j},\\
& \min_{y'\in Y'} v^{L-}_{2\lambda_j} + v^{L+}_{2\lambda_j} + v^{R-}_{2\lambda_j} + v^{R+}_{2\lambda_j} && (6.17\text{c})\\
& \text{subject to } c^L_{2\lambda_j}x + d^L_{2\lambda_j}y + v^{L-}_{2\lambda_j} - v^{L+}_{2\lambda_j} = g^L_{F\lambda_j},\\
& \qquad c^R_{2\lambda_j}x + d^R_{2\lambda_j}y + v^{R-}_{2\lambda_j} - v^{R+}_{2\lambda_j} = g^R_{F\lambda_j}, && (6.17\text{d})\\
& \qquad v^{L-}_{2\lambda_j}, v^{L+}_{2\lambda_j}, v^{R-}_{2\lambda_j}, v^{R+}_{2\lambda_j} \geqslant 0,
\end{aligned}$$

$$v_{2\lambda_j}^{L-} \cdot v_{2\lambda_j}^{L+} = 0,$$
$$v_{2\lambda_j}^{R-} \cdot v_{2\lambda_j}^{R+} = 0,$$
$$A_2{}_{\lambda_j}^{L} x + B_2{}_{\lambda_j}^{L} y \leqq b_2{}_{\lambda_j}^{L},$$
$$A_2{}_{\lambda_j}^{R} x + B_2{}_{\lambda_j}^{R} y \leqq b_2{}_{\lambda_j}^{R}$$
$$j = 0, 1, ..., l$$

The non-linear conditions of $v_{i\lambda_j}^{L-} \cdot v_{i\lambda_j}^{L+} = 0$, and $v_{i\lambda_j}^{R-} \cdot v_{i\lambda_j}^{R+} = 0$, $i = 1, 2$ need not be maintained if the Kuhn-Tucker approach(Shi *et al.* 2005b) together with Simplex algorithm are adopted, since only equivalence at an optimum is wanted. Further explanation can be found from (Charnes & Cooper 1961b). Thus Problem (6.17) is further transformed into:

For $(v_{1\lambda_j}^{-}, v_{1\lambda_j}^{+}), \in R^2$, $\bar{X}' \subseteq X \times R^2$, $(v_{2\lambda_j}^{-}, v_{2\lambda_j}^{+}) \in R^2$, $\bar{Y}' \subseteq Y \times R^2$, let $x = (x_1, \cdots, x_n) \in X$, $\bar{x'} = (x_1, \cdots, x_n, v_{1\lambda_j}^{-}, v_{1\lambda_j}^{+}) \in \bar{X}'$, $y = (y_1, \cdots, y_m) \in Y$, $\bar{y'} = (y_1, \cdots, y_m, v_{2\lambda_j}^{-}, v_{2\lambda_j}^{+},) \in \bar{Y}'$, and $v_{1\lambda_j}, v_{2\lambda_j} : \bar{X}' \times \bar{Y}' \to F(R)$.

$$\min_{\bar{x'} \in \bar{X}'} v_{1\lambda_j} = v_{1\lambda_j}^{-} + v_{1\lambda_j}^{+} \tag{6.18a}$$

subject to $(c_{1\lambda_j}^{L} + c_{1\lambda_j}^{R})x + (d_{1\lambda_j}^{L} + d_{1\lambda_j}^{R})y + v_{1\lambda_j}^{-} - v_{1\lambda_j}^{+} = g_{L\lambda_j}^{L} + g_{L\lambda_j}^{R}$,

$$A_1{}_{\lambda_j}^{L} x + B_1{}_{\lambda_j}^{L} y \leqq b_1{}_{\lambda_j}^{L}, \tag{6.18b}$$
$$A_1{}_{\lambda_j}^{R} x + B_1{}_{\lambda_j}^{R} y \leqq b_1{}_{\lambda_j}^{R},$$
$$\min_{\bar{y'} \in \bar{Y}'} v_{2\lambda_j} = v_{2\lambda_j}^{-} + v_{2\lambda_j}^{+} \tag{6.18c}$$

subject to $(c_{2\lambda_j}^{L} + c_{2\lambda_j}^{R})x + (d_{2\lambda_j}^{L} + d_{2\lambda_j}^{R})y + v_{2\lambda_j}^{-} - v_{2\lambda_j}^{+} = g_{F\lambda_j}^{L} + g_{F\lambda_j}^{R}$,

$$A_2{}_{\lambda_j}^{L} x + B_2{}_{\lambda_j}^{L} y \leqq b_2{}_{\lambda_j}^{L},$$
$$A_2{}_{\lambda_j}^{R} x + B_2{}_{\lambda_j}^{R} y \leqq b_2{}_{\lambda_j}^{R} \tag{6.18d}$$
$$j = 0, 1, ..., l$$

where $v_{i\lambda_j}^{-} = v_{i\lambda_j}^{L-} + v_{i\lambda_j}^{R-}$, $v_{i\lambda_j}^{+} = v_{i\lambda_j}^{L+} + v_{i\lambda_j}^{R+}$, $i = 1, 2$.

Problem (6.18) is a standard linear bi-level decision problem, which can be solved by Kuhn-Tucker approach (Shi *et al.* 2005b).

Based on discussions above, the $\lambda-$cut-based algorithm for solving FLBG problems is detailed as Algorithm 6:

Algorithm 6: A $\lambda-$cut-based algorithm for FLBG decision problems

Step 1 (Input)
Get relevant coefficients of a FLBG problem which include coefficients of problem defined by Definition 3.1.1, coefficients of $\tilde{g}_L$ and $\tilde{g}_F$, satisfactory degree: $\alpha, \varepsilon > 0$
Step 2 (Initialising)
Let $k = 1$, which is the counter to record current loop.
In (6.1), where $\lambda_j \in [\alpha, 1]$, let $\lambda_0 = \alpha$ and $\lambda_1 = 1$ respectively, then each objective will be transferred into four non-fuzzy objective functions, and each fuzzy constraint is converted into four non-fuzzy constraints.
Step 3 (Computing)
By introducing auxiliary variables $v^-_{i\lambda_j}$ and $v^+_{i\lambda_j}$, $i = 1, 2$, we get the format of (6.18).
The solution $(x, v^-_{1\lambda_j}, v^+_{1\lambda_j}, y, v^-_{2\lambda_j}, v^+_{2\lambda_j})_2$ of (6.18) is obtained by Kuhn-Tucker approach.
Step 4 (Comparison)
If $(k = 1)$ **Then**
$(x, v^-_{1\lambda_j}, v^+_{1\lambda_j}, y, v^-_{2\lambda_j}, v^+_{2\lambda_j})_1 = (x, v^-_{1\lambda_j}, v^+_{1\lambda_j}, y, v^-_{2\lambda_j}, v^+_{2\lambda_j})_2$;
goto [Step 5];
Else If $(|(x, v^-_{1\lambda_j}, v^+_{1\lambda_j}, y, v^-_{2\lambda_j}, v^+_{2\lambda_j})_2 - (x, v^-_{1\lambda_j}, v^+_{1\lambda_j}, y, v^-_{2\lambda_j}, v^+_{2\lambda_j})_1| < \varepsilon$) **Then**
goto [Step 7] ;
EndIf
Step 5 (Splitting)
Suppose there are $(L + 1)$ nodes $\lambda_j, (j = 0, 2, 4, ..., 2L)$ in the interval $[\alpha, 1]$, insert L new nodes λ_j, $(j = 1, 3..., 2L - 1)$ which satisfy:

$$\lambda_{2j+1} = (\lambda_{2j} + \lambda_{2j+2})/2, (j = 0, 1, 2, ..., L - 1).$$

Step 6 (Loop)
k=k+1;
goto [Step 3];

Step 7 (Output)
$(x, y)_2$ is obtained as a final solution.

6.3 An Example

This section employs a numerical example to show the running procedure of the proposed algorithm.

[Step 1](Input relevant coefficients):

1) Coefficients of (3.1):

Suppose the FLB problem is defined below:

$$\begin{aligned}
&\max_{x\in X} F(x,y) = \tilde{c}_1 x + \tilde{d}_1 y\\
&\text{subject to } \tilde{A}_1 x + \tilde{B}_1 y \leqslant \tilde{b}_1\\
&\qquad \min_{y\in Y} f(x,y) = \tilde{c}_2 x + \tilde{d}_2 y\\
&\qquad \text{subject to } \tilde{A}_2 x + \tilde{B}_2 y \leqslant \tilde{b}_2
\end{aligned}$$

where $x \in R, y \in R$, and $X = x \geqslant 0, Y = y \geqslant 0$.

The membership functions of the coefficients of the objective functions and the constraints of both the leader and the follower are as follows:

$$\mu_{\tilde{c}_1}(x) = \begin{cases} 0, & x < 5 \\ \frac{x^2-25}{11}, & 5 \leqq x < 8 \\ 1, & x = 6 \\ \frac{64-x^2}{28}, & 6 < x \leqslant 8 \\ 0, & x > 8 \end{cases}, \qquad \mu_{\tilde{d}_1}(x) = \begin{cases} 0, & x < 2 \\ \frac{x^2-4}{5}, & 2 \leqq x < 3 \\ 1, & x = 3 \\ \frac{25-x^2}{16}, & 3 < x \leqslant 5 \\ 0, & x > 5 \end{cases}.$$

$$\mu_{\tilde{c}_2}(x) = \begin{cases} 0, & x < -4 \\ \frac{16-x^2}{7}, & -4 \leqq x < -3 \\ 1, & x = -3 \\ \frac{x^2-1}{8}, & -3 < x \leqslant -1 \\ 0, & x > -1 \end{cases}, \qquad \mu_{\tilde{d}_2}(x) = \begin{cases} 0, & x < 5 \\ \frac{x^2-25}{11}, & 5 \leqq x < 6 \\ 1, & x = 6 \\ \frac{64-x^2}{28}, & 6 < x \leqslant 8 \\ 0, & x > 8 \end{cases}.$$

$$\mu_{\tilde{A}_1}(x) = \begin{cases} 0, & x < -2 \\ \frac{4-x^2}{3}, & -2 \leqq x < -1 \\ 1, & x = -1 \\ \frac{x^2-0.25}{0.75}, & -1 < x \leqslant -0.5 \\ 0, & x > -0.5 \end{cases}, \qquad \mu_{\tilde{B}_1}(x) = \begin{cases} 0, & x < 2 \\ \frac{x^2-4}{5}, & 2 \leqq x < 3 \\ 1, & x = 3 \\ \frac{25-x^2}{16}, & 3 < x \leqslant 5 \\ 0, & x > 5 \end{cases}.$$

$$\mu_{\tilde{A}_2}(x) = \begin{cases} 0, & x < 0.5 \\ \frac{x^2-0.25}{0.75}, & 0.5 \leqq x < 1 \\ 1, & x = 1 \\ \frac{4-x^2}{3}, & 1 < x \leqslant 2 \\ 0, & x > 2 \end{cases}, \qquad \mu_{\tilde{B}_2}(x) = \begin{cases} 0, & x < 2 \\ \frac{x^2-4}{5}, & 2 \leqq x < 3 \\ 1, & x = 3 \\ \frac{25-x^2}{16}, & 3 < x \leqslant 5 \\ 0, & x > 5 \end{cases}.$$

$$\mu_{\tilde{b}_1}(x) = \begin{cases} 0, & x < 19 \\ \frac{x^2-361}{80}, & 19 \leqq x < 21 \\ 1, & x = 21 \\ \frac{625-x^2}{184}, & 21 < x \leqslant 25 \\ 0, & x > 25 \end{cases}, \quad \mu_{\tilde{b}_2}(x) = \begin{cases} 0, & x < 25 \\ \frac{x^2-625}{104}, & 25 \leqq x < 27 \\ 1, & x = 27 \\ \frac{961-x^2}{232}, & 27 < x \leqslant 31 \\ 0, & x > 31 \end{cases}.$$

2) The membership functions for the fuzzy goals of $\tilde{g}_L$ and $\tilde{g}_F$ are:

$$\mu_{\tilde{g}_L}(x) = \begin{cases} 0, & x < 15 \\ \frac{x^2-225}{175}, & 15 \leqq x < 20 \\ 1, & x = 20 \\ \frac{900-x^2}{500}, & 20 < x \leqslant 30 \\ 0, & x > 30 \end{cases}, \mu_{\tilde{g}_F}(x) = \begin{cases} 0, & x < 4 \\ \frac{x^2-16}{48}, & 4 \leqq x < 8 \\ 1, & x = 8 \\ \frac{225-x^2}{161}, & 8 < x \leqslant 15 \\ 0, & x > 15 \end{cases}.$$

3) Satisfactory degree: $\alpha = 0.2$

4) $\varepsilon = 0.01$

[Step 2](Initialise):

let k=1. Associated with this example, the corresponding λ-cut set based FLBG problem is:

$$\begin{aligned}
&\min_{x \in X} |\sqrt{11\lambda + 25}x + \sqrt{5\lambda + 4}y - \sqrt{175\lambda + 225}| \\
&\min_{x \in X} |\sqrt{64 - 28\lambda}x + \sqrt{25 - 16\lambda}y - \sqrt{900 - 500\lambda}| \\
&\text{subject to } -\sqrt{4 - 3\lambda}x + \sqrt{5\lambda + 4}y \leqslant \sqrt{80\lambda + 36} \\
&\qquad -\sqrt{0.75\lambda + 0.25}x + \sqrt{25 - 16\lambda}y \leqslant \sqrt{625 - 184\lambda} \\
&\qquad \min_{y \in Y} | -\sqrt{16 - 7\lambda}x + \sqrt{11\lambda + 25}y - \sqrt{48\lambda + 16}| \\
&\qquad \min_{y \in Y} | -\sqrt{8\lambda + 1}x + \sqrt{64 - 28\lambda}y - \sqrt{225 - 161\lambda}| \\
&\qquad \text{subject to } \sqrt{0.75\lambda + 0.25}x + \sqrt{5\lambda + 4}y \leqslant \sqrt{104\lambda + 625} \\
&\qquad \qquad \sqrt{4 - 3\lambda}x + \sqrt{25 - 16\lambda}y \leqslant \sqrt{961 - 232\lambda}
\end{aligned}$$

where $\lambda \in [0.2, 1]$.

Referring to the algorithm, only $\lambda_0 = 0.2$ and $\lambda_1 = 1$ are considered initially. Thus four non-fuzzy objective functions and four non-fuzzy constraints for the leader and follower are generated respectively:

$$\begin{aligned}
&\min_{x\in X} |5.2x + 2.2y - 16.1| \\
&\min_{x\in X} |6x + 3y - 20| \\
&\min_{x\in X} |7.6x + 4.7y - 28.3| \\
&\min_{x\in X} |6x + 3y - 20| \\
&\text{subject to } -1.8x + 2.2y \leqslant 19.4 \\
&\qquad -x + 3y \leqslant 21 \\
&\qquad -0.6x + 4.7y \leqslant 24.3 \\
&\qquad -x + 3y \leqslant 21
\end{aligned}$$

$$\begin{aligned}
&\min_{y\in Y} |-3.8x + 5.2y - 5.1| \\
&\min_{y\in Y} |-3x + 6y - 8| \\
&\min_{y\in Y} |1.6x + 7.6y - 13.9| \\
&\min_{y\in Y} |-3x + 6y - 8| \\
&\text{subject to } 0.6x + 2.2y \leqslant 25.4 \\
&\qquad x + 3y \leqslant 27 \\
&\qquad 1.8x + 4.7y \leqslant 30.2 \\
&\qquad x + 3y \leqslant 27
\end{aligned}$$

[Step 3](Compute):

By introducing auxiliary variables $v_i^-, v_i^+, i = 1, 2$, we get:

$$\begin{aligned}
&\min_{(x,v_1^-,v_1^+)\in\bar{X}'} v_1^- + v_1^+ \\
&\text{subject to } 24.8x + 12.9y + v_1^- - v_1^+ = 84.4, \\
&\qquad -1.8x + 2.2y \leqslant 19.4 \\
&\qquad -x + 3y \leqslant 21 \\
&\qquad -0.6x + 4.7y \leqslant 24.3 \\
&\qquad -x + 3y \leqslant 21
\end{aligned}$$

$$\min_{(y,v_2^-,v_2^+)\in \bar{Y}'} v_2^- + v_2^+$$

$$\text{subject to } -11.4x + 24.8y + v_2^- - v_2^+ = 35,$$
$$0.6x + 2.2y \leqslant 25.4$$
$$x + 3y \leqslant 7$$
$$1.8x + 4.7y \leqslant 30.2$$
$$x + 3y \leqslant 27$$

Using extended Branch-and-Bound approach (Shi *et al.* 2006), current solution is (2.15366,0,0, 2.39243,0,0).

[Step 4](Compare) : Because $k = 1$, goto **[Step 5]**

[Step 5](Split): By inserting a new node $\lambda_1 = (0.2 + 1)/2 = 0.6$, there are total three nodes of $\lambda_0 = 0.2, \lambda_1 = 0.6$ and $\lambda_2 = 1$. Then total twelve non-fuzzy objective functions for the leader and follower together with twelve non-fuzzy constraints for the leader and follower respectively are generated.

[Step 6](Loop): $k = 1 + 1 = 2$, goto **[Step 3]**, and current solution of (2.17093, 0, 0, 2.41756, 0, 0) is obtained. As $|2.15366 - 2.17093| + |2.39243 - 2.41756| = 0.04 > \varepsilon = 0.01$, the algorithm keep going until the solution of (2.13535, 0, 0, 2.42797, 0, 0) is obtained. The computing results are listed in Table 6.1.

Table 6.1: Summary of the running solutions

k	x	y	$v_{1\lambda}^+$	$v_{1\lambda}^-$	$v_{2\lambda}^+$	$v_{2\lambda}^-$
1	2.15366	2.39243	0	0	0	0
2	2.17093	2.41756	0	0	0	0
3	2.12393	2.43436	0	0	0	0
4	2.13535	2.42797	0	0	0	0

[Step 7](Output): As $|2.12393 - 2.13535| + |2.43436 - 2.42797| = 0.0178 < \varepsilon = 0.02$, $(x^*, y^*) = (2.1354, 2.4280)$ is the final solution of this FLBG decision problem. The objectives obtained for the leader and the follower under $(x^*, y^*) = (2.1354, 2.4280)$ are:

$$\begin{cases} F(x^*, y^*) = F(2.1354, 2.4280) = 2.1354\tilde{c_1} + 2.4280\tilde{d_1} \\ f(x^*, y^*) = F(2.1354, 2.4280) = 2.1354\tilde{c_2} + 2.4280\tilde{d_2} \end{cases}$$

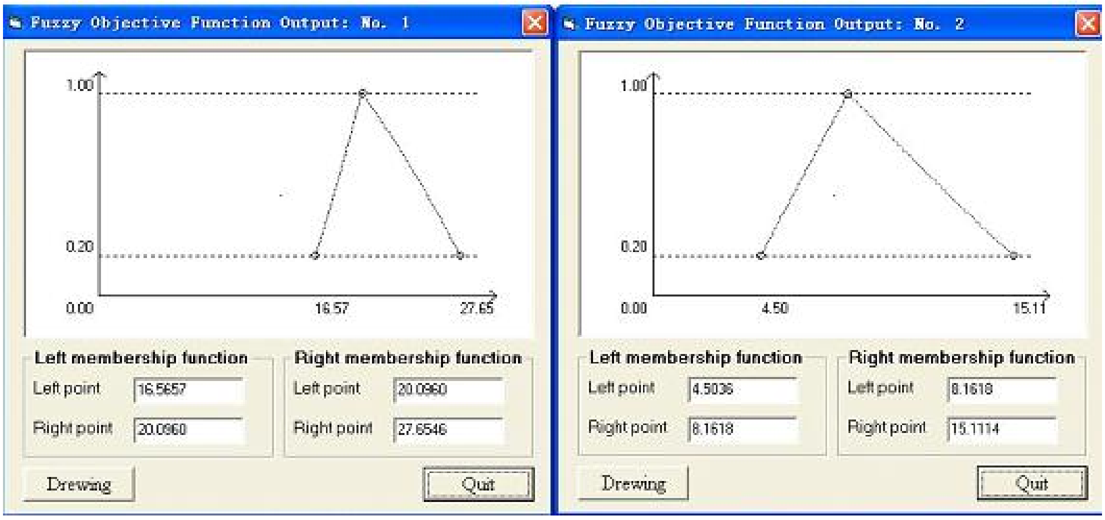

Figure 6.1: Membership functions of $F(x^*, y^*)$ and $f(x^*, y^*)$

and their membership functions are shown in Figure 6.1.

Above example illustrated the detailed working process of the proposed algorithm.

6.4 Summary

This chapter studies FLBG decision problems. In a bi-level decision model, the leader and/or the follower may wish that their objectives attain to some goals, which are different from simple optimisation problems. An FLBG model has been proposed, and a $\lambda-$cut-based algorithm for solving FLBG decision problems has been developed. A numerical example is employed to further explain this algorithm. This algorithm is implemented in the fuzzy bi-level decision support system which will be discussed in Chapter 8.

7 Nash-equilibrium-based Concept and PSO for General Bi-level Decision Making

This chapter studies general bi-level decision making, which means, the objective and constraint functions for leader(s) and follower(s) do not have to be linear. We classify general bi-level decision problems into multi-leader one-follower bi-level (MLOFB) decision problems, one-leader multi-follower bi-level (OLMFB) decision problems, and multi-leader multi-follower bi-level (MLMFB) decision problems. Then we give the corresponding definitions and models, based on which, PSO-based algorithms are developed to solve them.

7.1 Multi-leader One-follower Bi-level Decision Making

7.1.1 Definitions and Models

Definition 7.1.1. An MLOFB decision problem is defined as :

For $x_1 \in X_1 \subseteq R^{m_1}, \ldots, x_L \in X_L \subseteq R^{m_L}, y \in Y \subseteq R^n, X = X_1 \times X_2 \times \ldots \times X_L$, $F_i, f : X \times Y \to R^1$, $L \geqslant 2$

$$\min_{x_i \in X_i} F_i(x_1, \ldots, x_L, y), i = 1, \ldots, L \tag{7.1a}$$

$$\text{subject to } g_{ik}(x_1, \ldots, x_L, y) \leqslant 0, k = 1, \ldots, i_k, i = 1, \ldots, L \tag{7.1b}$$

$$\min_{y \in Y} f(x_1, \ldots, x_L, y) \tag{7.1c}$$

$$\text{subject to } q_j(x_1, \ldots, x_L, y) \leqslant 0, j = 1, \ldots, p \tag{7.1d}$$

In this MLOFB model defined above, there are L leaders, one follower, and both the leaders and the follower have their individual control variables, objectives and constraints.

Relating to this MLOFB model, following are some basic terms and symbols.

Definition 7.1.2.

(1) Constraint region of MLOFB problem (7.1):

$$\begin{aligned} S \triangleq & \{(x_1, \dots, x_L, y) : (x_1, \dots, x_L) \in X, y \in Y, g_{ik}(x_1, \dots, x_L, y) \leqslant 0, \\ & q_j(x_1, \dots, x_L, y) \leqslant 0\}, i = 1, \dots, L, k = 1, \dots, i_k, j = 1, \dots, p \end{aligned}$$

It refers to all possible combination of choices that the leaders and the follower may make.

(2) Projection of S onto the leaders' decision space:

$$\begin{aligned} S(X) \triangleq & \{(x_1, \dots, x_L) \in X : \exists y \in Y, g_{ik}(x_1, \dots, x_L, y) \leqslant 0, \\ & q_j(x_1, \dots, x_L, y) \leqslant 0, \quad i = 1, \dots, L, k = 1, \dots i_k, j = 1, \dots, p\} \end{aligned}$$

(3) Feasible set for the follower:

$\forall x = (x_1, \dots, x_L) \in X,$

$$Q(x) \triangleq \{y \in Y : q_1(x_1, \dots, x_L, y) \leqslant 0, \dots, q_p(x_1, \dots, x_L, y) \leqslant 0\}$$

(4) The follower's rational reaction set:

for $x \in S(x)$

$$P(x) \triangleq \{y \in Y : y \in argmin[f(x, \hat{y}) : \hat{y} \in Q(x)]\}$$

where $argmin[f(z) : z \in Z] = \{z^* \in Z | f(z^*) \leqslant f(z), z \in Z\}$.

The follower observes the leaders' action and reacts by selecting y from his or her feasible set to minimise his or her objective function.

(5) Inducible region:

$$IR \triangleq \{(x, y) : (x, y) \in G, y \in P(x)\}$$

which represents the set over which leaders may optimise their objectives.

In terms of above notation, an MLOFB decision problem can be written as:

$$min\{F_i(x, y) : (x, y) \in IR\}, i = 1, \dots, L \tag{7.2}$$

In a real world MLOFB decision problem, the leaders have their individual variables, objective and constraints. However, a decision from any particular leader will be inevitably made by guessing other leaders' strategies. In this case, the upper level optimisation problem is a kind of game problem, and the whole problem becomes a multi-leader-one-follower bi-level game (MLBG) problem. This kind of MLBG optimisation problem is different from common single objective, multi-objective or bi-level optimisation problems. The objective of an MLBG problem is to search for equilibria solutions. An MLBG problem is also different from the conventional game mode which has no hierarchical structure, and whose main concern is to search for a Nash equilibrium solution in some sense. Addressed to MLBG decision problems, we need to give the definition of a solution, and then develop a method by this solution definition.

Definition 7.1.3. A tuple $(x_1^*, \ldots, x_L^*) \in IR$ is said to be a generalised Nash equilibrium optimal solution of an MLOFB decision problem if (x_i^*, y^*) satisfy the following inequality:

$$F_i(x_1^*, \ldots, x_{i-1}^*, x_{i+1}^*, y^*) \leqslant F_i(x_1^*, \ldots, x_{i-1}^*, x_i, x_{i+1}^*, y^*), i = 1, \ldots, L$$

To obtain the Nash equilibrium solution for an MLOFB decision problem, we define the optimal reaction from a leader as follows:

If the $i-$th leader knows the strategies x_{-i} of other leaders, then the optimal reaction of the $i-$th leader is represented by a mapping:

$$(x_i) = r_i(x_{-i}),$$

that solves the sub problem:

$$\min_{x_i \in X_i} F_i(x_1, \ldots, x_L, y) \tag{7.3a}$$

$$\text{subject to } g_{ik}(x_1, \ldots, x_L, y) \leqslant 0, k = 1, \ldots, i_k \tag{7.3b}$$

$$\min_{y \in Y} f(x_1, \ldots, x_L, y) \tag{7.3c}$$

$$\text{subject to } q_j(x_1, \ldots, x_L, y) \leqslant 0, j = 1, \ldots, p \tag{7.3d}$$

Our aim is to make the choice from every leader as close to the rational reaction as possible. A solution is supposed to be the Nash equilibrium when we reach a point where the choices from all leaders are close enough to their corresponding rational reactions. Based on this strategy, we redefine an MLOFB problem (7.1) as:

Definition 7.1.4.

$$\min_{x_i \in X_i} F(x_1, \ldots, x_L, y) = \sum_{i=1}^{L} |x_i, -r_i(x_{-i})|, i = 1, \ldots, L \tag{7.4a}$$

$$\text{subject to } g_{ik}(x_1, \ldots, x_L, y) \leqslant 0, k = 1, \ldots, i_k, i = 1, \ldots, L \tag{7.4b}$$

$$\min_{y \in Y} f(x_1, \ldots, x_L, y) \tag{7.4c}$$

$$\text{subject to } q_j(x_1, \ldots, x_L, y) \leqslant 0, j = 1, \ldots, p \tag{7.4d}$$

7.1.2 A PSO-based Algorithm

In this section, we use the strategy adopted in the PSO method to develop a PSO-based algorithm to reach a Nash equilibrium solution for an MLOFB decision problem.

Figure 7.1 outlines the main structure of this algorithm. We first sample the leaders-controlled variables to get some candidate choices for leaders. Then, we use the PSO method together with the stretching technology (Parsopoulos & Vrahatis 2002) to get the follower's response for every choice from leaders. Thus a pool of candidate solutions for both the leaders and the follower is formed. By pushing every solution pair moving towards current best ones, the whole solution pool is updated. Once a solution is reached for the leader, we use the stretching technology (Parsopoulos & Vrahatis 2002) to escape the local optimisation. We repeat this procedure by a pre-defined count and reach a final solution.

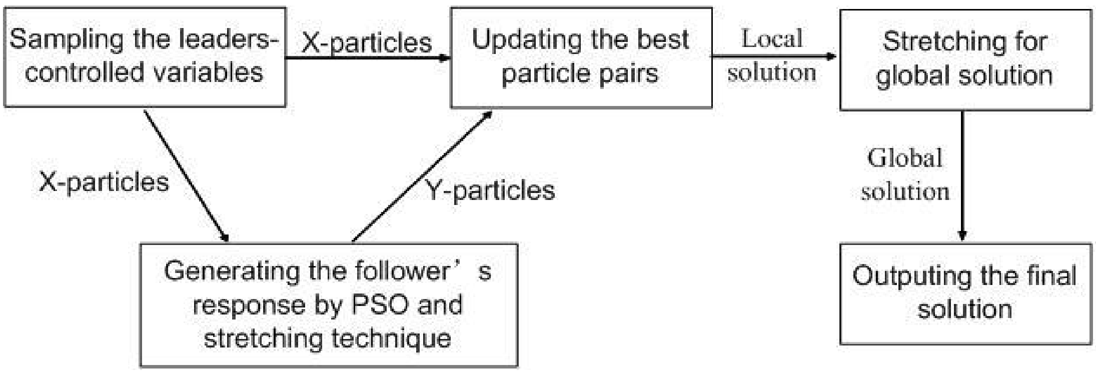

Figure 7.1: The outline of the PSO-based algorithm for MLOFB problems

The detailed algorithm is specified in Algorithm 7 and Algorithm 8. Notations used in the algorithms are detailed in Table 7.1.

Algorithm 7: Generate the response from a follower

Step 1 Input the values of $x_1, \ldots, x_L$ from the L leaders;
Step 2 Sample N_f candidates y_i and the corresponding velocities v_{y_i}, $i = 1, \ldots, N_f$;
Step 3 Initiate the follower's loop counter $k_f = 0$;
Step 4 Record the best particles p_{y_i} and y^* from P_{y_i}, $i = 1, 2, \ldots, N_f$:
if y_i satisfy (7.4d) AND $f(x_1, \ldots, x_L, y_i) < f(x_1, \ldots, x_L, p_{y_i})$, then $p_{y_i} = y_i$.
For each p_{y_i}, if $f(x_1, \ldots, x_L, p_{y_i}) < f(x_1, \ldots, x_L, y^*)$, then $y^* = p_{y_i}$;
Step 5 Update velocities and positions using
$v_{y_i}^{K+1} = w_f v_{y_i}^K + c_f r_{1f}^K (p_{y_i} - y_i^K) + c_l r_{2f}^K (y^{*K} - y_i^K)$
$y_i^{K+1} = y_i^K + v_{y_i}^{K+1}$
Step 6 $k_f = k_f + 1$;
Step 7 If $k_f \geqslant MaxK_f$ **OR** the solution changes for several consecutive generations are small enough, then we use Stretching technology to obtain the global solution and goto Step 8. Otherwise goto Step 5;
Step 8 Output y^* as the response from the follower.

Algorithm 8: Generate optimal strategies for leaders

Step 1 Sample N_l particles of $(x_{11}, \ldots, x_{L1}), \ldots, (x_{1N_l}, \ldots, x_{LN_l})$, and the corresponding velocities $(v_{11}, \ldots, v_{L1}), \ldots, (v_{1N_l}, \ldots, v_{LN_l})$;
Step 2 Initiate the leaders' loop counter $k_l = 0$;
Step 3 For $k-$particle, $k = 1, \ldots, N_l$, calculate the optimal response $r_i(x_{-i}), i = 1, \ldots, L$;
Step 3.1 Sample N_l particles x_i within the constraints of x_i;
Step 3.2 By calling Algorithm 7, we calculate the rational response from the follower;
Step 3.3 Using PSO technique, we obtain $r_i(x_{-i}), i = 1, \ldots, L$;
Step 4 Calculate the function value of every particle by (7.4a);
Step 5 Record $p_{x_{ij}}$, x_{ij}^*, $j = 1, \ldots, N_l$:
For each $x_{ij}, j = 1, \ldots, N_l$, if x_{ij} satisfy (7.4c) AND $F(x_{ij}) < F(p_{x_{ij}})$, then $p_{x_{ij}} = x_{ij}$;
For each $p_{x_{ij}}$, if $F(p_{x_{ij}}) < F(x_i^*)$, then $x_i^* = p_{x_{ij}}$;
Step 6 Move particles by best positions:
$v_{x_{ij}}^{K+1} = w_l v_{x_{ij}}^K + c_l r_{1l}^K (p_{x_{ij}} - x_{ij}^K) + c_l r_{2l}^K (x_{ij}^{*K} - x_{ij}^K)$
$x_{ij}^{K+1} = x_{ij}^K + v_{x_{ij}}^{K+1}$
Step 7 $k_l = k_l + 1$;
Step 8 If $\sum_{i=1}^{L} |x_i - r_i(x_i)| \leqslant \varepsilon$ **OR** $k_l > MaxK_l$, we use the Stretching technology to current leaders' solutions to obtain the global solution. Otherwise goto Step 3.

Table 7.1: The explanation of some notations in Algorithms 7 and 8

N_l	the number of candidate solutions (particles) for the leaders
N_f	the number of candidate solutions (particles) for the follower
x_{ij}	the $j-$th candidate solutions for the controlling variables from $i-$th leader
$p_{x_{ij}}$	the best previously visited position of x_{ij}
x_i^*	current best one for particle $x_{ij}, j = 1, \ldots, N_l$
$v_{x_{ij}}$	the velocity of x_{ij}
k_l	current iteration number for the upper-level problem
y_i	$i-$th candidate solution for the controlling variables from the follower
p_{y_i}	the best previously visited position of y_i
y^*	current best one for particle y
v_{y_i}	the velocity of y_i
k_f	current iteration number for the lower-level problem
$MaxK_l$	the predefined max iteration number for k_l
$MaxK_f$	the predefined max iteration number for k_f

7.2 One-leader Multi-follower Bi-level Decision Making

7.2.1 Definitions and Models

Definition 7.2.1. A one-leader multi-follower bi-level (OLMFB) decision problem is defined as :

For $x \in X \subseteq R^m \subseteq R^m$, $y_1 \in Y_1 \subseteq R^{n_1}, \ldots, y_L \in Y_L \subseteq R^{n_L}$, $Y = Y_1 \times Y_2 \times \ldots \times Y_L$, $F, f_i : X \times Y \to R^1$, $L \geqslant 2$

$$\min_{x \in X} F(x, y_1, \ldots, y_L) \tag{7.5a}$$

$$\text{subject to } g_j(x, y_1, \ldots, Y_L) \leqslant 0, j = 1, \ldots, p \tag{7.5b}$$

$$\min_{y_i \in Y_i} f_i(x, y_1, \ldots, y_L), i = 1, \ldots, L \tag{7.5c}$$

$$\text{subject to } q_{ik}(x, y_1, \ldots, y_L) \leqslant 0, k = 1, \ldots, i_k, i = 1, \ldots, L \tag{7.5d}$$

In this OLMFB model defined above, there are one leader and L followers. Both a leader and followers have their individual control variables, objectives and constraints.

Relating to this OLMFB model, the following are some basic terms and symbols:

Definition 7.2.2.

(1) Constraint region of OLMFB problem (7.5):

$$S \triangleq \{(x, y_1, \ldots, y_L) : x \in X, (y_1, \ldots, y_L) \in Y, j = 1, \ldots, p,$$
$$g_j(x, y_1, \ldots, y_L) \leqslant 0, q_{ik}(x, y_1, \ldots, y_L) \leqslant 0, k = 1, \ldots, i_k, i = 1, \ldots, L$$

It refers to all possible combination of choices that the leader and followers may make.

(2) Projection of S onto the leader's decision space:

$$S(X) \triangleq \{x \in X : \exists y \in Y, g_j(x, y_1, \ldots, y_L) \leqslant 0, j = 1, \ldots, p,$$
$$q_{ik}(x, y_1, \ldots, Y_L) \leqslant 0, k = 1, \ldots, i_k, i = 1, \ldots, L\}$$

(3) Feasible set for the followers:

$$\forall x \in X, Q(x) \triangleq \{y = (y_1, \ldots, y_L) \in Y : q_{ik}(x, y_1, \ldots, y_L) \leqslant 0,$$
$$k = 1, \ldots, i_k, i = 1, \ldots, L, \}$$

(4) The followers' rational reaction set:

$$\text{for } x \in S(x), P(x) \triangleq \{y \in Y : y \in argmin[f(x, \hat{y}) : \hat{y} \in Q(x)]\}$$

where $argmin[f(z) : z \in Z] = \{z^* \in Z | f(z^*) \leqslant f(z), z \in Z\}$.

The followers observe the leader's action and react by selecting y from their feasible set to minimise their objective functions.

(5) Inducible region:

$$IR \triangleq \{(x, y) : (x, y) \in G, y \in P(x)\}$$

which represents the set over which a leader may optimise his or her objective.

In terms of above notation, an OLMFB decision problem can be written as:

$$\min\{F(x, y) : (x, y) \in IR\} \tag{7.6}$$

7.2.2 A PSO-based Algorithm

Like MLOFB decision problems, in an OLMFB decision problem, when a follower has to make his or her decision based on estimating or guessing optimisation strategies from other followers, the "game playing relationship" among them needs to be considered to calculate an overall solution. However, if a follower's choice is influenced by only the leader and his or her own concerns, the problem is then a common OLMFB problem, addressed to which, a PSO-based algorithm is developed in this section.

The detailed algorithm is specified in Algorithm 9 and Algorithm 10. Notations used in the algorithm are detailed in Table 7.2.

Algorithm 9: Generate the response from a follower
Step 1 Input coefficients of x; **Step 2** Sample N_f candidates y_{il} and the corresponding velocities $v_{y_{il}}$, $i = 1, \ldots, N_f$; **Step 3** Initiate the follower's loop counter $k_f = 0$; **Step 4** Record the best particles $p_{y_{il}}$ and y_l^* from y_{il}, $i = 1, \ldots, N_f$: if y_{il} satisfy (7.5d) AND $f_l(x, y_{il}) < f_l(x, p_{y_{il}})$, then $p_{y_{il}} = y_{il}$; For each $p_{y_{il}}$, if $f_l(x, p_{y_{il}}) < f_l(x, y_l^*)$, then $y_l^* = p_{y_{il}}$; **Step 5** Update velocities and positions using $v_{y_i}^{K+1} = w_f v_{y_i}^K + c_l r_{1f}^K (p_{y_i} - y_i^K) + c_l r_{2f}^K (y^{*K} - y_i^K)$ and $y_i^{K+1} = y_i^K + v_{y_i}^{K+1}$ **Step 6** $k_f = k_f + 1$; **Step 7** If $k_f \geqslant MaxK_f$ **OR** the solution changes for several consecutive generations are small enough, then we use the stretching technology to obtain the global solution and goto Step 8. Otherwise goto Step 5; **Step 8** Output y_l^* as the response form the $l-$th follower.

Algorithm 10: Generate the optimal strategy for a leader
Step 1 Sample N_l particles of $x_1, \ldots, x_{N_l}$, and the corresponding velocities $v_1, \ldots, v_{N_l}$; **Step 2** Initiate the leaders' loop counter $k_l = 0$; **Step 3** For $k-$th particle, $k = 1, \ldots, N_l$, calculate the optimal responses from $l - th$ follower by Algorithm 9, $l = 1, \ldots, L$; **Step 4** Calculate the objective value of every particle by (7.5a); **Step 5** Record p_{x_i}, x^*, $i = 1, \ldots, N_l$: For each $x_i, i = 1, \ldots, N_l$, if x_i satisfies (7.5b) AND $F(x_i) < F(p_{x_i})$, then $p_{x_i} = x_i$; For each p_{x_i}, if $F(p_{x_i}) < F(x^*)$, then $x^* = p_{x_i}$; **Step 6** Move particles by best positions: $v_{x_i}^{K+1} = w_l v_{x_i}^K + c_l r_{1l}^K (p_{x_i} - x_i^K) + c_l r_{2l}^K (x_i^{*K} - x_i^K)$ $x_i^{K+1} = x_i^K + v_{x_i}^{K+1}$ **Step 7** $k_l = k_l + 1$; **Step 8** If $\sum_{i=1}^{m} \lvert x_i^{K+1} - x_i^K \rvert \leqslant \varepsilon$ **OR** $k_l > MaxK_l$, we use Stretching technology to current leaders' solutions to obtain the global solution. Otherwise, goto Step 3.

Table 7.2: The explanation of some notations in Algorithms 9 and 10

N_l	the number of candidate solutions (particles) for the leader
N_f	the number of candidate solutions (particles) for each follower
x_j	the $j-$th candidate solution for the leader
p_{x_i}	the best previously visited position of x_j
x^*	current best one for particle $x_j, j = 1, \ldots, n$
v_{x_j}	the velocity of x_j
k_l	current iteration number for the upper-level problem
y_{ij}	$i-$th candidate solution for the $j-$follower
$p_{y_{ij}}$	the best previously visited position of y_{ij}
y_i^*	current best one for particle $y_{ij}, i = 1, \ldots, N_f$
k_f	current iteration number for the lower-level problem
$MaxK_l$	the predefined max iteration number for k_l
$MaxK_f$	the predefined max iteration number for k_f

7.3 Multi-leader Multi-follower Bi-level Decision Making

7.3.1 Definitions and Models

Definition 7.3.1. A multi-leader multi-follower bi-level (MLMFB) decision problem is defined as :

For $x_1 \in X_1 \subseteq R^{m_1}, \ldots, x_L \in X_L \subseteq R^{m_L}, y_1 \in Y_1 \subseteq R^{n_1}, \ldots, y_M \in Y_M \subseteq R^{n_M}$ $X = X_1 \times X_2 \times \ldots \times X_L, y \in Y = Y_1 \times Y_2 \times \ldots \times Y_M$ $F_i, f_j : X \times Y \to R^1$, $L \geqslant 2$, $M \geqslant 2$, $i = 1, \ldots, L, j = 1, \ldots, M$.

$$\min_{x_i \in X_i} F_i(x_1, \ldots, x_L, y_1, \ldots, y_M), \quad i = 1, \ldots, L \tag{7.7a}$$

$$\text{subject to } g_{ik}(x_1, \ldots, x_L, y_1, \ldots, y_M) \leqslant 0, \quad k = 1, \ldots, i_k, i = 1, \ldots, L \tag{7.7b}$$

$$\min_{y_j \in Y_j} f_j(x_1, \ldots, x_L, y_1, \ldots, y_M), \quad j = 1, \ldots, M \tag{7.7c}$$

$$\text{subject to } q_{jk}(x_1, \ldots, x_L, y_1, \ldots, y_M) \leqslant 0, \tag{7.7d}$$

$$k = 1, \ldots, j_k, j = 1, \ldots, M$$

In this MLMFB model defined above, there are L leaders, M follower, and both the leaders and the followers have their individual control variables, objectives and constraints.

Relating to this MLMFB model, following are some basic terms and symbols:

Definition 7.3.2.

(1) Constraint region of MLB problem (7.7):

$$S \triangleq \{(x_1,\ldots,x_L,y_1,\ldots,y_M) : (x_1,\ldots,x_L) \in X, (y_1,\ldots,y_M) \in Y,$$
$$g_{ik}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0, q_{jk}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0,$$
$$j = 1,\ldots,M, k = 1,\ldots,j_k\}$$

It refers to all possible combination of choices that the leaders and followers may make.

(2) Projection of S onto the leaders' decision space:

$$S(X) \triangleq \{(x_1,\ldots,x_L) \in X : g_{ik}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0,$$
$$q_{jk}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0,$$
$$i = 1,\ldots,L, k = 1,\ldots,i_k, j = 1,\ldots,M, k = 1,\ldots,j_k\}$$

(3) Feasible set for the followers: $\forall x = (x_1,\ldots,x_L) \in X$,

$$Q(x) \triangleq \{(y_1,\ldots,y_M) \in Y : q_{jk}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0,$$
$$j = 1,\ldots,M, k = 1,\ldots,j_k\}$$

(4) The followers' rational reaction set: for $x \in S(x), y = (y_1,\ldots,y_M)$,

$$P(x) \triangleq \{(y_1,\ldots,y_M) \in Y : y \in argmin[f(x,\hat{y}) : \hat{y} \in Q(x)]\}$$

where $argmin[f(z) : z \in Z] = \{z^* \in Z | f(z^*) \leqslant f(z), z \in Z\}$.

The followers observe the leaders' action and react by selecting y from their feasible sets to minimise their objective functions.

(5) Inducible region:

$$IR \triangleq \{(x,y) : (x,y) \in G, y \in P(x)\}$$

which represents the set over which leaders may optimise their objectives.

In terms of above notation, an MLMFB decision problem can be written as:

$$\min\{F_i(x,y) : (x,y) \in IR, i = 1,\ldots,L\} \tag{7.8}$$

Now we define a Nash equilibrium optimal solution for an MLMFB decision problem as follows:

Definition 7.3.3. A tuple $(x_1^*,\ldots,x_L^*) \in IR$ is said to be a generalised Nash equilibrium optimal solution of an MLMFB decision problem if $(x_1^*,\ldots,x_L^*,y_1^*,\ldots,y_M^*)$ satisfy the following inequality:

$$F_i(x_1^*,\ldots,x_{i-1}^*,x_{i+1}^*,y_1^*,\ldots,y_M^*) \leqslant F_i(x_1^*,\ldots,x_{i-1}^*,x_i,x_{i+1}^*,y_1^*,\ldots,y_M^*),$$
$$i = 1,\ldots,L$$

To obtain the Nash equilibrium solution for an MLMFB decision problem, we define the optimal reaction from a leader as follows:

If the $i-$th leader knows the strategies x_{-i} of other leaders, then the optimal reaction of the $i-$th leader is represented by a mapping:

$$(x_i) = r_i(x_{-i}),$$

that solves the sub problem:

$$\min_{x_i \in X_i} F_i(x_1,\ldots,x_L,y_1,\ldots,y_M) \tag{7.9a}$$

$$\text{subject to } g_{ik}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0, k = 1,\ldots,i_k, i = 1,\ldots,L \tag{7.9b}$$

$$\min_{y_j \in Y_j} f_j(x_1,\ldots,x_L,y_1,\ldots,y_M), j = 1,\ldots,M \tag{7.9c}$$

$$\text{subject to } q_{jk}(x_1,\ldots,x_L,y_1,\ldots,y_M) \leqslant 0, \tag{7.9d}$$
$$k = 1,\ldots,j_k, j = 1,\ldots,M$$

Our aim is to make the choice from every leader as close to the rational reaction as possible. A solution is supposed to be the Nash equilibrium when we reach a point where the choices from all leaders are close enough to their corresponding rational reactions. Based on this strategy, we redefine an MLMFB problem (7.7) as:

Definition 7.3.4.

$$\min_{x_i \in X_i} F(x_1, \ldots, x_L, y_1, \ldots, y_M) = \sum_{i=1}^{L} |x_i, -r_i(x_{-i})|, i = 1, \ldots, L \tag{7.10a}$$

$$\text{subject to } g_{ik}(x_1, \ldots, x_L, y_1, \ldots, y_M) \leqslant 0, k = 1, \ldots, i_k, i = 1, \ldots, L \tag{7.10b}$$

$$\min_{y_j \in Y_j} f_j(x_1, \ldots, x_L, y_1, \ldots, y_M), j = 1, \ldots, M \tag{7.10c}$$

$$\text{subject to } q_{jk}(x_1, \ldots, x_L, y_1, \ldots, y_M) \leqslant 0, \tag{7.10d}$$

$$k = 1, \ldots, j_k, j = 1, \ldots, M$$

7.3.2 A PSO-based Algorithm

In this section, we use the strategy adopted in the PSO method to develop a PSO-based algorithm to reach a Nash equilibrium solution for an MLMFB decision problem.

Figure 7.2 outlines the main structure of this algorithm. We first sample the leaders-controlled variables to get some candidate choices for leaders. Then, we use the PSO method together with the stretching technology (Parsopoulos & Vrahatis 2002) to get the followers' response for every leader's choice. Thus a pool of candidate solutions for both the leaders and the followers is formed. By pushing every solution pair moving towards current best ones, the whole solution pool is updated. Once a solution is reached for the leaders, we use the stretching technology (Parsopoulos & Vrahatis 2002) to escape local the optimisation. We repeat this procedure by a pre-defined count and reach a final solution.

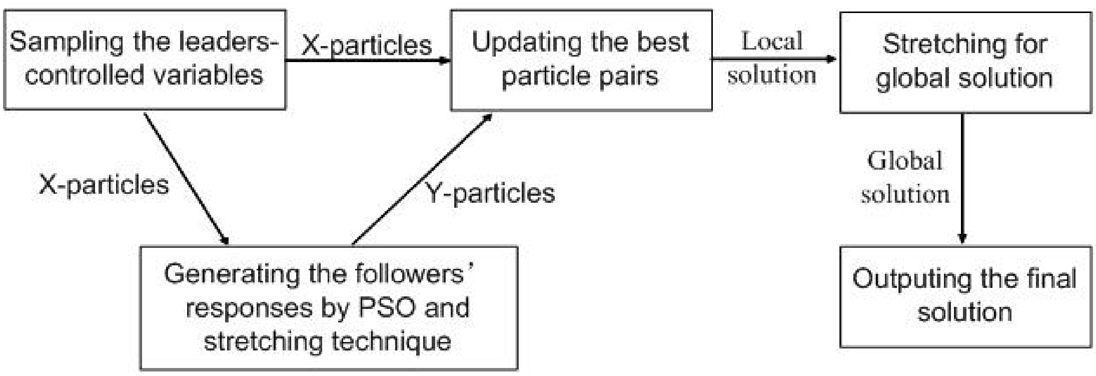

Figure 7.2: The outline of the PSO-based algorithm for MLMFB problems

The detailed algorithm is specified in Algorithm 11 and Algorithm 12. Notations used in the algorithms are detailed in Table 7.3.

Algorithm 11: Generate the response from the $m-$th follower

Step 1 Input the values of $x_1, \ldots, x_L$ from the L leaders;
Step 2 Sample N_f candidates y_{il} and the corresponding velocities $v_{y_{il}}$, $i = 1, \ldots, N_f$;
Step 3 Initiate the follower's loop counter $k_f = 0$;
Step 4 Record the best particles $p_{y_{il}}$ and y_l^* from y_{il}, $i = 1, \ldots, N_f$:
if y_{il} satisfy (7.5d) AND $f_l(x, y_{il}) < f_l(x, p_{y_{il}})$, then $p_{y_{il}} = y_{il}$.
For each $p_{y_{il}}$, if $f_l(x, p_{y_{il}} < f_l(x, y_l^*)$, then $y_l^* = p_{y_{il}}$.
Step 5 Update velocities and positions using
$v_{y_i}^{K+1} = w_f v_{y_i}^K + c_l r_{1f}^K (p_{y_i} - y_i^K) + c_l r_{2f}^K (y^{*K} - y_i^K)$
$y_i^{K+1} = y_i^K + v_{y_i}^{K+1}$
Step 6 $k_f = k_f + 1$;
Step 7 If $k_f \geqslant MaxK_f$ **OR** the solution changes for several consecutive generations are small enough, then we use Stretching technology to obtain the global solution and goto Step 8. Otherwise goto Step 5;
Step 8 Output y_l^* as the response form the $m-$th follower.

Algorithm 12: Generate optimal strategies for leaders

Step 1 Sample N_l particles of $(x_{11}, \ldots, x_{L1}), \ldots, (x_{1N_l}, \ldots, x_{LN_l})$, and the corresponding velocities $(v_{11}, \ldots, v_{L1}), \ldots, (v_{1N_l}, \ldots, v_{LN_l})$;
Step 2 Initiate the leaders' loop counter $k_l = 0$;
Step 3 For $k-$th particle, $k = 1, \ldots, N_l$, calculate the optimal response $r_i(x_{-i}), i = 1, \ldots, L$
Step 3.1 Sample N_l particles x_i within the constraints of x_i;
Step 3.2 For $m = 1$ to M, we calculate the rational response from the $m-$th follower by calling Algorithm 11 ;
Step 3.3 Using PSO technique, we obtain $r_i(x_{-i}), i = 1, \ldots, L$
Step 4 Calculate the function value of every particle by (7.4a);
Step 5 Record $p_{x_{ij}}$, x_{ij}^*, $j = 1, \ldots, N_l$:
For each $x_{ij}, j = 1, \ldots, N_l$, if x_{ij} satisfy (7.4c) $F(x_{ij}) < F(p_{x_{ij}})$, then $p_{x_{ij}} = x_{ij}$;
For each $p_{x_{ij}}$, if $F(p_{x_{ij}}) < F(x_i^*)$, then $x_i^* = p_{x_{ij}}$;
Step 6 Move particles by best positions:
$v_{x_{ij}}^{K+1} = w_l v_{x_{ij}}^K + c_l r_{1l}^K (p_{x_{ij}} - x_{ij}^K) + c_l r_{2l}^K (x_{ij}^{*K} - x_{ij}^K)$
$x_{ij}^{K+1} = x_{ij}^K + v_{x_{ij}}^{K+1}$
Step 7 $k_l = k_l + 1$;
Step 8 If $\sum_{i=1}^{L} |x_i - r_i(x_i)| \leqslant \varepsilon$ **OR** $k_l > MaxK_l$, we use Stretching technology to current leaders' solutions to obtain the global solution.

7.4 Summary

This chapter studies general bi-level problems where the objectives and constraints for leaders and followers may have arbitrary formats. We divide general bi-level deci-

Table 7.3: The explanation of some notations in Algorithms 11 and 12

N_l	the number of candidate solutions (particles) for the leaders
N_f	the number of candidate solutions (particles) for the followers
x_{ij}	the $j-$th candidate solutions for the controlling variables from $i-$th leader
$p_{x_{ij}}$	the best previously visited position of x_{ij}
x_i^*	current best one for particle $x_{ij}, j = 1, \ldots, N_l$
$v_{x_{ij}}$	the velocity of x_{ij}
k_l	current iteration number for the upper-level problem
y_{ij}	$i-$th candidate solution for the $j-$follower
p_{y_i}	the best previously visited position of y_i
$p_{y_{ij}}$	the best previously visited position of y_{ij}
y_i^*	current best one for particle $y_{ij}, i = 1, \ldots, N_f$
k_f	current iteration number for the lower-level problem
$MaxK_l$	the predefined max iteration number for k_l
$MaxK_f$	the predefined max iteration number for k_f

sion problems into three categories, one with multiple leaders, one with multiple followers, and one with both multiple leaders and multiple followers at the same time. After giving mathematical definitions of MLOFB, OLMFB and MLMFB decision problems by Nash-equilibrium-based concept, PSO-based algorithms are developed to solve them respectively. These algorithms are implemented in the bi-level decision support system in Chapter 8. Some of these algorithms will be used to solve real world bi-level decision problems in Chapter 9.

8 A Fuzzy Bi-level Decision Support System

This chapter develops a fuzzy bi-level decision support system (FBDSS), which incorporates the optimisation algorithms developed in this study for both linear and non-linear bi-level decision problems, to help decision makers in bi-level situation achieving well-informed solutions. Meanwhile, decision makers can adjust their subjective preference to obtain balance among different objectives during the solution process through interacting with the system. Furthermore, this FBDSS can solve bi-level decision problems with fuzzy coefficients. It is also helpful to experts who establish the fuzzy coefficients in both objective functions and constraints. Through experiments conducted on this FBDSS, they can compare and refine any coefficient in a bi-level decision model for more accurate definition and format.

8.1 System Configuration and Main Interfaces

The FBDSS is developed in the Microsoft 32-bit Windows environment. We adopt the object-oriented approach and implement it by Microsoft Visual Basic 6.0. With the Windows-based interface, it takes full advantages of the graphical capabilities of Windows environment that enables users to exploit the capabilities of the system.

In the main interface, as shown in Figure 8.1, the menu includes items of “File”, “Methods”, “Model”, “Result”, and “Help” that perform all kinds of decision support activities. Figure 8.2 shows the overall structure of the menu, where we can create a linear bi-level decision model or a general (linear or non-linear) bi-level decision model by clicking the item of “New a general bi-level model” or “New a linear bi-level model”; clear current model configuration by clicking “Reset current model” item; trigger the “approximation Branch-and-Bound-based algorithm”, “approximation Kth-Best-based algorithm”, “PSO-based algorithm” or “goal-programming-based algorithm” by the items of “Approximation B_B”, “Approximation kbest” “PSO” or “Approximation goal”; display the solution for current bi-level decision model (linear or non-linear) by clicking “Linear optimisation result” or “General bi-level optimisation result” respectively.

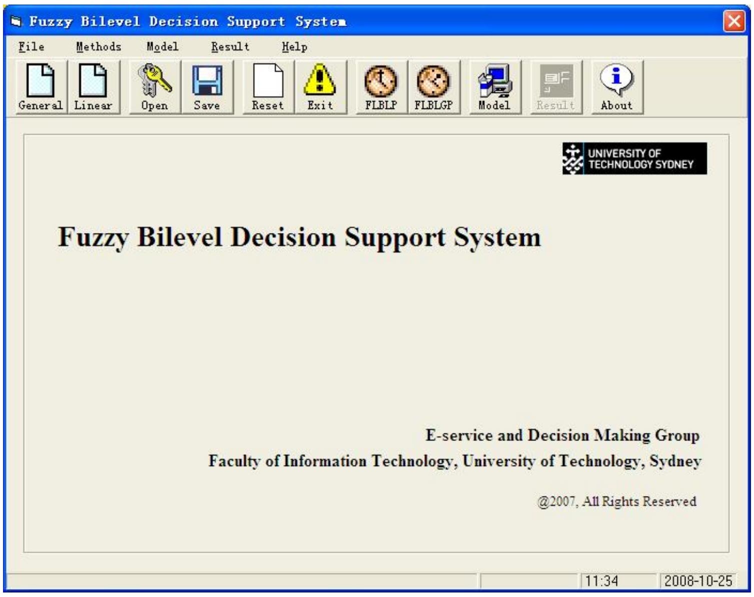

Figure 8.1: The main interface of the FBDSS

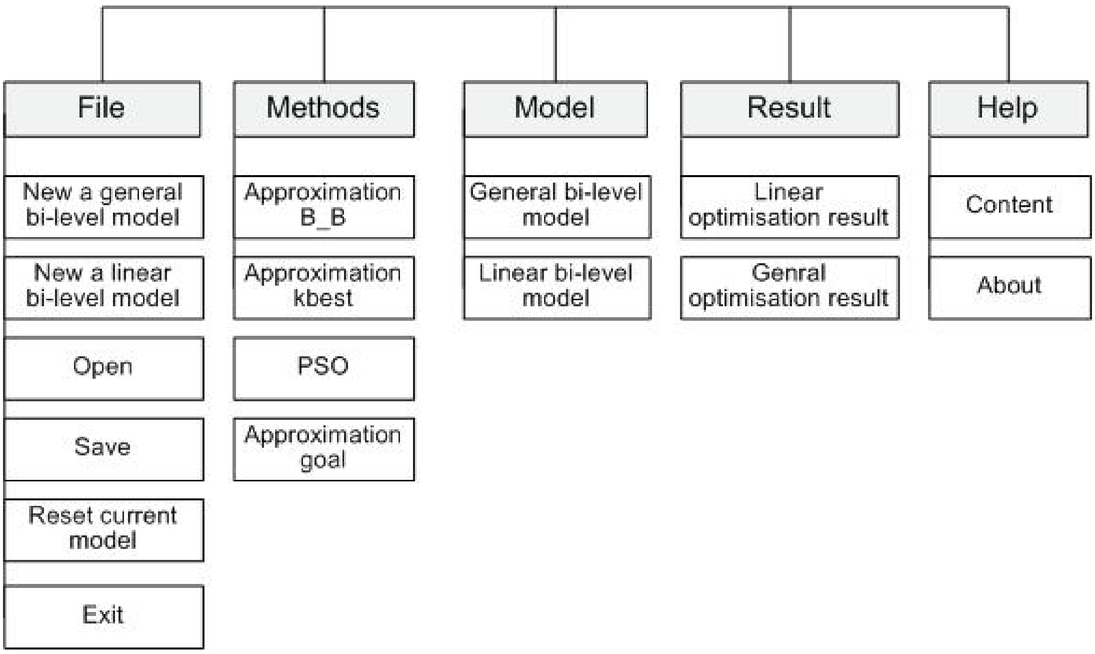

Figure 8.2: The menu structure of the FBDSS

8.2 System Structure

As a specific type of DSS, this FBDSS provides computerised assistance to the decision makers in a decentralised organisation to gather knowledge about a bi-level decision problem and control the decision making process for a better-informed decision.

The structure of the FBDSS is depicted in Figure 8.3. Within this architecture, five modules are involved, i.e. “user interface”, “model management”, “algorithm engine”, “updating system”, and “visualisation”. Data are collected through the user interface, and formatted as a corresponding bi-level decision model by “model management” module. The core calculations are carried in “algorithm engine” over this model, and solution is outputted through “visualisation” module to the end user by “user interface”.

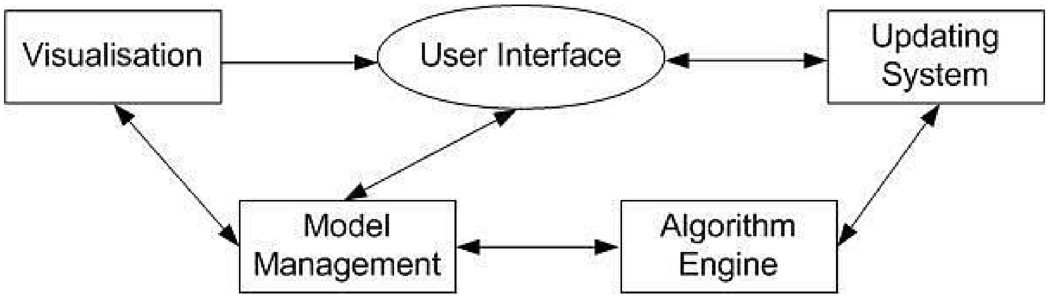

Figure 8.3: The system structure of the FBDSS

These modules are detailed below:

(1) User Interface:

Data can enter the system from two sources: system users and data sets. A user can key in the data directly through the interface, which will build a bi-level decision model. Meanwhile, the data may be stored in terms of existed projects and be retrieved for later calculation. These data are passed to the “model management” module where “algorithm engine” can be triggered to run for a solution.

Outputs from the system include feasible solutions and all objectives for both the leader and the follower. If a leader is not satisfied with current solution for a fuzzy bi-level decision problem, he/she can adjust the satisfactory degree and trigger the algorithm engineer for another solution.

(2) Model Management:

The “model management” module formulates a bi-level problem in terms of objectives and constraints based on the user’s input, and controls the model retrieval and use. The modelling procedure contains three major steps consisting of the generation of objectives, constraints and the elicitation of a certain satisfactory degree for a fuzzy bi-level decision problem. These functions are data driven, each requiring a set of coefficients.

(3) Algorithm Engine:

The "algorithm engine" performs the core calculation in the system including defuzzification, comparison among fuzzy sets, and bi-level optimisation. While a non-linear bi-level decision problem can only be solved by the PSO-based algorithms, the system provides three options i.e. the approximation Branch-and-Bound-based algorithm, the approximation Kth-Best-based algorithm, and the PSO-based algorithm for fuzzy linear bi-level decision problems. For multi-objective bi-level problems, an extra option of the goal-programming-based algorithm is provided in the system for decision makers to choose. Once the satisfactory degree is adjusted in the user interface module, the "algorithm engine" will be triggered to run on the selected bi-level decision model retrieved from the "model management" module.

(4) Updating System:

In this FBDSS, update routines are provided from two aspects, i.e. users can modify an established bi-level decision model retrieved from the model base; and users can adjust the satisfactory degree to obtain solutions under different aspiration levels for a fuzzy bi-level decision model.

To define a fuzzy bi-level decision model, a set of crisp numbers are employed to describe the fuzzy objective functions and constraints by fixing four points and the format of a fuzzy number. The capacity of incorporating new information into the established crisp numbers sometimes may introduce conflicts between constraints, or cause an existing feasible solution invalid. However the satisfactory-degree-adjustable mechanism can reduce the possibility of the non-solution situation.

(5) Visualisation:

To model a fuzzy bi-level decision problem, we need the information on objectives and constraints for both the leader and the follower. As fuzzy numbers are used to interpret coefficients, the way to describe fuzzy sets becomes crucial.

To solve a bi-level decision problem, we need to search equilibrium between the leader and the follower, both of who achieve the optimality while the leader takes the priority within the constraints. To describe this equilibria, the visualisation module will complete the presentation and interpretation functions. The output solutions will help the user identify preferred equilibrium, and give insights into how they can be achieved. As some bi-level decision problems addressed involve fuzzy numbers, their final objective values for both the leader and the follower are inevitably fuzzy numbers.

To implement the functions of intuitive fuzzy number input and pellucid fuzzy objective output, we describe a fuzzy number by the information of the formats of the left and right functions, and the two points where the membership value equal zero, and the other two points where the membership value equal one. Here, the formats of membership functions can be selected as linear, quadratic, cubic, exponential, and logarithmic. Figure 8.8 shows a window to identify a fuzzy number, and Figure 8.11 shows a window to display a fuzzy objective.

8.3 Decision Support Process

8.3.1 Linear Bi-level Decision Support

The whole decision supporting procedure for linear bi-level decision problems by this FBDSS involves three phases, i.e. problem identification, preference elicitation, and solution searching. The relationship among these phases is illustrated in Figure 8.4.

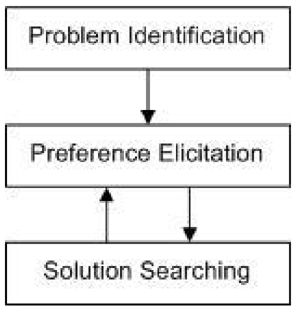

Figure 8.4: Optimisation process by the FBDSS

The first phase is to set up a framework for a problem in which the decision will be made. In this framework, we define what the decision makers wish to achieve as objectives, and any limitations and conditions as constraints. In specification, the following data need to be input to set up the framework.

(1) The number of followers and the number of decision variables, objectives and constraints for the leader, as shown in Figure 8.5;

(2) The variables, objectives and constraints for a leader, as shown in Figure 8.6;

(3) The variables, objectives and constraints for followers, as shown in Figure 8.6;

(4) The max/min choice for individual objective, as shown in Figure 8.7;

(5) The fuzzy coefficients occurred in the objectives and constraints, which should be entered one by one in Figure 8.8.

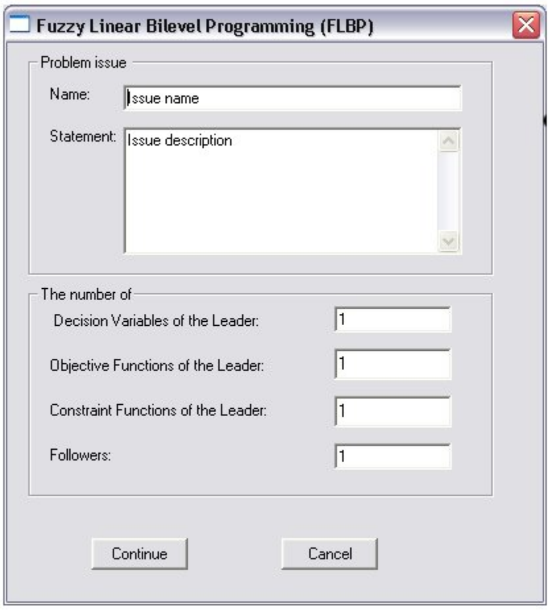

Figure 8.5: Interface for variable input-linear-1

The second phase is to elicit the preferences of decision makers. Once there exist multiple options, the decision makers at both levels are allowed to rank these options by assigning specific weights. (Refer to Figure 8.9 and Figure 8.10)

The final phase is to search a solution. Once a bi-level decision model is built up, the window, as shown in Figure 8.9, can be activated to reach a solution. In this window, the "approximation Branch-and-Bound-based algorithm", "approximation Kth-Best-based algorithm", or the "PSO-based algorithm" can be selected as the approach for the current linear bi-level problem by clicking the option button of "B_B", "K_Best" or "PSO". Provided the problem is a multi-objective bi-level decision problem and the decision maker chooses the goal-programming-based algorithm to solve it, the window, as shown in Figure 8.10, can be activated to reach a solution. In this window, the "approximation Branch-and-Bound-based algorithm", or the "approximation Kth-Best-based algorithm" can be selected as the default approach used by the goal-programming-based algorithm.

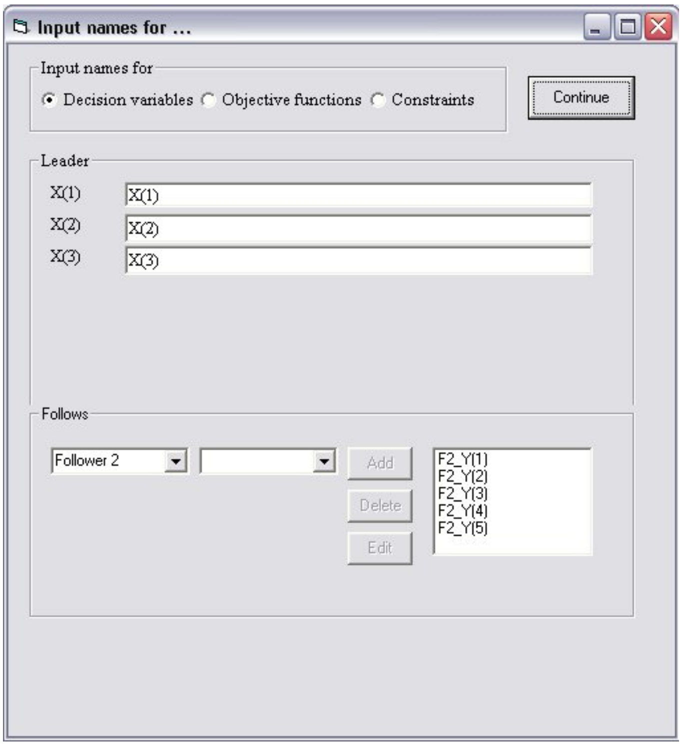

Figure 8.6: Interface for variable input-linear-2

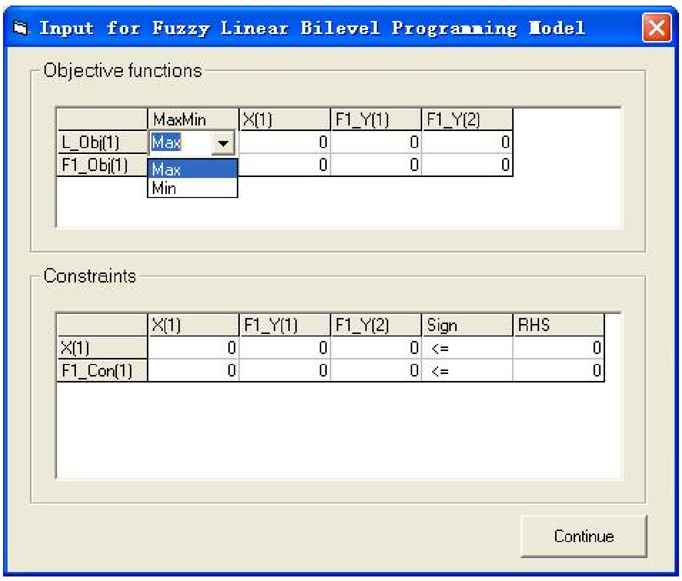

Figure 8.7: Interface for variable input-linear-3

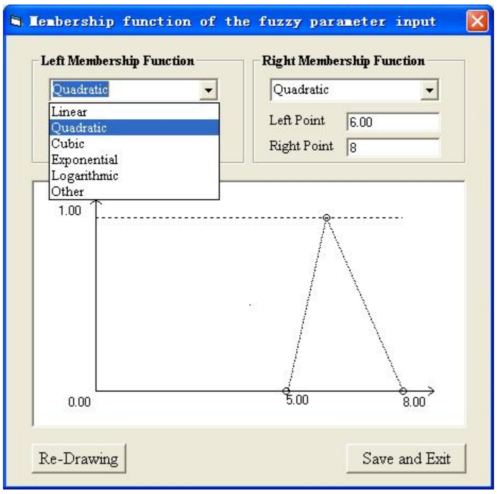

Figure 8.8: The input window for the membership function of a fuzzy number

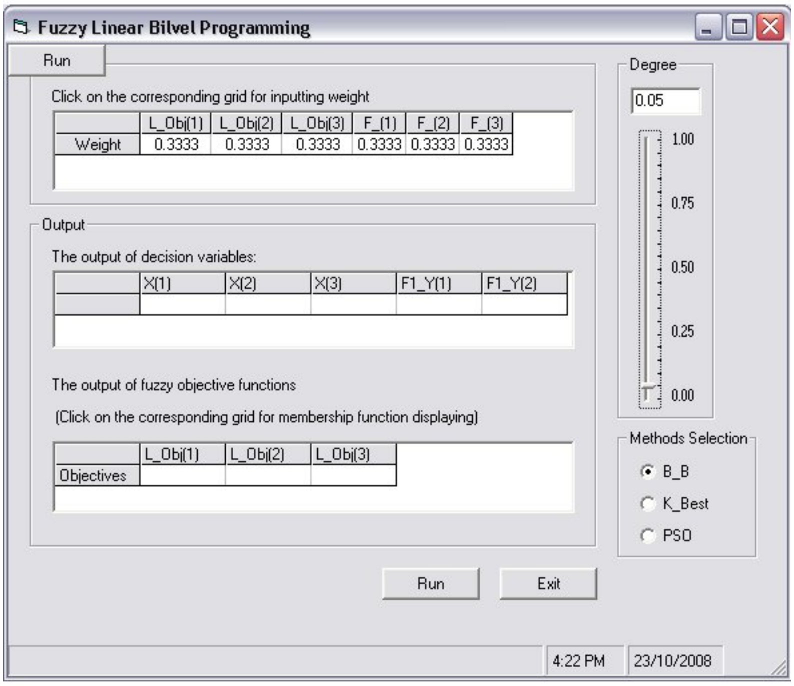

Figure 8.9: Interface for result displaying-linear

In the windows shown as Figure 8.9 and Figure 8.10, by adjusting the slider or keying in a value in the text box above the slider, a user can set a specific satisfactory degree for all membership functions of the fuzzy numbers. Thus, a solution under a specific satisfactory degree chosen by the user can be obtained.

By clicking the "Run" button in windows shown as Figure 8.9 and Figure 8.10,

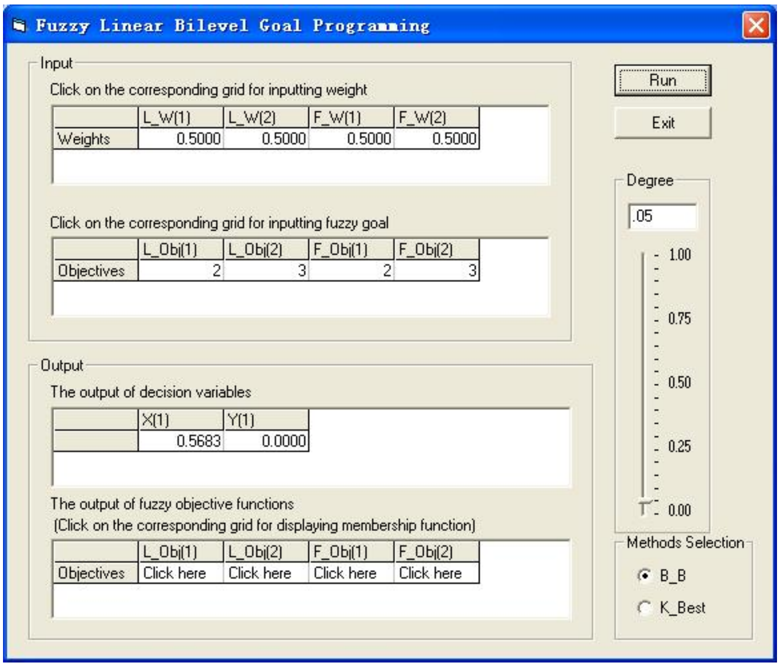

Figure 8.10: Interface for result displaying-goal

the optimisation algorithm selected will run and the solutions will be shown and the corresponding fuzzy objective can be displayed in the window shown as Figure 8.11 sequentially.

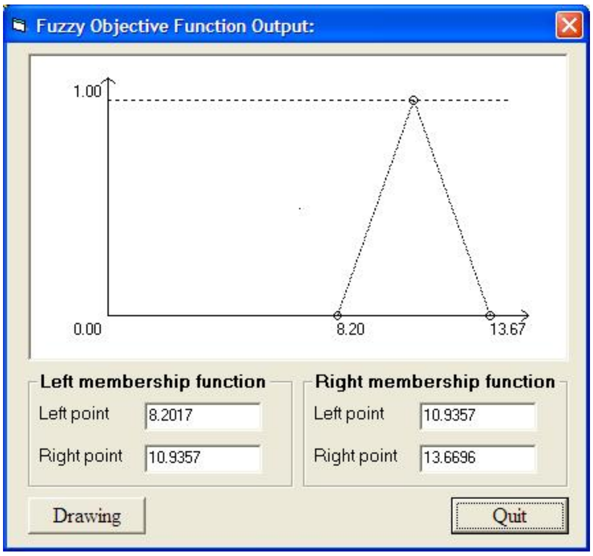

Figure 8.11: The window for displaying a fuzzy objective

If a leader is not satisfied with current solution, he/she can change the weights for each objective or adjust the satisfactory degree for another solution.

8.3.2 Non-linear Bi-level Decision Support

To solve a non-linear bi-level decision problem, we need to build a framework for the problem in which the decision will be made. In this framework, we define what the decision makers wish to achieve as objectives, and any limitations and conditions as constraints. As non-linear formulas have very flexible and unexpected forms, this system gives tremendous flexibilities to users by providing text boxes to input objective and constraint functions. In specification, the following data need to be input to set up a non-linear model.

(1) The number of leaders and followers, as shown in Figure 8.12;

(2) The variables, objectives and constraints for the leaders and followers, as shown in Figure 8.13;

(3) The fuzzy coefficients (if there are fuzzy coefficients involved in the problem) occurred in the objectives and constraints, which should be entered in Figure 8.14, and whose membership functions could be entered one by one in Figure 8.8;

(4) The max/min choice for individual objective, as shown in Figure 8.14;

(5) The "$\geqslant$", "$\leqslant$", or "=" choice for individual constraint, as shown in Figure 8.14;

(6) The function formats for objectives and constraints for both leaders and followers, as shown in Figure 8.14.

To facilitate formula inputting, a list box, as shown in Figure 8.14 can be activated to show variable names entered in Figure 8.13. Thus users can refer to the variable names to reduce the chance of mistyping.

Once a non-linear bi-level decision model is established, a window shown in Figure 8.15 will show up to display the solutions and objective values for both leaders and followers. If there are fuzzy coefficients involved in this model, the objective values will be fuzzy numbers. In such a situation, a window shown in Figure 8.16 will be shown to display the solutions and fuzzy objective values for both leaders and followers. Once a user clicks the text box of an objective value, a button labelled as "membership" will show up. The clicking of this "membership" button will activate a window shown in Figure 8.17 to display the corresponding fuzzy value for the objective.

Figure 8.12: Interface for variable input-nonlinear-1

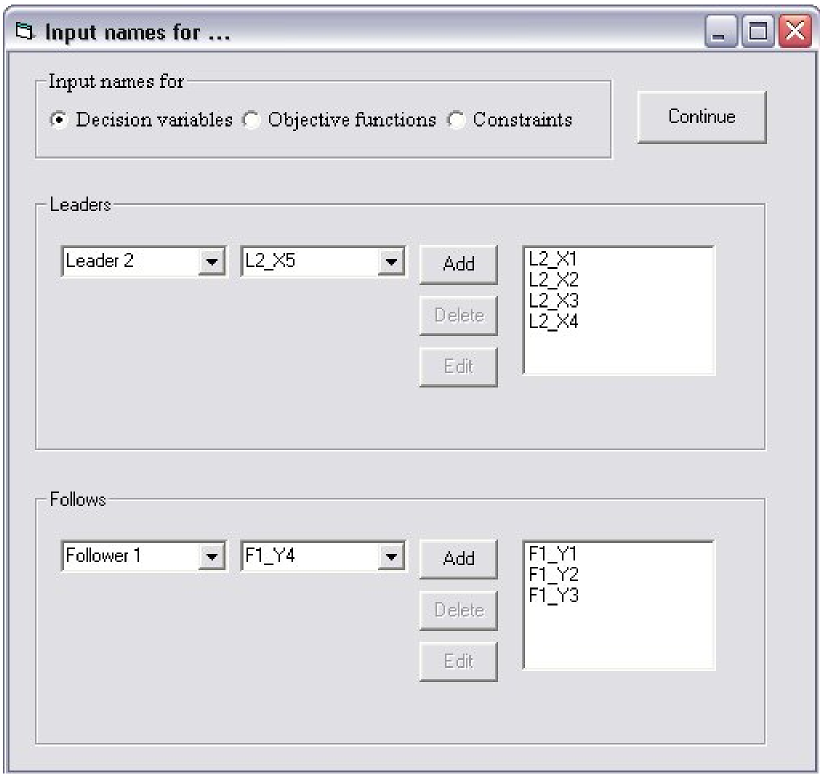

Figure 8.13: Interface for variable input-nonlinear-2

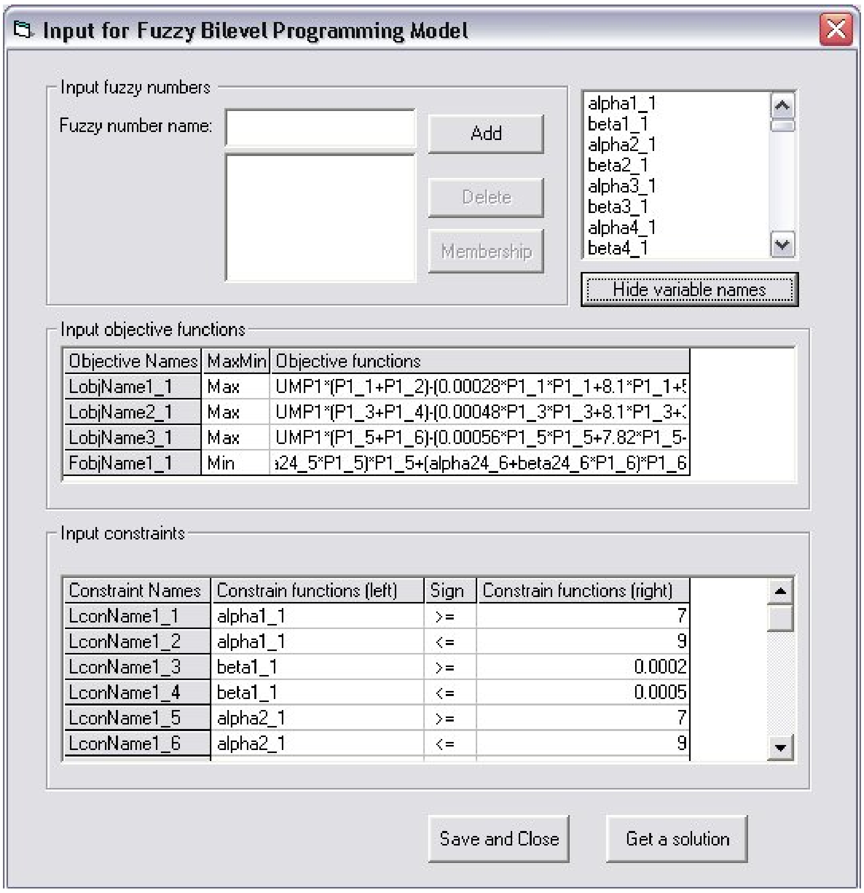

Figure 8.14: Interface for variable input-nonlinear-3

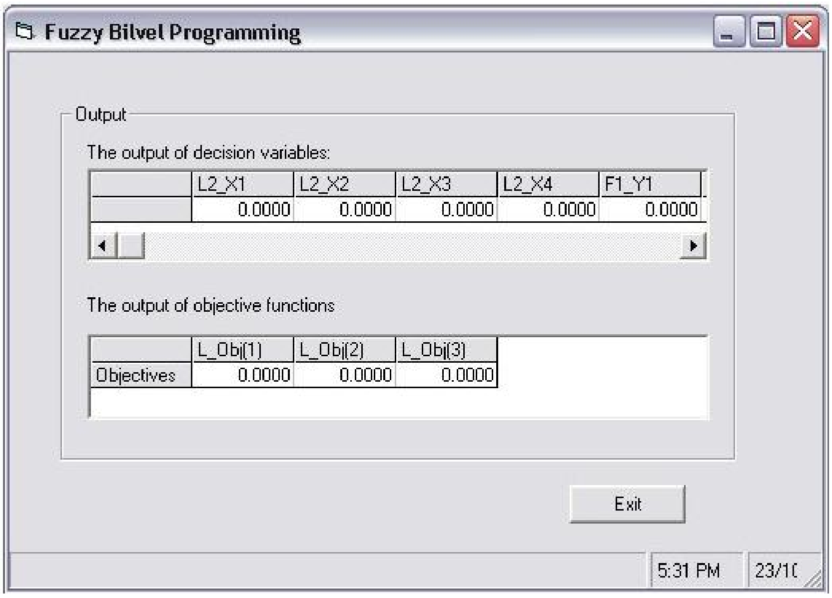

Figure 8.15: Interface for output-nonlinear

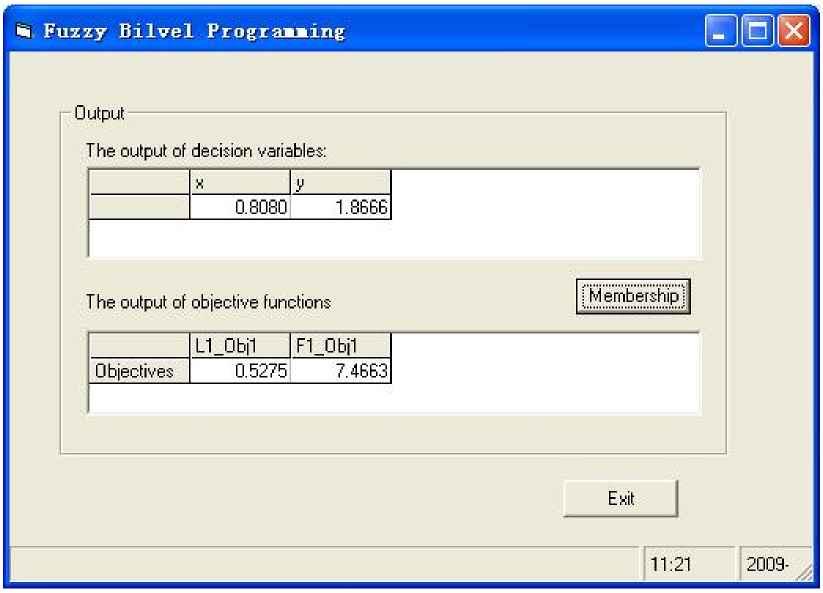

Figure 8.16: Interface for output-with-fuzzy-coefficients-nonlinear-1

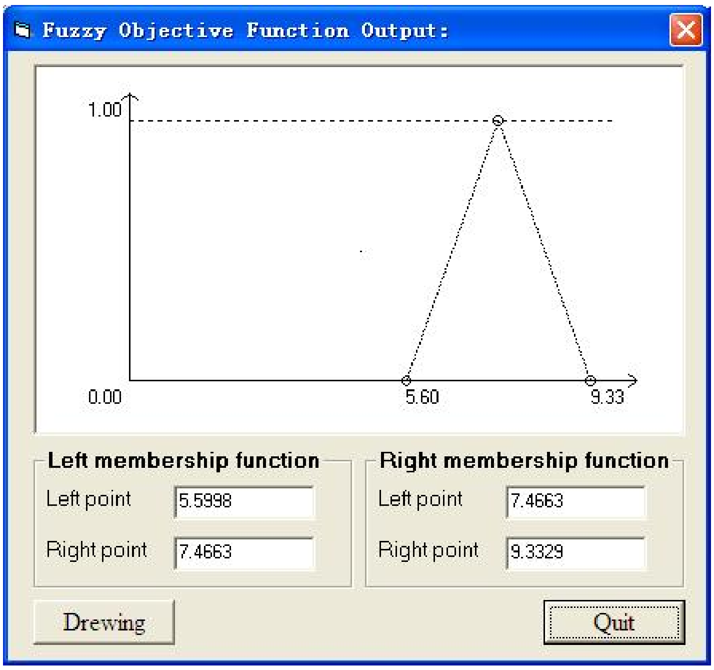

Figure 8.17: Interface for output-with-fuzzy-coefficients-nonlinear-2

8.4 Summary

This chapter presents an FBDSS developed to support bi-level decision making. This system has four features. The first is that the system can process and solve fuzzy bi-level problems with different satisfactory degrees. The second is that the system can deal with any form of membership functions of coefficients in a fuzzy bi-level decision model. The third is that the system involves several methods to suit much wider bi-level decision making circumstances under various uncertain environments. The last one is that this system can recognise and deal with a large amount of arbitrary function inputs, which gives much flexibilities for decision makers to describe their particular bi-level decision problems. When choosing a specific method to solve a bi-level decision problem, the approximation Branch-and-Bound-based algorithm and the approximation Kth-Best-based algorithm are recommended for a decision maker if a linear bi-level decision problem is involved as these two algorithms can reach precise solutions with faster running compared to the PSO-based algorithm, which is however the only option for a non-linear bi-level decision problem.

9 Bi-level Decision Applications

This chapter applies the techniques developed in this study on real world bi-level decision problems in the fields of railway train set organisation, railway wagon flow management, electricity markets, and supply chains. Bi-level decision models are established for these problems and the algorithms developed in this research are applied to reach solutions for these problems.

9.1 A Bi-level Decision Model for Railway Train Set Organising Optimisation

9.1.1 Background

Railway transportation, as one of the most important vehicles ways, has always been playing an irreplaceable role in social economics. For railway freight transportation, about 80% of the whole transportation time is allotted to the operations of loading/unloading, transferring, and overhauling in railway technical stations (Li & Du 2002). The working state of technical stations, therefore, will influence the whole overpass ability of the railway network. Thus the research on the railway transportation optimisation will be bound to focus on the operation of technical stations. The main methodologies used include scheduling theory (Li & Du 2002), graph theory (Li & Du 2002; Wang 2004), mathematical programming (Li & Du 2002), and operational theory (Li & Du 2002). Also, some scholars have addressed the problem of traffic controlling from multiple levels' angle (Feng & Wen 2005; Berezinski, Holubiec & Petriczek 1994; Shi, Fang, Li, Mo & Huang 2003).

Train set organisation (TSO) is to arrange the train set in railway freight transportation. Most current research by bi-level techniques on traffic controlling centers on the transformation network design and layout (Feng & Wen 2005). Little research has been conducted towards TSO problems from multiple levels' angle. In this section, we use the bi-level method to study the problem of TSO in the railway running and management.

9.1.2 Problem Analysis

Train set organisation, aiming at arranging the train set in railway freight transportation and with extraordinary professional and technical specialties, is one of the main subjects in railway transportation management. The objectives of TSO include: to make the transportation efficient and even; to use the transporting device reasonably; and to promote the cooperation among different departments involved in the freighting procedure. The term of "organising" here means arranging, deciding and managing, while "train set organising" acts to arrange the train set, make decisions on related issues, and manage the procedure in railway transportation.

There exist multiple levels among the running of TSO, i.e. the "national railway network level" the top, the "local bureau railway network level" the second, the "stations" the third, and the "operating group" the bottom. However, as the operating objects of both the national railway network and the local bureau railway network are train set, while those of the two lower levels are trains, the organisation of TSO can be generalised into two levels: the railway network the leader, and the stations the followers. Thus bi-level programming techniques can be used to analyse the problem.

The main concerns of the railway network are to decide the train type (pick-up-and-drop-train, district-train, transit-train, or through-train), the train constitution, the train number, and the detailed route of the departing train set. The objectives of the railway network include: improving the transportation capacity and service speed, reducing the cost, balancing the working rhythm among divisions, and assigning the break-up and make-up jobs among different stations rationally.

The tasks assigned to a station are to constitute normative train set required by the railway network from all kinds of freight wagons that stop by this station. Involved with these tasks, there also include a series of relevant operations, such as: collecting or delivering, shunting, loading/unloading, and wagon checking. The main concerns of stations include: making the operating efficient, economical and safe; rationally using the transportation devices such as track, shunting locomotive, and hump; deciding the operation steps together with its schedules; and cooperating among steps within the schedule-frame of the railway network.

Two levels though TSO can be divided into, the separate levels still share intrinsic consistency. For the upper level, when making a TSO plan, the railway network must consider the influence from the specific operating ability and device conditions of stations, while calculating the influence factors from itself such as the amount and destinations of trains and the track conditions; For stations located at the lower level, when implementing the working goals, they should try their best to harmonise between their own operation abilities and the working arrangement from their top counterpart.

Railway stations can be grouped into two classes: "through stations" and "technical stations". Compared with technical stations, through stations are small sized and their daily works, mainly on helping trains go through or two train set from opposite directions meet, are simple and the workload are small. Except for all the functions of through stations, technical stations are to make new train set by breaking up the old ones and adding transship trains and trains originated there. Tasks also include: arrival/departure operating, collection-and-delivering operating, shunting, loading/unloading, and wagon checking. We generalise these operations at technical stations as "shunting and transship operation".

For the reason of facilitating the modelling, we simplify the tasks of TSO by assuming that:

1) The railway transportation supply is less than the demand; the aim of TSO is to fully use the transportation ability to provide as much transportation as possible.

2) The topo structure of a railway network is a circle formed by train lines. This is to embody the continuous nature of the net and transportation circulation.

3) The main line is double-track with every track's direction fixed, which means there allowed two train set running in reverse directions between two stations simultaneously. This is to avoid the meeting problem of two train set with opposite running directions.

4) Within a railway net, there located only technical stations, and runs only one type of trains, the "district trains", which are from one technical station A and to the other technical station B. Between A and B there is no other technical stations.

5) The unit workload of "shunting and transship operation" for all technical stations are the same. In other words, every technical station shares the identical amount of operating time for the same train set.

Based on these assumptions above, the decision-maker on a railway network wishes that the density of train set (calculated by the time intervals between any two adjacent running train set) and the length of train set (the number of trains of any train set) as large as possible to obtain the maximal transport capacity. However, for the sake of safety, the density has its upper limit set by the railway network. And restricted by the motive power of locomotive and the useful length limit of arrival-departure track, the train set length has its upper limit as well.

Ignoring the constraints by a railway network, the stations, on one hand, wish the length of train set large because the larger the length, the more efficient the operating and the lower the unit operating cost. The operating efficiency is the amount of trains shunted and transshipped per unit time, while the unit operating cost is the cost for every single train. On the other hand, the operating time for shunting and transshipping, which influences the cost, will increase if the length of train set increases. However, the overall effect of the trains set length is that the general unit operating cost will decrease with the increase of the train set length.

From the analysis above, we can include that, for the variable of the length of train set, the two levels share the same objective: the larger the length of the train set the better. For the variable of the density of train set, the decision makers at the upper level pursue its minimum while those at the lower level wish it change with the train set length in the same direction. Generally speaking, the shunting and transshipping time in stations is larger than the safe time intervals of any two adjacent running train sets, so the variable of the density of train set is determined by the lower level, the stations, while the variable of the length of train set is controlled by the top level, the railway network.

9.1.3 Model Building

Based on the analysis above, a bi-level decision model of TSO is built as:
For $x = (x_1, x_2, ..., x_n) \in X \subset R^n$, $y \in Y \subset R^m$, $F, f : X \times Y \to F(R)$,

Leader: Decision-maker of the railway network

$$\max_{x \in X} F(x,y) = \frac{a_1 \cdot \sum_{i=1}^{n} w_i \cdot x_i}{\sum_{i=1}^{n} w_i \cdot y_i} \tag{9.1a}$$

$$\text{subject to } \sum_{i=1}^{n} w_i \cdot x_i < m \tag{9.1b}$$

$$\sum_{i=1}^{n} w_i \cdot y_i > c_1 \tag{9.1c}$$

the i-th follower: the i-th technical station

$$\min_{y_i} f_i(x_i, y_i) = -b_1 \cdot x_i - b_2 \cdot y_i \tag{9.1d}$$

$$\text{subject to } c_2 \leqslant \frac{x_i}{y_i} \leqslant c_3 \tag{9.1e}$$

$$y_i > c_4 \tag{9.1f}$$

Explanation:

1) Variables:

   x_i: the length of train set for the i-th station, which is the number of trains of any train set controlled by the leader, the decision-maker of the railway network.

   y_i: the density of train set for the i-th station, which is the time interval between any two adjacent running train set, controlled by the i-th follower, the i-th technical station.

2) Coefficients and constants:

   n: the number of technical stations in the railway network.

   w_i: the relative weight for the i-th station in the railway network.

   a_1: the time interval. If $a_1 = 24$, then $a_1/\sum_{i=1}^{n} w_i \cdot y_i$ means the number of train set going through the network within 24 hours. $a_1 \sum_{i=1}^{n} w_i \cdot x_i / \sum_{i=1}^{n} w_i \cdot y_i$ is the number of trains going through the network per day, and $a_1 > 0$.

   m: the maximum number of trains of any train set regulated by the "Safety Terms". When the trains are empty, the main concern is not to exceed the length limit. When the trains are loaded, the weight limit becomes the decisive factor. However, for the sake of safety, when computing, both the length and weight must meet the requirements. No matter whether it is the weight or length, the ultimate limit is put on the number of trains.

   c_1: the minimum time interval between any trains list regulated by the "Safety Terms".

   b_1 and b_2: the weights set for the influencing power by the length and density to the unit cost.

   c_2 and c_3: the lower and upper number limits of the trains for technical stations to shunt and transship per time unit.

c_4: the least time for the technical stations to complete the shunting and transshipping.

3) Formula:

(9.1a) means the leader aims at obtaining the maximum throughput capacity within certain period of time. $\sum_{i=1}^{n} w_i \cdot x_i / \sum_{i=1}^{n} w_i \cdot y_i$ means the number of trains shunted and transshipped per time unit.

(9.1b) means the length of train set has its upper limit imposed by the locomotive's motive power and the arrival-departure track's useful length. When the trains are loaded, except for the length limit, there is still weight restriction set upon the train set, which means, the weights of goods loaded together with the weights of trains cannot exceed its upper limit.

(9.1c) means any two adjacent running train set cannot be too close for the sake of safety. c_1 is the minimum time interval between any trains list according to the "Safety Regulation".

(9.1d) means the followers wish that the cost is as low as possible. The first part of (9.1d) means the more the length of trains set, the more efficient of the shunting, and the lower the unit cost. The second part means the longer the train set remains in the station, the higher the cost.

(9.1e) means technical stations have their own lower and upper time limit to shunt and transship trains.

(9.1f) means there exists a least period of time for the technical station to complete the operation.

9.1.4 Experiments

In this section, we take the railway freight operation in a railway station "Station A" into consideration. Station A is a technical railway station with the duty of managing both the passenger transportation and freight transportation within the precinct of its Railway Bureau. The data collected from Station A cover the duration between November 1, 2006 and December 31, 2006.

Suppose the trains shunted and transshipped are to the direction of Station B, which is another station located next to Station A along its downlink. And the weight distribution of trains is listed in Table 9.1, with the locomotive is SS1(137 ton, 1.9 unit length).

Table 9.1: Train set distribution

WT	WS (ton)	Load (ton)	%	EL
B23	38	40	3	2.1
P64A	26	58	3	1.5
G70	23	58	9	1.1
G60	23	50	50	1.1
G70	23	55	35	1.1

The terms in Table 9.1 are explained as below:

WT: wagon type, the type of wagon used.

WS: wagon suttle, the weight of the empty wagon.

Load: the weight of the goods loaded.

EL: equivalent length, the equivalent length of a wagon is calculated from the front clasp to the rear clasp, with the unit length as 11 meters. If the equivalent length is "1.1", then its actual length is $11 \times 1.1 = 12.1$ meters.

According to the model defined by (9.1), the coefficients are calculated and discussed below:

- a_1: as the computation is within the "Basal Daily Working Plan", which is to arrange wagon assignment and schedule necessary operations based on the "Trains Running Chart", 'Trains Shunting Plan", "Detailed Rules on Technical Station Management", and constraints set by operating spots, the computing of the freighting wagon organisation is limited within a working day of 24 hours. So a_1 is set to 24.

- m: limited by the pulling ability of the locomotive and the territorial landform, such as grading, within Station A's precinct, the weight of train set must not be larger than 3500 tons; The departure track used for trains set to the direction to Station B is Track IV, Filed II, whose effective length is 890 meters. Deduced by 30 meters of braking distance, which is left for trains to stop safely, the maximum length for the trains set is 860 meters.

  Taking the constitution of the trains listed in Table 9.1, we set 1 "unit train" as a virtue train whose equivalent length, denoted by l_1 (meters), and weight, denoted by w_1 (ton), are calculated below:

$$\begin{aligned} l_1 &= 2.1 \times 0.03 + 1.5 \times 0.03 + 1.1 \times 0.09 \\ &\quad + 1.1 \times 0.5 + 1.1 \times 0.35 \\ &= 1.142 \\ w_1 &= (38 + 40) \times 0.03 + (26 + 58) \times 0.03 \\ &\quad + (23 + 58) \times 0.09 + (23 + 50) \times 0.5 \\ &\quad + (23 + 55) \times 0.35 \\ &= 66.95 \end{aligned}$$

The maximum number of such empty "unit train", denoted by m_e, is $(860 - 1.9 \times 11)/(1.142 \times 11) = 66$, and the maximum number of such loaded "unit train", denoted by m_l, is $(3500 - 137)/66.95 = 50$.

From above analysing and computing, we obtain

$$m = \min\{m_e, m_l\} = \min\{66, 50\} = 50.$$

- c_1: for the sake of safety, the pursuing distance, the minimum distance interval between any adjacent running trains list, is 10 kilometers, which costs about 0.2 hours in the journey from Station A to Station B. So c_1 is set to 0.2.

- b_1 and b_2: we set the weights of length and density of trains set on the cost of the station as 0.4 and 0.6 respectively.

- c_2 and c_3: the least number of trains Station A can shunt and transship is 30 per hour, while the max number is 150.

- c_4: the least time for Station A to complete the shunting and transshipping for a train set is 0.68 hour.

Thus, the bi-level problem defined by (9.1) is specilised as (9.2) in Station A:

Leader: Decision-maker of the railway network

Objective: $\max_x F(x, y) = \dfrac{24x}{y}$ (9.2a)

subject to $x < 50$ (9.2b)

$y > 0.2$ (9.2c)

Follower: Station A

$$\textbf{Objective: } \min_{y} f(x,y) = -0.4x - 0.6y \tag{9.2d}$$

$$\text{subject to } 30 \leqslant \frac{x}{y} \leqslant 150 \tag{9.2e}$$

$$y > 0.68 \tag{9.2f}$$

As illustrated in Fig. 9.1, the triangle “ABC” depicts the constraint region for this example. The dotted lines with arrows of “F” and “f” represent the optimising directions for the leader and follower respectively. The optimal solution to this example occurs at the point $(x^*, y^*) = (50, 1.67)$ with $F^* = 718.6$ and $f^* = -21.002$, which means, the railway network will obtain its maximum throughput capacity of 718.6 trains per day, if the decision makers of the railway network set the average number of train to 50, followed by Station A setting the time interval between every two adjacent train sets to 1.67 hour.

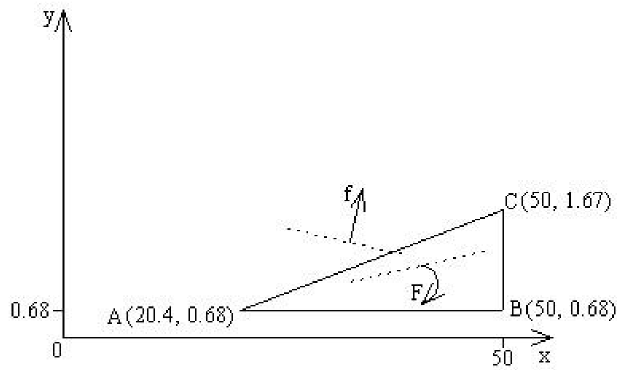

Figure 9.1: Geometry of the bi-level programming

9.1.5 Conclusions

In this section, the bi-level nature in train set organisation has been first put forward by abstracting and simplifying railway trains management. First, the bi-level decision model is developed. Then, this model is applied to Station A for a real case study. The testing result obtained from Station A is reasonable and could be helpful to its train organisation. However, as a lot of practical details have to be ignored for articulating the model building, this model has its limitations when applied directly for TSO decision making. Future efforts will be focused on relating more practical and randomly occurred issues from field work.

9.2 A Bi-level Optimisation Strategy on Railway Wagon Flow Management

9.2.1 Background

Railway freight transportation, as one of the most important transportation methods, has always been playing an irreplaceable role in social economics. The optimisation on wagon flow organisation and management can impose tremendous influence for both railway bureaus and technical stations.

Railway wagon flow management (RWFM) is to arrange wagon flows in railway freight transportation. One of the key issues faced by RWFM is how to arrange wagons generated or transferred in technical stations to form new wagon flows, while aiming at making transportation cost minimum and under constraints from both technical stations and rail tracks. An optimised solution to this problem can not only ensure freights to be sent to the destinations economically, but also make full use of all transportation facilities, thus reduce jamming probabilities and improve the transportation ability as a whole.

Due to the difficulties arising from both wagon routing and marshalling plan optimisation, it is even more difficult to integrate these two issues. The most popular way is to choose wagon routs first, then optimise the marshalling plans in every station. Although this strategy can decrease the problem solving difficulties, it still can not reach global optimisation solutions as the benefits from the most optimised routing can be offset by some extra workload brought in stations (Lin & Zhu 1996). Other methodologies used include scheduling theory (Ginzberg & Stohr 1982), graph theory (Ginzberg & Stohr 1982; Lu *et al.* 2007e), mathematical programming (Ginzberg & Stohr 1982; Zhang & Lu 2007b), 0-1 programming (Lin & Zhu 1996), and operational theory (Ginzberg & Stohr 1982). Also, some scholars have addressed the problem of traffic controlling from multiple levels' angle (Feng & Wen 2005; Zhang & Lu 2007a; Yu, Dang & Wang 2006; Gao, Zhang, Lu & Gao 2007).

Most current research by bi-level techniques on traffic controlling centers on the transformation network design and layout (Feng & Wen 2005; Lu *et al.* 2007b). Little research has been conducted towards wagon flow management problems from multiple levels' angle. In this research, we use a bi-level method to study the problem of RWFM.

9.2.2 Problem Analysis

Before establishing the bi-level decision model for RWFM, we list some terms used in following content.

(1) Local wagon flow: Wagons that are loaded/uploaded or repaired in one technical station are called local wagon flow for this station.

(2) Local district wagon flow: Some wagons are loaded/uploaded or repaired in intermediate stations between two technical stations. This kind of wagon flow is called local district wagon flow for the two technical stations.

(3) Long-distance wagon flow: For a technical station, if a wagon flow is not its local wagon flow or local district wagon flow but belongs to another technical station (local wagon flow or local district wagon flow), this wagon flow is called long-distance wagon flow for this technical station.

(4) Service operation: To assist on the marshalling operation within one station, some auxiliary operations must be made, including: taking-out and placing-in of cars, picking-up and dropping trains, loading/uploading goods, and repairing. We call this kind of auxiliary operation as service operation.

RWFM, characterised by monopolisation, is usually run by three levels, i.e. railway ministry level, railway bureau level, and station level. However, when carrying out tasks assigned by its corresponding superior, a lower level can arrange its own resources to achieve as much profit as possible. The communication among levels is through marshalling plans which are designed by upper level but implemented by the lower counterparts. Marshalling plans are regulations on organising vans which may be destined to different destinations to form van lists. Optimisation on marshalling plans aims at minimising the time spent for centralising and detention in technical stations.

In this research, we take railway bureaus as leaders, and stations as followers. A railway bureau controls the workload and working rhythm of the stations in its administration area. A station, while controlling its own producing resources, decides which specific method it will use to achieve tasks to be carried in this station. Thus, the cost in a station is determined by both the station and its upper administrator, the railway bureau. However, the most optimal cost level for a station does not necessarily produce the most ideal cost status for the railway bureau who seeks an equilibrium with traffic and cost. Although there locates the railway ministry above railway bureaus and technical stations, this study, while not focusing on the reciprocal decision relation between a railway ministry and its bureaus, only takes the decision from the railway ministry as input constraints.

Once a railway bureau selects a marshaling plan, it means two kinds of data are determined. One is the technical station sequences where some marshaling operation will be carried for every long-distance wagon flow. The other is the number of vans to be marshaled in every station. For the leader (the decision maker in a railway bureau), his or her decision involves whether accepting a carriage and the way to deliver it. The marshaling plans made by a railway bureau involve only long-distance wagons. In some technical stations, some long-distance wagons should be merged or separated to decrease cost in stations and increase traffic efficiencies.

Technical stations perform marshalling operations as well as relevant following services, such as collecting, delivering, shunting, loading/unloading, and wagon checking. The facilities of these services depend on the quality of marshalling operation which is performed beforehand. Having local wagons, local district wagons and long-distance wagons as three kinds of marshalling objectives, marshalling operations with local wagons and local district wagons are flexible in technical stations. Stations can determine the extent and depth of the marshalling operation for local wagons and local district wagons. The better performed of marshalling operation, the easier the following services and thus the lower the cost. With the objective of making the costs as low as possible, technical stations reasonably marshal local wagons and local district wagons as thorough as possible. However, profound marshalling operation will inevitably raise the cost and the time allocated for marshalling in a technical station within some limitations. Thus a technical station will seek a best point where its marshalling operations can bring itself the lowest cost. The decision on how to marshal local wagons and local district wagons becomes key content for technical stations.

Among 1440 minutes a day, some time is allocated for other operations than marshaling. Also, some marshaling operations are fixed so that a technical station can not adjust it. Thus, a station can only decide on flexible wagon flows within available working time. A station needs first to distribute working time between local wagons and local district wagons, then divide it among different sections of a local district wagon flow. Based on it, a station will decide the amount of marshaling a day, the amount of wagons and time for every marshaling.

Generally speaking, technical stations make decisions from the following aspects.

(1) Marshaling percentage

Influenced by time limitation, some marshaling operations can be executed to only some wagons while others must be treated as if they had the same sequence number (the same destination station) to reduce marshaling load. Thus, the percentage of wagons which will be marshaled is a decision made by a technical station.

(2) Shunting choice

Within limited working time, a technical station can decrease the shunting precision to finish a marshaling operation on time. Different shunting precisions occur in both sort-shunting and group-shunting. For sort-shunting, every wagon should be placed sequentially by their destinations. For group-shunting, marshaling is supposed to be finished as long as wagons with the same destination are placed together. Group-shunting takes less working time than sort-shunting.

(3) Marshaling precision

Marshaling can be divided into different precise degrees. Actually, the destination of a wagon can be defined from generality to nicety by stations, operation areas, operation lines, or operation spots. The more precise, the more working time will be needed.

To facilitate modelling the RWFM problem, we have the following assumptions:

(1) Marshalling difficulty is decided by the disorder degree of the wagon flow to be marshaled. Disorder degree depends on the destination stations of every wagon and the relationship among them, which occurs randomly. In this research, we hold that the disorder degrees for wagon flows have no difference.

(2) Marshalling costs from two train flows, one of which is from Station A to Station B and the other is from Station B to Station A, may have trivial difference on marshalling cost. When making plans and calculating the cost, decision makers sometimes need to consider the influence from these differences. However, compared with other influencing factors, the influence from different directions is trivial and can be ignored. In this research, we hold that the marshalling costs with two train flows with different directions are exactly the same.

9.2.3 An OLMFB Decision Model on RWFM problems

From the analysis in Section 9.2.2, we can see that a RWFM is a kind of OLMFB problem. Based on the OLMFB decision model developed in Section 7.2.1 and the analysis on RWFM problems in Section 9.2.2, an OLMFB decision model for RWFM is built in this section:

For $x = (x_1, x_{21}, x_{22}, \ldots, x_{2m}) \in X \subset R^{m+1}$, $y_i = (\eta_{li}, \eta_{di}, y_{1Gil}, y_{1Sil}, y_{1Gid}, y_{1Sid}, y_{2Gil}, y_{2Sil}, y_{2Gid}, y_{2Sid}, y_{2ik}) \in Y_i \subset R^{11}$, F, $f : X \times Y_i \to F(R)$, $i = 1, 2, \ldots, n$

$$\max_{x \in X} F(x,y) = \sum_{p_j, l_j \in D} (p_j \times l_j)[J_w(1-r_j) - \bar{C}_{w_j}(x)] + \sum_{s_i \in S}[q_{d_i}(x_1) \times \xi_i \times J_s - C_{s_i}(x, y_i)] \tag{9.3a}$$

subject to

$$p_{ju\_min} \leqslant p_{ju} \leqslant p_{ju\_max}, j = 1, 2, \ldots, m \tag{9.3b}$$

$$p_{jd\_min} \leqslant p_{jd} \leqslant p_{jd\_max}, j = 1, 2, \ldots, m \tag{9.3c}$$

$$m_{j\_min} \leqslant x_{2j} \leqslant m_{j\_max}, j = 1, 2, \ldots, m \tag{9.3d}$$

$$m_{j\_min} = \max\{I_{ju} \times t_{T_j d}, I_{jd} \times t_{T_j d}\}, j = 1, 2, \ldots, m \tag{9.3e}$$

$$0 \leqslant q_{di} \leqslant v_i \tag{9.3f}$$

$$0 \leqslant q_{2i} + q_{di} \leqslant u_i \tag{9.3g}$$

$$\min_{y_i \in Y_i} f_i(x, y_i) = \sum_{s_i \in S}[(C'_{z1i} + c''_{z1i} + c'''_{z1i} + C'_{z2i} + C''_{z2i} + C_{z3i})(1 + A_i \sigma_i^{B_i}) + \triangle C_{2i}] \tag{9.4a}$$

subject to

$$q_i = q'_{d_i} + \sum_{d_k \in D_i} q''_{dik} + q_{zi} \tag{9.4b}$$

$$T_{ilmin} \leqslant \eta_{li} \times q_{di} \times (y_{1Gil} \times y_{2Gil} \times T_{Gi} + y_{1Sil} \times y_{2Sil} \times T_{Si}) + q_{di} \times S_{il} \times T_{si} \leqslant T_{ilmax} \tag{9.4c}$$

$$T_{idmin} \leqslant \eta_{di} \times q_{di} \times (y_{1Gild} \times y_{2Gid} \times T_{Gi} + y_{1Sid} \times y_{2Sid} \times T_{Si}) + q_{di} \times S_{id} \times T_{si} \leqslant T_{idmax} \tag{9.4d}$$

$$i = 1, 2, \ldots, m \tag{9.4e}$$

where

$$\bar{C}_{w_j}(x) = \bar{C}_{w_0} + \frac{\bar{C}_{w_{1j}}}{p_j} + \frac{\bar{C}_{w_{2j}}}{x_{2j}} + \triangle \bar{C}_{w_j}$$

$$\triangle \bar{C}_{w_j} = |p_{ju} - p_{jd}| \times \frac{\bar{C}'_{w2j}}{x_{2j} \times p_j}$$

$$I_{ju} = \frac{p_{ju}}{1440 \times \varphi_{ju}}$$

$$I_{jd} = \frac{p_{jd}}{1440 \times \varphi_{jd}}$$

$$C'_{z1i} = \bar{C}'_{z1i} \times q'_{d_i}$$

$$\bar{C}'_{z1i} = Z'_{10i} + \frac{Z_{11i}}{a'_{1i}\bar{q}_{d_i}^{b'_{1i}}} + Z_{12i}((y_{1Gil} + 2 \times y_{1Sil} + y_{1i})^{b'_{2i}} + (y_{2Gil} + y_{2Sil})^{b'_{3i}} + \eta_{li}{}^{b'_{4i}})$$

$$C''_{z1i} = \bar{C}''_{z1i} \times q''_{d_i}$$

$$\bar{C}''_{z1i} = Z''_{10i} + \frac{Z_{11i}}{\sum\limits_{d_k \in D_i} a''_{1ik} x_{2ik}^{b''_{1ik}}} + Z_{12i} \times \sum_{d_k \in D_i} [a_{2ik}((y_{1Gil} + 2 \times y_{1Sil} + y_{1i})^{b''_{2ik}} + y_{2ik}{}^{b''_{3ik}} + \eta_{dik}{}^{b''_{4ik}})]$$

$$C'''_{z1i} = \bar{C}'''_{z1i} \times q_{zi}$$

$$\bar{C}'''_{z1i} = Z'''_{10i} + \frac{Z_{11i}}{\sum\limits_{d_k \in D_i} a'''_{1ik} x_{2ik}^{b'''_{1ik}}}$$

$$C'_{z2i} = \bar{C}'_{z2i} \times q'_{d_i}$$

$$\bar{C}'_{z2i} = Z'_{20i} + \frac{Z'_{21i}}{\alpha'_{21i}((y_{1Gil} + y_{1Sil} + y_{1i})^{\beta'_{21i}} + (y_{2Gil} + y_{2Sil})^{\beta'_{22i}} + \eta_{li}{}^{\beta'_{23i}})}$$

$$C''_{z2i} = \bar{C}''_{z2i} \times q''_{d_i}$$

$$\bar{C}''_{z2i} = Z''_{20i} + \frac{Z''_{21i}}{\sum\limits_{d_k \in D_i} \alpha''_{21i}((y_{1Gil} + 2 \times y_{1Sil} + y_{1i})^{\beta''_{21i}} + y_{2ik}{}^{\beta''_{22i}} + \eta_{dik}{}^{\beta''_{23i}})}$$

$$C_{z3i} = N_i \times \bar{C}_{cx}$$

$$N_i = \sum_{d_k \in D_i} [(C_{di} + C_{cHi} \times \varsigma_{ik}) \times x_{2ik} + C''_{fik} q''_{dik}] + C'_{fi} \times q'_{di}$$

$$C'_{fi} = \frac{Z'_{3i}}{[\alpha'_{3i}((y_{1Gil} + y_{1Sil} + y_{1i})^{\beta'_{31i}} + (y_{2Gil} + y_{2Sil})^{\beta'_{32i}} + \eta_{li}{}^{\beta'_{33i}})]}$$

$$C''_{fik} = \frac{Z''_{3i}}{[\alpha''_{3ik}((y_{1Gil} + 2 \times y_{1Sil} + y_{1i})^{\beta''_{31ik}} + y_{2ik}{}^{\beta''_{32ik}} + \eta_{dik}{}^{\beta''_{33ik}})]}$$

$$\triangle C_{2i} = (\sum_{d_k, d_l \in D_i, k \neq i} |x_{2ik} - x_{2il}| \times n_{kl}) \times Z_{4i}$$

$$Z_{4i} = Z_{40i} + \frac{Z_{41i}}{\sum\limits_{d_k, d_l \in D_i, k \neq i} \alpha_{4ik} |x_{2ik} - x_{2il}|}$$

Explanation:

1) Controlling variables:

x_1: Assignment of wagons which will go through the area administrated by a railway bureau and have more than one shunting operation in some technical station in this area.

x_{2j}: The number of vans within a shunted wagon list from the $j-$th section, which is from one technical station to another in a railway bureau.

x_{2ik}: The number of wagons in a wagon list which is to the $k-$th section in the $i-$th station.

η_{li}: Percentage of wagons to be marshaled for local wagon list in the $i-$th station.

η_{di}: Percentage of wagons to be marshaled for local district wagon list in the $i-$th station.

η_{dik}: Percentage of wagons to be marshaled for local district wagon list to the $k-$th direction in the $i-$th station.

y_{1Gil}: The percentage of wagons to be marshaled by group-shunting of local wagons in the $i-$th station.

y_{1Sil}: The percentage of wagons to be marshaled by sort-shunting of local wagons in the $i-$th station.

y_{1Gid}: The percentage of wagons to be marshaled by group-shunting of local district wagons in the $i-$th station.

y_{1Sid}: The percentage of wagons to be marshaled by sort-shunting of local district wagons in the $i-$th station.

y_{2Gil}: The shunting precision for local wagons to be marshaled by group-shunting in the $i-$th station.

y_{2Sil}: The shunting precision for local wagons to be marshaled by sort-shunting in the $i-$th station.

y_{2Gid}: The shunting precision for local district wagons to be marshaled by group-shunting in the $i-$th station.

y_{2Sid}: The shunting precision for local district wagons to be marshaled by sort-shunting in the $i-$th station.

y_{2ik}: The shunting precision for local district wagon flow marshaled in the $i-$th station to the $k-$th direction.

2) Other variables: While decision makers from both the upper and lower levels directly control variables of x and y, there are some other variables whose values are influenced by x and y directly or indirectly. These variables are summarised below:

Variables influenced by x_1

p_j: The average wagon flow in the $j-$th section. $p_j = p_{ju} + p_{jd}$.

p_{ju}: The average wagon flow in the $j-$th section in up-direction, which fluctuates with the change of x_1.

p_{jd}: The average wagon flow in the $j-$th section in down-direction, which fluctuates with the change of x_1.

q_{di}: The number of local district wagons and local wagons operated per day in the $i-$th station.

q'_{d_i}: The number of local wagons operated in the $i-$th station.

q''_{dik}: The number of local district wagons operated to the $k-$th direction in the $i-$th station.

q''_{d_i}: The number of local district wagons operated in the $i-$th station.

q_{2i}: The number of long-distance wagons operated in the $i-$th station.

ξ_i: The loading percentage in the $i-$th station.

r_j: The percentage of empty to loaded wagon kilometres in the $j-$th section. r_j fluctuates with the change of x_1.

Variables influenced by x

q_{zi}: The number of wagons marshaled in the $i-$th station a day.

ς_{ik}: The number of long distance wagons to the $k-$th direction in the $i-$th station.

n_{kl}: the number of wagon lists some of whose wagons have been added/removed from the $k-$th section to the $l-$th section.

Variables influenced by y

y_{1i}: Marshalling degree, which is determined by different operating depth such as group shunting, sort-shunting and the fit degree of the regulation of "Safety terms", for the local district wagon in the $i-$th station.

σ_i: The average time difference among the operations for local wagon flow, local district wagon flow, and long-distance wagon flow in the $i-$th station.

Variables influenced by x and y

q_i: The number of wagons operated in the $i-$th station.

3) Coefficients and constants:

$S = \{s_i, i = 1, 2, \ldots, n\}$: The set of the technic stations administrated by a railway bureau.

$D = \{d_j, j = 1, 2, \ldots, m\}$: The set of the train running sections administrated by a railway bureau.

$D_i = \{d_k, k = 1, 2, \ldots, l\}$: The set of train running sections which are adjacent to Station i.

l_j: The hauling distance in the $j-$th section, which is a constant.

J_w: Railway average tariff, which is a constant.

$\bar{C}_{w0}$: Freight traffic fixed unit cost.

$\bar{C}_{w_{1j}}$: The freight traffic unit cost in the $j-$th section per day per kilometer.

$\bar{C}_{w_{2j}}$: Hauling cost in the $j-$th section per wagon per kilometer.

$\bar{C}_{w'_{2j}}$: The locomotive cost in the $j-$th section per kilometer when there is no wagon hauled by the locomotive.

J_s: Fees charged per wagon.

p_{ju_min}: The minimum wagon flow which can meet the requisite traffic demand required for the $j-$th section in up-direction.

p_{ju_max}: The maximum wagon flow which can be run for the $j-$th section in up-direction.

p_{jd_min}: The minimum wagon flow which can meet the requisite traffic demand required for the $j-$th section in down-direction.

p_{jd_max}: The maximum wagon flow which can be run for the $j-$th section in down-direction.

m_{j_max}: The maximum number of wagons to form a wagon list in the $j-$th section. It is determined by the locomotive hauling limit and the useful length of the receiving and departure tracks in the $j-$th section.

v_i: The maximum possible number of wagons that can be operated by service operation in the $i-$th station.

u_i: The maximum possible number of wagons that can be marshalled in the $i-$th station.

φ_{ju}: The percentage of time that can be used a day (1440 minutes) for freight transportation in the $j-$th section in the up direction.

φ_{jd}: The percentage of time that can be used a day (1440 minutes) for freight transportation in the $j-$th section in the down direction.

t_{T_ju}: The minimum time interval between two wagon lists of the $j-$th section in the up direction regulated by the train working diagram.

t_{T_jd}: The minimum time interval between two wagon lists of the $j-$th section in the down direction regulated by the train working diagram.

Z'_{10i}: The minimum cost for marshalling one local wagon in the $i-$th station. This cost happens in an ideal situation when the number of wagons to be marshaled is large enough and the marshalling degree is deep enough for one marshalling operation.

Z_{11i}: Coefficient for the effect from centralised marshalling operation.

$\bar{q}_{di}$: The number of local wagons to be marshaled for one marshalling operation. It is determined by the loading/uploading capacity in the $i-$th station.

Z_{12i}: Coefficient for the effect from deepened marshalling operation. It is an average additional cost spent for one wagon for marshalling operation.

$A_i, B_i, a'_{1i}, b'_{1i}, b'_{2i}, b'_{3i}, b'_{4i}, a''_{1ik}, b''_{1ik}$,$\alpha'_{21i}, \beta'_{21i}, \beta'_{22i}, \beta'_{23i}$', $\alpha''_{21i}, \beta'_{21i}, \beta''_{22i}, \beta''_{23i}, \alpha'_{3ik}$, β'_{31ik}, β'_{32ik}, $\beta'_{33ik}, \alpha''_{3ik}$, β''_{31ik}, β''_{32ik}, β''_{33ik}: Coefficients which are to be obtained through statistic data.

Z''_{10i}: The minimum cost for marshalling one local district wagon in the $i-$th station. This cost happens in an ideal situation when the number of wagons to be marshaled is large enough and the marshalling degree is deep enough for one marshalling operation.

Z'''_{10i}: The minimum cost for marshalling one long distance wagon in the $i-$th station. This cost happens in an ideal situation when the number of wagons to be marshaled is large enough and the marshalling degree is deep enough for one marshalling operation.

Z'_{20i}: The minimum cost for service operation for one local wagon in the $i-$th station. This cost happens in an ideal situation when marshalling is deep enough such that service operation can be operated easily and conveniently.

Z'_{21i}: The additional cost for service operation for one local wagon in the $i-$th station. This cost happens when marshalling is superficial thus service operation become unhandy.

Z''_{20i}: The minimum cost for service operation for one local district wagon in the $i-$th station. This cost happens in an ideal situation when marshalling is deep enough such that service operation can be operated easily and conveniently.

Z''_{21i}: The additional cost for service operation for one local district wagon in the $i-$th station. This cost happens when marshalling is superficial thus service operation become unhandy.

C_{di}: The coefficient on centralisation and detention for local wagon flow, which is a number between eight and twelve. The number is decided by specialties from different wagon flows.

C_{cHi}: The coefficient on centralisation and detention for long distance wagon flow, which is a number between eight and twelve.

Z'_{3i}: Coefficient on service facilitation for local wagons, which equals the additional halting time when service operation is totally inconvenient.

Z''_{3i}: Coefficient on service facilitation for local district wagons, which equals the additional halting time when service operation is totally inconvenient.

Z_{40i}: The basic cost for adding/removing one wagon to/from a wagon list in the $i-$th station. This cost happens when the number of wagons to be added/removed is large enough.

Z_{41i}: The additional cost for adding/removing one wagon to/from a wagon list in the $i-$th station. This cost happens when the number of wagons to be added/removed is small enough (equaling one).

$\bar{C}_{cx}$: The cost for one wagon to halt one hour, which is caused by wagon depreciation.

S_{il}: The percentage of wagons which have special safety requirement of local wagons in the $i-$th station.

S_{id}: The percentage of wagons which have special safety requirement of local district wagons in the $i-$th station.

T_{si}: Average time to marshal a wagon which has special safety requirement in the $i-$th station.

T_{ilmin}: The least time for marshaling one local wagon list in the $i-$th station.

T_{ilmax}: The time spent for marshaling one local wagon list with the highest specification and completeness in the $i-$th station.

T_{idmin}: The least time for marshaling one local district wagon list in the $i-$th station.

T_{idmax}: The time spent for marshaling one local district wagon list with the highest specification and completeness in the $i-$th station.

4) Formula:

(9.3a) describes a railway bureau's objective which aims at achieving the maximum profit for freight operation administrated by this railway bureau. It has two parts: the profit from the railway network and technical stations in this bureau.

(9.3b) and (9.3c) mean that wagon flows in both up-direction and down-direction have their minimum and maximum limits. Thus the total number of vans from one station to another has its limits too.

(9.3d) and (9.3e) tell how the limits set for wagon flows in up-direction and down-direction are determined and calculated.

(9.3f) and (9.3g) mean the number of long-distance wagon flows to be marshalled in technical stations can not exceed their operating abilities.

(9.4a) is the objective for a technical station, which aims at lowering its operation cost. The operation cost is from three parts: local wagon flow, local district wagon flow, and long-distance wagon flow.

(9.4b) denotes how the operation cost for local wagon flow, local district wagon flow, and long-distance wagon flow are calculated.

(9.4c)-(9.4d) mean there exist minimum and maximum limits for marshalling both local wagon flow and local district wagon flow.

5) Symbols:

F_1: The economical benefit of the railway network within the area administrated by a railway bureau.

F_2: The economical benefit obtained by all of the technical stations administrated by the railway bureau.

$\bar{C}_{w_j}$: The freight traffic unit cost in the $j-$th section.

$\triangle\bar{C}_{w_j}$: Additional unit cost in the $j-$th section.

C_{s_i}: The operating cost in the $i-$th station. It fluctuates with the change of controlling variables from both the leader and the followers.

I_{ju}: The number of wagons that go through the $j-$th section in the up direction per minute.

m_{j_min}: The minimum number of wagons to form a wagon list in the $j-$th section.

I_{jd}: The number of wagons that go through the $j-$th section in the down direction per minute.

C'_{z1i}: Daily cost spent for marshalling local wagon flow for the $i-$th station.

C''_{z1i}: Daily cost spent for marshalling local district wagon flow for the $i-$th station.

C'''_{z1i}: Daily cost spent for marshalling long-distance wagon flow for the $i-$th station.

C'_{z2i}: Daily cost spent for service operation made for local wagon flow for the $i-$th station.

C''_{z2i}: Daily cost spent for service operation made for local district wagon flow for the $i-$th station.

C_{z3i}: Daily cost spent for centralising and detention wagons in the $i-$th station.

$\triangle C_{2i}$: Different sections may have different requests on the number of wagon lists to be run in that section. Thus adding/reducing wagons may be needed in technical stations to meet the requirements of its adjacent sections. $\triangle C_{2i}$ is the daily cost spent for adding/reducing wagons in the $i-$th station.

$\bar{C}'_{z1i}$: Average daily cost spent for marshalling one local wagon for the $i-$th station.

$\bar{C}''_{z1i}$: Average daily cost spent for marshalling one local district wagon for the $i-$th station.

$\bar{C}'_{z2i}$: Average daily cost spent for service operation made for local wagon flow for the $i-$th station.

$\bar{C}''_{z2i}$: Average daily cost spent for service operation made for local district wagon flow for the $i-$th station.

N_i: The total hours spent by all wagons which are halted in the $i-$th station.

C'_{fi}: The number of additional hours spent for service operation for a local wagon in the $i-$th station.

C''_{fik}: The number of additional hours spent for service operation for a local district wagon to the $k-$th direction in the $i-$th station.

Z_{4i}: The cost spent for adding/removing one wagon.

9.2.4 Experiments

In this section, we consider the RWFM problem in a railway bureau "Bureau A". Within the area administrated by Bureau A, there are three technical stations: Station A, Station B, and Station C. Connecting these stations, we have three sections: Section A that connects Station A and Section B, Section B that connects Station B and Section C, and Section C that connects Station C and Station A. We list the values of some of the main coefficients, which are used to build up the RWFM OLMFB decision model, in Table 9.2 and 9.3.

Table 9.2: Summary of the coefficient values in the case study - 1

Stations	p_{ju_min}	p_{ju_max}	p_{jd_min}	p_{jd_max}	u_i	v_i
A	10	29	10	29	19	19
B	10	29	10	29	19	19
C	10	29	10	29	19	19

Table 9.3: Summary of the coefficient values in the case study - 2

Sections	T_{ilmin}	T_{ilmax}	T_{idmin}	T_{idmax}
A	15 minutes	20 minutes	15 minutes	22 minutes
B	15 minutes	23 minutes	15 minutes	25 minutes
C	15 minutes	22 minutes	15 minutes	21 minutes

To help the decision maker in Bureau A make an optimal RWFM plan, we use the PSO-based algorithm developed in Section 7.2.2, which was implemented by Visual Basic 6.0, and tested on a desktop computer with CPU Pentium 4 2.8GHz, RAM 1G, Windows XP. By 342 seconds running, the solutions for Bureau A are reached and summarised in Table 9.4.

Table 9.4: Summary of the solutions for Bureau A and the Stations

Stations	p_{ju}	p_{jd}	$q_{di} + q'_{di}$	q_{2i}	x_{2k}	y_{1Gil}	y_{1Sil}	y_{1i}
A	24	28	42	10	18	0.2	0.78	0.41
B	27	23	38	12	11	0.55	0.67	0.35
C	24	17	25	16	7	0.53	0.98	0.48

Table 9.5: Summary of the solution differences

Stations	p_{ju}	p_{jd}	$q_{di} + q'_{di}$	q_{2i}	x_{2k}	y_{1Gil}	y_{1Sil}	y_{1i}
A	0.002	0.12	0.1	0.05	0.37	0.16	0.00043	0.1
B	0.04	0.29	0.02	0.27	0.42	0.33	0.01	1.23
C	0.17	0.37	0.42	0.07	0.57	0.02	0.27	0.09

To test the stability of this PSO-based algorithm, this case has been run six times by the algorithm. The solution variances are summarised in Table 9.5.

In Table 9.5, we can see that, there is no tremendous diversion among the solutions obtained. For every running, the solution has been obtained within 400 seconds. Thus, we can come to the conclusion that the PSO-based algorithm proposed in this study could explore veracious solutions for RWFM problems with quite effective and stable performance.

9.2.5 Conclusions

In this research, the bi-level nature of RWFM has been put forward. A RWFM OLMFB decision model has been developed. We apply this model and the PSO-based algorithm on a case study. Experiment results show that the approach proposed in this research could effectively deal with RWFM problems. However, as a lot of practical details have to be ignored for articulating the model building, this decision model has its limitations when applied directly for RWFM decision making. Future efforts will be focused on relating more practical and randomly occurred issues from field works.

9.3 Competitive Strategic Bidding Optimisation in Electricity Markets [1]

9.3.1 Background

Throughout the world, electric power industries are undergoing enormous restructuring processes from nationalised monopolies to individual organisations in a competitive market (Huang & Pai 2002) with the support of digital ecosystems. Because of the significance and particularity of electricity energy to national economics and society (Guerrero, Hang & Uceda 2008), electricity markets must be operated under extensive conditions of absolute security and stabilisation. The research on electricity markets has attracted many researchers, owners and managers from electricity entities and authorities with the development of current digital ecosystems. The competitive mechanism of day-ahead markets is one very important research issue in the electricity market study, which can be described as follows. Each generating company submits a set of hourly (half-hourly) generation prices and the available capacities for the following day. According to this data and an hourly (half-hourly) load forecast, a market operator allocates generation output for each unit.

Nowadays, the viewpoint that electric power industries are monopoly industries is in doubt and has been extensively challenged. Many countries are considering changing the operation of their electric power industries. The aim of marketing electric power industries is to break the monopoly and introduce competition. As no determinate operation model for electricity markets exists, the marketing procedure of electric power industries varies from country to country. Generally speaking, there are three kinds of running models in electricity markets: the power pool model, wholesale competitive model, and retail competitive model. These models adopt three kinds of electric power trading methods: long term contract, day-ahead market, and facility service. Among them, day-ahead market is the most competitive and active part, which imposes great influence on profits for each participant in the market. Specifically, each generating company submits a set of generation prices and other related data, based on which the market operator makes a generating plan for the following day. To optimise this procedure, many models and algorithms have been proposed.

A lot of research has been done on how to strategically bid prices for those generating companies, and how to dispatch generation output for market operators to each

[1] The work presented in this section is a joint research with Professor Guoli Zhang, from the Department of Mathematics and Physics, North China Electric Power University, China.

of their units. David and Wen (David & Wen 2000) conducted a literature survey on strategic bidding in competitive electricity markets. Literature (Li & Shahidehpour 2007; Haghighat, Seifi & Ashkan 2007) uses supply function equilibrium model to maximise generating companies' profits and obtain a Nash equilibrium in a day-ahead electricity market. Literature (Wen & Kumar 2001; Weber & Overbye 2002; Pang & Fukushima 2005) uses game theory (Munz, P.Schumm, A.Wiesebrock & Allgower 2007) to build a strategic bidding model for generating companies and reach a Nash equilibrium solution. However, these models do not include ramp rate constraints, which are crucial to guarantee a real optimal solution. In addition, because strategic bidding problems involve two hierarchical optimisations, and are different from a conventional game model, a new Nash equilibrium is needed as a solution. Furthermore, because strategic bidding problems involve two hierarchical optimisations, and there exists a game relationship among the upper partners, the decision from every game player (generating company) will be influenced by other players (generating companies) as well as by the generating dispatch policy from the lower partner (the market operator). This problem is different from a conventional game model, and a new Nash equilibrium is thus needed as a solution. From the literature, only Pang and Fukushima (Pang & Fukushima 2005) have discussed the generalised Nash equilibrium concept. However, their method is still not suitable for describing a strategic bidding problem in electricity markets. Literature (Bjondal & Jornsten 2005) has used a bi-level optimisation method to build a generation output allocation model, but does not consider competitive bidding problems from generating companies. Literature (Fampa, Barroso, Candal & Simonetti 2008) has built a competitive strategic bidding model using bi-level optimisation by means of the Cournot and Bertrand model, but did not include ramp rate constraints in their model.

Compared with current research on bidding problems in electricity markets, this study applies bi-level techniques in electricity markets to build up an MLOFB decision model for strategic bidding problems including the ramp rate constraints.

9.3.2 Bidding Strategy Analysis in Competitive Electricity Markets

In an auction-based day-ahead electricity market, each generating company will try to maximise its own profit by strategic bidding. Normally, each generating company submits a set of hourly (half-hourly) generation prices and available capacities for the following day. Based on this data and an hourly-load (half-hourly-load) forecast, a market operator will allocate generation output. In this section, under the analysis of bidding strategy optimisation problems, we build a competitive strategic bidding model for generating companies and a generation output dispatch model for a market operator in a day-ahead electricity market.

A Strategic Pricing Model for Generating Companies

In the upper level, each generating company is concerned with how to choose a bidding strategy, which includes generation price and available capacity. Many bidding functions have been proposed. For a power system, the generation cost function generally adopts a quadratic function of the generation output, i.e. the generation cost function can be represented as

$$C_j(P_j) = a_j P_j^2 + b_j P_j + c_j \tag{9.5}$$

where P_j is the generation output of generator j, and a_j, b_j, c_j are co-efficient of generation cost function of generator j.

The marginal cost of generator j is calculated by:

$$\lambda_j = 2a_j P_j + b_j \tag{9.6}$$

It is a linear function of its generation output P_j. The rule in a goods market may expect each generating company to bid according to its own generation cost. Therefore, we adopt this linear bid function. Suppose the bidding for the j-th unit at time t is

$$R_{tj} = \alpha_{tj} + \beta_{tj} P_{tj} \tag{9.7}$$

where $t \in T$ is the time interval, T is time interval number, j represents the unit number, P_{tj} is the generation output of unit j at time t, and α_{tj} and β_{tj} are the bidding coefficients of unit j at time t.

According to the justice principle of "the same quality, the same network, and the same price", we adopt a uniform marginal price (UMP) as the market clearing price. Once the energy market is cleared, each unit will be paid according to its generation output and UMP. The payoff of the i-th generating company is:

$$F_i = \sum_{t=1}^{T} \Big(\sum_{j \in G_i} UMP_t P_{tj} - \sum_{j \in G_i} (a_j P_{tj}^2 + b_{tj} P_{tj} + c_{tj}) \Big) \tag{9.8}$$

where G_i is the suffix set of the units belonging to the i-th generating company. Each generating company wishes to maximise its own profit F_i. In fact, F_i is the function of P_{tj} and UMP_t, and UMP_t is the function of all units' bidding α_{tj}, β_{tj} and output power P_{tj}, which will impact on each other. Therefore, we establish a strategic pricing model for generating companies as follows:

$$\begin{aligned}\max_{\alpha_{tj},\beta_{tj},j\in G_i} F_i &= F_i(\alpha_{t1},\beta t1,\cdots,\alpha_{tN},\beta_{tN},P_{t1},\cdots,P_{tN})\\ &= \sum_{t=1}^{T}(UMP_tP_{ti}-\sum_{j\in G_i}(a_jP_{tj}^2+b_jP_{tj}+c_j)) \qquad (9.9)\\ i &= 1,2,\cdots,L \qquad (9.10)\end{aligned}$$

where L is the number of generating companies, $P_tj = \sum_{j\in G_i} P_{tj}$, $t = 1,2,\cdots,T$.

The profit calculated for each generating company will consider both P_{tj} and UMP_t, which can be computed by a market operator, according to the market clearing model.

A Generation Output Dispatch Model for a Market Operator

A market operator actually represents the consumer electricity purchase from generating companies, under the conditions of security and stabilisation. The objective of a market operator is to minimise the total purchase fare, while encouraging generating companies to use a bid price as low as possible. It is reasonable that the lower the price, the more the output. Thus, the function value of a market operator's objective will be calculated according to the bidding price. Most previous strategic bidding models do not include ramp rate constraints, without which the solution for generating dispatch may not be a truly optimal one. We should consider the ramp rate constraints in the real world when modelling a generating dispatch. However, if a model includes ramp rate as a constraint, the number of decision variables involved in the problem will increase dramatically, which requires a more powerful solution algorithm. Based on the analysis above, we build a market operator's generation output dispatch model as follows:

$$\min_{P_{tj}} f = f(\alpha_{t1},\beta_{t1},\cdots,\alpha_{tN},\beta tN,P_{t1},\cdots,P_{tN}) = \sum_{t=1}^{T}\sum_{j=1}^{N}R_{tj}P_{tj} \qquad (9.11a)$$

$$\text{subject to } \sum_{j=1}^{N}P_{tj} = P_{tD} \qquad (9.11b)$$

$$P_{jmin} \leqslant P_{tj} \leqslant P_{jmax} \qquad (9.11c)$$

$$-D_j \leqslant P_{tj} - P_{t-1,j} \leqslant U_j, t = 1, 2, \cdots, T \tag{9.11d}$$

where $t \in T$ is the time interval, T is the time interval number, j represents the unit number, P_{tj} is the generation output of unit j at the time t, and α_{tj} and β_{tj} are the bidding co-efficients of unit j at the time t, P_{tD} is the load demand at the time t, P_{jmin} is the minimum output power of the j-th unit, P_{jmax} is the maximum output power of the j-th unit, D_j is the maximum downwards ramp rate of the j-th unit, and U_j is the maximum upwards ramp rate of the j-th unit.

After receiving all generating companies' bid data, a market operator determines the output power of each unit and UMP_t in time slot t. UMP_t can be calculated according to the following steps:

Step 1 calculate output power of each unit j for all time slot t using formula 9.11;

Step 2 compute bidding R_{tj} corresponding to the generation output P_{tj};

Step 3 account $UMP_t = \max_{j=1}^{N} R_{tj}$.

9.3.3 An MLOFB Decision Model in Competitive Electricity Markets

From the analysis above, we know that in an auction-based day-ahead electricity market, each generating company tries to maximise its own profit by strategic bidding, and each market operator tries to minimise its total electricity purchase fare. The decision of one will influence the other. This is a typical bi-level decision problem, which has multi-leaders and only one follower, with generating companies as leaders and a market operator as a follower.

By combining the strategic pricing model defined in (9.9) with the generation output dispatch model defined in (9.11), we establish an MLOFB decision model for competitive strategic bidding-generation output dispatch in an auction-based day-ahead electricity market as follows:

$$\begin{aligned}\max_{\alpha_{tj},\beta_{tj},j\in G_i} F_i &= F_i(\alpha_{t1}, \beta t1, \cdots, \alpha_{tN}, \beta_{tN}, P_{t1}, \cdots, P_{tN}) \\ &= \sum_{t=1}^{T}(UMP_t P_{ti} - \sum_{j\in G_i}(a_j P_{tj}^2 + b_j P_{tj} + c_j)) && (9.12) \\ & i = 1, 2, \cdots, L && (9.13)\end{aligned}$$

$$\min_{P_{tj}} f = f(\alpha_{t1}, \beta_{t1}, \cdots, \alpha_{tN}, \beta tN, P_{t1}, \cdots, P_{tN}) = \sum_{t=1}^{T}\sum_{j=1}^{N} R_{tj}P_{tj} \tag{9.14}$$

$$\text{subject to } \sum_{j=1}^{N} P_{tj} = P_{tD} \tag{9.15}$$

$$P_{jmin} \leqslant P_{tj} \leqslant P_{jmax} \tag{9.16}$$

$$-D_j \leqslant P_{tj} - P_{t-1,j} \leqslant U_j, t = 1, 2, \cdots, T \tag{9.17}$$

where α_{tj} and β_{tj} are the bidding co-efficients of unit j at time t, α_{tmin}, α_{tmax}, β_{tmin}, β_{tmax} are the lower and upper limits for α_{tj} and β_{tj} respectively, L is the number of generating companies, $P_{tj} = \sum_{j \in G_L} P_{tj}$, P_{jmin} is the minimum output power of the j-th unit, P_{jmax} is the maximum output power of the j-th unit, D_j is the maximum downwards ramp rate of the j-th unit, and U_j is the maximum upwards ramp rate of the j-th unit.

9.3.4 Experiments

In this section, we employ a real world strategic bidding problem in an electricity market to test the MLOFB decision model and use the PSO-based algorithm developed in section 7.1.2 to solve the problem.

Test Data

In order to test the effectiveness of the proposed bi-level decision model and the PSO-based algorithm when solving the model defined by (9.12), a typical competitive strategic bidding case consisting of three companies with six units and twenty-four time-intervals is chosen. The generation cost function can be calculated by using (9.5), where the cost co-efficients of unit j and other technical data are given in Table 9.6, the load demands for each time interval t are given in Table 9.7.

In Table 9.6, Units 1 and 2 belong to the first generating company, Units 3 and 4 belong to the second generating company, and Units 5 and 6 belong to the third generating company.

To simplify computation, the limit of strategic bidding coefficients does not vary by different time slots and we suppose: $\alpha_{tmin} = 7, \alpha_{tmax} = 9, \beta_{tmin} = 0.0002, \beta_{tmax} = 0.007, t = 1, 2, \cdots, T, j = 1, 2, \cdots, N$.

Table 9.6: Technical data of units - 1

Unit No.	a_j	P_{min} (MW)	P_{max} (MW)	b_j	c_j	D_j (MW/h)	U_j (MW/h)
1	0.00028	50	680	4.10	150	80	85
2	0.00312	30	150	4.50	80	45	60
3	0.00048	50	360	4.10	109	60	65
4	0.00324	60	240	3.74	125	45	80
5	0.00056	60	300	3.82	130	70	80
6	0.00334	40	160	3.78	100	55	40

Table 9.7: Technical data of units - 2

t	1	2	3	4	5	6	7	8
P_{tD}	1033	1000	1013	1027	1066	1120	1186	1253
t	9	10	11	12	13	14	15	16
P_{tD}	1300	1340	1313	1313	1273	1322	1233	1253
t	17	18	19	20	21	22	23	24
P_{tD}	1280	1433	1273	1580	1520	1420	1300	1193

Experiment Results

This example is run by the PSO-based algorithm proposed in Section 7.1.2, which was implemented by Visual Basic 6.0, and tested on a desktop computer with CPU Pentium 4, 2.8GHz, RAM 1G, Windows XP. The running results are listed from Table 9.8 to Table 9.12.

Under these solutions, the objective values for both the leaders and the follower are listed in Table 9.12.

Experiment Analysis and Evaluation

In this section, a strategic bidding problem in an electricity market is employed. By the PSO-based algorithm, solutions are reached for both the generating companies and the market operator to help them make strategic decisions. From the experiment, we can see that the MLOFB decision model can effectively model strategic bidding problems from electricity markets. Existing models either fail to deal with the gaming relationships among generating companies or ignore the hierarchical nature between generating companies and a market operator. By considering the gaming and bi-level relationships among generating companies and market operators simultaneously, the

Table 9.8: Running results for α_{tj} from the example

t(column) j (row)	1	2	3	4	5	6
1	7.37	7.61	7.32	7.03	7.27	8.98
2	8.76	7.74	8.71	7.15	8.13	7.10
3	7.18	8.89	8.60	8.84	8.55	8.26
4	7.36	8.33	7.31	8.28	8.73	7.70
5	7.10	8.80	8.51	8.75	8.46	8.17
6	8.59	7.57	8.54	8.98	7.96	8.93
7	7.11	7.35	7.06	8.77	7.01	8.72
8	8.45	8.90	7.87	8.85	7.82	8.80
9	8.29	8.00	8.24	7.95	7.66	7.37
10	8.42	7.39	8.37	8.81	7.79	8.76
11	7.57	7.28	7.52	7.23	8.94	7.18
12	7.02	7.99	8.97	7.94	8.39	7.36
13	7.49	7.20	7.44	7.15	8.86	7.10
14	8.25	7.22	8.20	8.64	7.62	8.59
15	8.04	7.74	7.98	7.69	7.40	7.11
16	8.11	8.56	7.53	8.51	7.48	8.45
17	8.68	8.92	8.63	8.34	8.58	8.29
18	8.08	7.05	8.03	8.47	7.45	8.42
19	8.50	8.21	7.92	8.16	7.86	7.57
20	8.68	7.65	8.63	7.07	8.04	7.02
21	8.41	8.12	7.83	8.07	7.78	7.49
22	7.91	8.88	7.33	8.30	7.27	8.25
23	8.43	8.67	8.38	8.09	8.33	8.04
24	7.24	8.21	7.19	8.16	7.14	8.11

MLOFB decision model can better reflect the features of real-world strategic bidding problems in electricity markets and format these problems more practically.

9.3.5 Conclusions

With the development of digital ecosystems, competitive strategic bidding optimisation of generating companies in electricity markets becomes more practically important and technically implementable. This section applies bi-level programming and swarm algorithms to model competitive strategic bidding decision-making in electricity markets and obtain solutions. Experiment results show that the proposed PSO-based algorithm can achieve a generalised Nash equilibrium for an MLOFB problem in an electricity market by providing generating companies with competitive strategic

Table 9.9: Running results for β_{tj} from the example

t(column) j (row)	1	2	3	4	5	6
1	0.00089	0.00205	0.00321	0.00438	0.00554	0.00670
2	0.00208	0.00074	0.00440	0.00307	0.00673	0.00539
3	0.00641	0.00077	0.00193	0.00310	0.00426	0.00542
4	0.00278	0.00644	0.00510	0.00377	0.00063	0.00609
5	0.00048	0.00164	0.00280	0.00397	0.00513	0.00629
6	0.00382	0.00249	0.00615	0.00481	0.00348	0.00034
7	0.00350	0.00467	0.00583	0.00699	0.00135	0.00251
8	0.00022	0.00568	0.00435	0.00121	0.00667	0.00353
9	0.00687	0.00123	0.00420	0.00536	0.00653	0.00089
10	0.00556	0.00423	0.00109	0.00655	0.00522	0.00208
11	0.00060	0.00176	0.00292	0.00408	0.00525	0.00641
12	0.00626	0.00493	0.00179	0.00045	0.00411	0.00278
13	0.00147	0.00263	0.00379	0.00496	0.00612	0.00048
14	0.00051	0.00597	0.00464	0.00150	0.00696	0.00382
15	0.00449	0.00565	0.00682	0.00118	0.00234	0.00350
16	0.00551	0.00237	0.00103	0.00469	0.00336	0.00022
17	0.00106	0.00222	0.00339	0.00455	0.00571	0.00687
18	0.00406	0.00092	0.00638	0.00324	0.00191	0.00556
19	0.00658	0.00094	0.00391	0.00507	0.00624	0.00060
20	0.00295	0.00162	0.00527	0.00394	0.00080	0.00626
21	0.00065	0.00362	0.00478	0.00595	0.00031	0.00147
22	0.00580	0.00266	0.00132	0.00498	0.00365	0.00051
23	0.00368	0.00484	0.00600	0.00036	0.00333	0.00449
24	0.00220	0.00586	0.00452	0.00138	0.00684	0.00551

Table 9.10: Running results for UMP_t from the example

t =1	t =2	t =3	t =4	t =5	t =6	t =7	t =8
17.81	8.62	1.49	8.19	2.77	4.35	14.31	4.43
t =9	t =10	t =11	t =12	t =13	t =14	t =15	t =16
8.75	6.40	13.45	12.30	1.06	9.47	18.93	15.88
t =17	t =18	t =19	t =20	t =21	t =22	t =23	t =24
14.39	18.87	5.42	11.10	19.35	14.60	18.23	7.34

bidding within network security constraints.

Table 9.11: Running results for P_{tj} from the example

t(column) j (row)	1	2	3	4	5	6
1	493	150	50	240	60	40
2	445	145	63	232	70	45
3	443	140	76	224	80	50
4	442	135	89	216	90	55
5	466	130	102	208	100	60
6	505	125	115	200	110	65
7	556	120	128	192	120	70
8	608	115	141	184	130	75
9	640	110	154	176	140	80
10	665	105	167	168	150	85
11	623	100	180	160	160	90
12	593	110	193	152	170	95
13	538	105	206	144	180	100
14	478	108	231	165	210	130
15	412	109	252	150	200	110
16	333	130	285	180	210	115
17	275	130	320	210	220	125
18	268	150	360	240	260	155
19	322	120	300	200	211	120
20	390	150	360	240	290	150
21	326	150	355	230	299	160
22	266	143	356	240	270	145
23	191	120	320	239	280	150
24	207	100	300	200	254	132

Table 9.12: Objective values for the decision makers

The 1st generating company	The 2nd generating company	The 3rd generating company	The market operator
73313	65799	46376	225272

9.4 Bi-level Pricing Models in a Supply Chain [2]

9.4.1 Background

Due to rapid technological innovation and severe competition, in hi-tech industries like computers and communication, the upstream component price and the downstream product cost usually decline significantly with time. In such a background, an effective pricing supply chain model becomes crucial. In a supply chain, both a buyer and a vendor aim to maximise their profits in the inventory system but their decisions are related with each other in a hierarchical way. By taking a buyer and a vendor as the leader respectively, this section establishes two bi-level pricing models for pricing problems for a buyer and a vendor in a supply chain. These two models consider a buyer and a vendor as the leader and the follower alternatively, allowing them make decisions sequentially and fully considering the mutual influences from each other.

9.4.2 Bi-level Pricing Models

In this section, by switching the leader/follower role respectively between a buyer and a vendor, two bi-level pricing models are developed for them in a supply chain.

In a buyer-vendor system, a buyer's net profit can be calculated by Yang et al. (2007):

$$\begin{aligned} NP_b = & \frac{P_{m0}D}{\ln(1-r_m)}[e^{H\ln(1-r_m)} - 1] - P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} \\ & - \frac{F_bHP_{b0}Q}{2mn}\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - mnC_b \end{aligned} \tag{9.18}$$

A vendor's net profit can be calculated by Yang et al. (2007):

$$\begin{aligned} NP_v = & P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - P_{v0}mQ\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} \\ & - \frac{F_vHP_{v0}(m-1)Q}{2n}\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - nC_v \end{aligned} \tag{9.19}$$

In (9.18), a buyer controls m, the number of buyer's lot size deliveries per vendor's lot size; and r_m, the weekly decline-rate of market price to the end-consumer. In (9.19), a vendor controls n, the number of orders that a vendor places for the item

[2]The work presented in this section is a joint research with Professor Hui-Ming Wee, from the Department of Industrial and Systems Engineering, Chung Yuan Christian University, Taiwan.

from a supplier in the planning horizon; r_b, the weekly decline-rate of the buyer's purchase cost; and r_v, the weekly decline-rate of the vendor's purchase cost. All other parameters defined in the problem are constants. The explanations of symbols used in the above two formulas are listed in Table 9.13.

Table 9.13: Explanations on symbols used in (9.18) and (9.19)

n	number of orders that a vendor places for the item from a supplier in the planning horizon
m	the number of a buyer's lot size deliveries per vendor's lot size
Q	a buyer's lot size
r_b	the weekly decline-rate of a buyer's purchase cost
D	the weekly demand rate
r_v	the weekly decline-rate of a vendor's purchase cost
r_m	the weekly decline-rate of market price to an end-consumer
H	the weekly length of the planning horizon
F_v	a vendor's holding cost per dollar per week
F_b	a buyer's holding cost per dollar per week
C_v	a vendor's ordering cost per order
C_b	a buyer's ordering cost per order
P_{v0}	a vendor's unit purchase cost at the initial time
P_{b0}	a buyer's unit purchase cost at the initial time
P_{m0}	the market price to an end consumer at the initial time
$P_v(t)$	a vendor's unit purchase cost in week t
$P_b(t)$	a buyer's unit purchase cost in week t
$P_m(t)$	a market price to an end consumer in week t
NP_v	a vendor's net profit in the planning horizon
NP_b	a buyer's net profit in the planning horizon
NP	the joint net profit of both a vendor and a buyer in the planning horizon

When making the pricing strategy, if we take the point of view from a buyer to make his or her profit a priority over a vendor, we can make the buyer as the leader and a vendor as the follower. By combining Formulas (9.18) and (9.19), we establish a bi-level pricing model in a supply chain as follows:

$$\max_{m,r_m} NP_b(m,r_m,n,r_b,r_v) = \frac{P_{m0}D}{\ln(1-r_m)}[e^{H\ln(1-r_m)}-1] - P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}}$$
$$-\frac{F_bHP_{b0}Q}{2mn}\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - mnC_b \tag{9.20a}$$

subject to $m > 0$

$$0.0001 \leqslant r_m \leqslant 0.5 \tag{9.20b}$$

$$\max_{n,r_b,r_v} NP_v(m,r_m,n,r_b,r_v) = P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - P_{v0}mQ\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}}$$
$$-\frac{F_vHP_{v0}(m-1)Q}{2n}\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - nC_v \tag{9.20c}$$

subject to $n > 0$

$$0.0001 \leqslant r_b \leqslant 0.5$$
$$0.0001 \leqslant r_v \leqslant 0.5 \tag{9.20d}$$

In this model, both a buyer and a vendor adjust their own controlling variables respectively, wishing to maximise their own profits, under specific constraints. The buyer is the leader, who makes decision first; and the vendor is the follower, who makes decision after the buyer.

If we take the point of view from a vendor to make his or her profit a priority over a buyer, we can make the vendor as the leader and a buyer as the follower. By combining (9.18) and (9.19), we establish another bi-level pricing model in a supply chain as follows:

$$\max_{n,r_b,r_v} NP_v(m,r_m,n,r_b,r_v) = P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - P_{v0}mQ\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}}$$
$$-\frac{F_vHP_{v0}(m-1)Q}{2n}\frac{1-(1-r_v)^H}{1-(1-r_v)^{\frac{H}{n}}} - nC_v \tag{9.21a}$$

subject to $n > 0$

$$0.0001 \leqslant r_b \leqslant 0.5$$
$$0.0001 \leqslant r_v \leqslant 0.5 \tag{9.21b}$$

$$\max_{m,r_m} NP_b(m, r_m, n, r_b, r_v) = \frac{P_{m0}D}{\ln(1-r_m)}[e^{H\ln(1-r_m)} - 1] - P_{b0}Q\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}}$$
$$- \frac{F_bHP_{b0}Q}{2mn}\frac{1-(1-r_b)^H}{1-(1-r_b)^{\frac{H}{mn}}} - mnC_b \tag{9.21c}$$

subject to $m > 0$

$$0.0001 \leqslant r_m \leqslant 0.5 \tag{9.21d}$$

In this model, both a buyer and a vendor adjust their own controlling variables respectively, wishing to maximise their own profits, under specific constraints. The vendor is the leader, who makes decision first; and the buyer is the follower, who makes decision after the buyer.

9.4.3 Experiments

In this section, we illustrate the models developed in this study by the following numerical example where the parameters are given as follows:

(1) The demand rate per week, $D = 400$ units

(2) The vendor's unit purchase cost at the initial time, $P_{v0} = \$4$

(3) The buyer's unit purchase cost at the initial time, $P_{b0} = \$5$

(4) The market price to the end consumer from the buyer at the initial time, $P_{m0} = \$6$

(5) The buyer's ordering cost per order, $C_b = \$30$

(6) The vendor's ordering cost per order, $C_v = \$1,000$

(7) The buyer's holding cost per dollar per week, $F_b = 0.004$

(8) The vendor's holding cost per dollar per week, $F_v = 0.004$

(9) Time horizon considered, $H = 52$ weeks

Yang et al.(2007) deals with this problem by solving a single level optimisation problem: $NP = NP_b + NP_v$, where only the net profit of a buyer and a vendor must be the same and only m and n are adjustable decision variables. We relax the constraint of equal profit, and add r_m, r_b, and r_v as decision variables. By using the PSO-based algorithm in Section 7.2.2 to solve problems defined by Formulas (9.20) and (9.21), we obtain solutions for both the buyer and vendor. To evaluate the results of this research, we compare these results with the results from the original model by Yang et al. (2007) under a different negotiation factor α, which is defined as $\alpha = NP_v/NP_b$. To make the comparison fair and reasonable, besides m and n, we add r_m, r_b, and r_v as decision variables to be changeable to maximise the profit in Yang et al's model (2007). Table 9.14 lists solutions from this research and solutions from the model by Yang et al. (2007).

Table 9.14: Summary and comparison of running results

	m	r_m	N	r_b	r_v	NP_b	NP_v
Yang et al. (2007) ($\alpha \geqslant 2$)	2	0.0001	9	0.0068	0.5	35,008	69,946
Yang et al. (2007) ($1.5 \leqslant \alpha \leqslant 2$)	2	0.0001	9	0.01	0.5	41,280	63,710
Yang et al. (2007) ($1 \leqslant \alpha \leqslant 1.5$)	2	0.0001	9	0.017	0.5	52,990	52,068
Yang et al. (2007) ($0.5 \leqslant \alpha \leqslant 1$)	1	0.0001	9	0.032	0.5	68,548	36,605
Yang et al. (2007) ($\alpha < 0.5$)	Not applicable						
This study (buyer as leader)	5	0.0071	6	0.0372	0.0753	52,399	16,866
This study (vendor as leader)	3	0.0015	7	0.0026	0.0767	21,359	64,165

From Table 9.14, we can see that, using the bi-level pricing model (buyer as leader) developed in this paper, the buyer's profit will increase compared with Yang's model when $\alpha \geqslant 0.5$. If the vendor is taken as the leader, he or she can achieve a profit increase when $\alpha \leqslant 2$, which is true for most pricing problems in a supply chain. As the follower, the vendor or the buyer is bound to lose, despite the range of the negotiation factor α. This is understandable, because in a bi-level decision situation, we always take the leader's interest as a priority.

These results reveal that when applying bi-level programming technologies on pricing problems in supply chains, some improvements can be achieved for a play (a buyer or a vendor) if he or she is the leader.

In the two-stage vendor-buyer inventory system, our experimental data show that the vendor, as leader, outperforms the buyer as leader. This is because a vendor, as the leader, improves the actual consumption rates; the vendor making the first decision ensures that production matches demand more closely, reduces inventory and improves business performance. This is why the VMI (vendor managed inventory) has become very popular in recent years.

9.5 Summary

In this chapter, we apply bi-level programming techniques on three application fields of railway transportation, electricity markets and supply chains. To begin with, a non-linear bi-level decision model is used to analyse railway train set organising optimisation problems. Then, an OLMFB decision model and an MLOFB decision model are built up on railway wagon flow management and day-ahead electricity markets respectively. Finally, bi-level decision models are established for pricing problems in supply chains. Addressed to these real world problems, the PSO-based algorithms are used to reach solutions for decision makers in railway industries, electricity markets and supply chains. The experiments show that these bi-level decision models and algorithms are quite reasonable by effectively providing optimisation solutions.

10 Summary and Future Study

10.1 Summary of this Thesis

This thesis studies on the topic of bi-level decision making with different features: multiple leaders/followers/objectives, fuzzy coefficients, arbitrary formats for objectives and constraints. By combining these features, seven bi-level decision problems are addressed in this thesis: FLB decision problems, FMOLB decision problems, FMMLB decision problems, FLBG decision problems, MLOFB, OLMFB, and MLMFB decision problems.

Based on the modelling of these seven bi-level decision problems by $\lambda-$cut and the Nash equilibrium concept, corresponding algorithms are proposed. By implementing these algorithms, an FBDSS is developed to support bi-level decision making. Applying the bi-level programming techniques developed in this study, bi-level problems in the fields of railway transportation, electricity markets and supply chains are explored and solved by the FBDSS.

10.2 Future Study

Our future research on bi-level decision making can take many directions. Some are listed below:

(1) **Multi-level decision problems.** Bi-level decision making is only a special case of multi-level decision making. To extend current bi-level programming techniques to deal with more complicated situations on three or more level optimisation problems will be one of our future studies.

(2) **Bi-level decision problems which are hard, or impossible, to describe by mathematical optimisational models.** Existing bi-level programming techniques have almost exclusively focused on bi-level decision problems whose objective and constraint functions can be generalised as specific mathematical forms. However, most bi-level applications only have the information stored in

large databases, from which it is almost impossible to generate their mathematical definitions. Such a situation has frequently appeared in real world problems. Unfortunately, modelling and solving a formless bi-level decision problem has not received much attention in the research literature. To deal with bi-level decision problems which cannot be modelled by mathematical forms will be a new direction in bi-level decision making in the future.

(3) **Further explorations on real world bi-level problems.** Many real world decision problems have hierarchical features, such as 1) optimisation units exist within a predominantly hierarchical structure; 2) each lower level executes its plans after, and in view of, decisions made at the upper level; 3) each unit independently optimises its objective but is affected by the actions of other units; 4) all relevant information is stored in a database from which it is very difficult to generalise the mathematical formulations for the problem. However, to model and solve them by appropriate bi-level programming techniques is still a challenge, as many special situations exist for a real world optimisation problem. Our future effort will be channelled into applying the existing bi-level programming techniques to real world bi-level decision problems, and adapting or extending existing bi-level programming techniques to real world bi-level decision problems.

References

Aiyoshi, E. & Shimizu, K. 1981, 'Hierarchical decentralized systems and its new solution by a barrier method', *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 444–449.

Anandalingam, G., Mathieu, R., Pittard, L. & Sinha, R. 1992, 'Artificial intelligence based approaches for hierarchical optimization', *Annals of Operations Research*, vol. 34, pp. 1–4.

Augusto, O., Rabeau, S., Dépincé, P. & Bennis, F. 2006, 'Multi-objective genetic algorithms: A way to improve the convergence rate', *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 501–510.

Bard, J. 1983, 'An efficient point algorithm for a linear two-stage optimization problem', *Operation Research*, vol. 38, pp. 556–560.

Bard, J. 1984a, 'Optimality conditions for the bilevel programming problems', *Naval research logistics quarterly*, vol. 31, pp. 13–26.

Bard, J. F. 1984b, 'An investigation of the linear three level programming problem', *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 14, pp. 711–717.

Bard, J. F. 1991, 'Some properties of the bilevel programming problem', *J. of Optimiz. Theory & Appl.*, vol. 68, no. 2, pp. 371–378.

Bard, J. F. 1998, *Practical bilevel optimization: algorithms and applications*, Kluwer Academic Publishers, Boston.

Bard, J. F. & Moore, J. T. 1990, 'A branch and bound algorithm for the bilevel programming problem', *SIAM Journal on Scientific and Statistical Computing*, vol. 11, pp. 281–292.

Bem-Ayed, O. 1993, 'Bilevel linear programming', *Computers Ops. Res.*, vol. 20, no. 5, pp. 485–501.

Ben-Ayed, O., Boyce, D. E. & Blair, C. E. 1988, 'A general bilevel linear programming formulation of the network design problem', *Transportation Research Part B: Methodological*, vol. 22, no. 4, pp. 311–318.

Berezinski, M., Holubiec, J. & Petriczek, G. 1994, 'Multilevel models for railway traffic control', in *Control'94*, pp. 54–56.

Bialas, W. & Karwan, M. 1978, 'Multilevel linear programming', *Tech. Rep. 78-1*, University of New York at Buffalo, operations Research Program.

Bialas, W. & Karwan, M. 1984a, 'Two-level linear programming', *Management Science*, vol. 30, pp. 1004–1020.

Bialas, W. F. & Karwan, M. H. 1982, 'On two-level optimization', *IEEE Trans. Auto. Cont.*, vol. AC-27, pp. 211–214.

Bialas, W. F. & Karwan, M. H. 1984b, 'Two-level linear programming', *Management Science*, vol. 30, pp. 1004–1020.

Bjondal, M. & Jornsten, K. 2005, 'The deregulated electricity market biewed as a bilevel programming problem', *Journal of Global Optimization*, vol. 33, pp. 465–475.

Calamai, P. & Vicente, L. 1994, 'Generating quadratic bilevel programming problems', *ACM Transactions on Mathematical Software*, vol. 20, pp. 103–119.

Candler, W. & Townsley, R. 1982, 'A linear two-level programming problem', *Computers and Operations Research*, vol. 9, pp. 59–76.

Charnes, A. & Cooper, W. 1961a, *Management Models and Industrial Application of Linear Programming*, Wiley, New Yourk.

Charnes, A. & Cooper, W. W. 1961b, *Management Models and Industrial Applications of Linear Programming*, vol. I, Wiley.

Charnes, A. & Cooper, W. W. 1977, 'Goal programming and multiple objective optimizations', *European Journal of Operational Research*, vol. 1, pp. 39–54.

Chaudhuri, B. B. & Rosenfeld, A. 1999, 'A modified hausdorff distance between fuzzy sets', *Information Sciences*, vol. 118, no. 1-4, pp. 159–171.

Chen, Y. & Florian, M. 1994, 'The nonlinear bilevel programming problem: formulations, regularity and optimality conditions', *Optimization*, vol. 32, pp. 193–209.

Clegg, J., Smith, M., Xiang, Y. & Yarrow, R. 2001, 'Bilevel programming applied to optimising urban transportation', *Transportation Research Part B: Methodological*, vol. 35, no. 1, pp. 41–70.

Craven, B. D. & Luu, D. V. 1997, 'Optimaization with set functions described by functions', *Optimaization*, vol. 42, pp. 39–50.

David, A. K. & Wen, F. 2000, 'Strategic bidding in competitive electricity markets: a literature survey', *Power Engineering Society Summer Meeting, IEEE*, vol. 4, pp. 2168–2173.

Dempe, S. 2001, *Encyclopedia os optimition*, Kluwer academic.

Diamond, P. & Kloeden, P. 1994, *Metric Spaces of Fuzzy Sets Theory and Applications*, Word Scientific Publishing, Singapore.

Eberhart, R. & Kennedy, J. 1995, 'A new optimizer using particle swarm theory', in *Proceedings Sixth International Symposium on Micro Machine and Human Science*.

Eberhart, R., Simpson, P. & Dobbins, R. 1996, *Computational Intelligence PC Tools*, Academic Press.

Ehrenmann, A. 2004, 'Manifolds of multi-leader cournot equilibria', *Operations research letters*, vol. 32, no. 2, pp. 121–125.

Ehrgott, M. & Gandibleux, X. 2003, 'multiple objective combinatorial optimization-a tutorial', *Multi-objective programming and goal-programming: theory and application*, pp. 3–18.

Fampa, M., Barroso, L., Candal, D. & Simonetti, L. 2008, 'Bilevel optimization applied to strategic pricing in competitive electricity markets', *Computational Optimization and Applications*, vol. 39, pp. 121–142.

Fang, L., Hipel, K. W., Kilgour, D. M. & Peng, X. 2003, 'A decision support system for interactive decision making - part i: Model furmulation', *IEEE transactions on systems, man, and cybernetics - Part C: application and reviews*, vol. 33, no. 1, pp. 42–55.

Feng, C. & Wen, C. 2005, ‘Bi-level and multi-objective model to control traffic flow into the disaster area post earthquake’, *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 4253 – 4268.

Ferris, M. C. & Tin-Loi, F. 2001, ‘Limit analysis of frictional block assemblies as a mathematical program with complementarity constraints’, *International Journal of Mechanical Sciences*, vol. 43, pp. 209–224.

Furth, D. 1979, ‘The stability of generalized stackelberg equilibria in heterogeneous oligopoly’, *J. Econ.*, vol. 39, pp. 315–332.

Gallo, G. & Ulkucu, A. 1977, ‘Bilinear programming: an exact algorithm’, *Mathematical programming*, vol. 12, pp. 173–194.

Gandibleux, X., Mezdaoui, N. & Freville, A. 1997, ‘A tabu search procedure to solve multiobjective combinatorial optimization problems’, in R. Caballero, F. Ruiz & e. R. Steuer (eds), *Advances in Multiple Objective and Goal Programming*, vol. 455, Springer, Berlin, vol. 455, pp. 291–300.

Gao, Y., Zhang, G., Lu, J. & Gao, S. 2007, ‘A bilevel model for railway train set organizing optimization’, in *2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE2007)*, Atlantis Press, pp. 777–782.

Ginzberg, M. J. & Stohr, E. A. 1982, ‘Decision support systems: issues and perspectives’, *Decision Support Systems*.

Guerrero, J. M., Hang, L. & Uceda, J. 2008, ‘Control of distributed uninterruptible power supply systems’, *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2845–2859.

Haghighat, H., Seifi, H. & Ashkan, R. K. 2007, ‘Gaming analysis in joint energy and spinning reserve markets’, *IEEE Transactions on Power Systems*, vol. 22, pp. 2074–2085.

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

Ho1, S. L., Yang, S., Ni, G., Lo, E. W. C. & Wong, H. C. 2005, ‘A particle swarm optimization-based method for multiobjective design optimizations’, *IEEE Transactions On Magnetics*, vol. 41, no. 5, pp. 557–581.

Hobbs, B. F., Metzler, B. & Pang, J. S. 2000, 'Strategic gaming analysis for electric power system: an mpec approach', *IEEE Transactions on Power System*, vol. 15, pp. 637–645.

Hu, C., Teng, C. & Li, S. 2007, 'A fuzzy goal programming approach to multi-objective optimization problem with priorities', *European Journal of Operational Research*, vol. 176, no. 3, pp. 1319–1333.

Huang, S. J. & Pai, F. S. 2002, 'Design and operation of burn-in test system for three-phase uninterruptible power supplies', *IEEE Transactions on Industrial Electronics*, vol. 49, pp. 256–263.

Hwang, C. L. & Yoon, K. 1981, *Multiple attribute decision making - a state of the art survey*, Springer, Berlin.

Hwang, Y. W. 2002, 'Optimality conditions for vector with setvalued maps', *Tech. rep.*, The Chinese university of Hong Kong, Hong Kong.

Ignizio, J. P. 1976, *Goal Programming and Extensions*, Lexington Books, Massachusetts.

Ignizio, J. P. 1983, 'Generalized goal programming: an overview', *Computer & Operations Research*, vol. 10, no. 277-289.

J., B. & J., M. 1973, 'Mathematical programs with optimization problems in the constraints', *Operations Research*, vol. 21, pp. 37–44.

Jeroslow, R. G. 1985, 'The polynomial hierarchy and a simple model for competitive analysis', *Mathematical programming*, vol. 32, pp. 146–164.

Ji, X. & Shao, Z. 2006, 'Model and algorithm for bilevel newsboy problem with fuzzy demands and discounts', *Applied Mathematics and Computation*, vol. 172, no. 1, pp. 163–174.

Kacprzyk, J. 1997, *Multistage Fuzzy Control*, Wiley, Chichester.

Kennedy, J. & Eberhart, R. C. 1995, 'Particle swarm optimization', in *Proc IEEE International Conference on Neural Networks*, pp. 1942–1948.

Kitayama, S. & Yasuda, K. 2006, 'A method for mixed integer programming problems by particle swarm optimization', *Electrical Engineering in Japan*, vol. 157, no. 2, pp. 40–49.

Kuhn, H. W. & Tucker, A. W. 1951, ‘Nonlinear programming’, in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492.

Lai, Y. 1996, ‘Hierarchical optimization: a satisfactory solution’, *Fuzzy Sets and Systems*, vol. 77, pp. 321–335.

Lee, S. M. 1972, *Goal Programming for Decision Analysis*, Auerbach Publishers.

Li, S., Wu, H. & Yang, Y. 2004, ‘Receding horizon fuzzy optimization under local information environment with a case study’, *International Journal of Information Technology & Decision Making*, vol. 3, no. 1, pp. 109–127.

Li, T. & Shahidehpour, M. 2007, ‘Strategic bidding of transmission-constrained gencos with incomplete information’, *IEEE Transactions on Power Systems*, vol. 4, pp. 437–447.

Li, W. & Du, W. 2002, *Optimization Models and Algorithms for the Daily Working Plan of Railway Technical Station*, Southwest Jiaotong University Publisher, Chengdu.

Li, X., Tian, P. & Min, X. 2006, ‘A hierarchical particle swarm optimization for solving bilevel programming problems’, in *Artificial Intelligence and Soft Computing - ICAISC 2006. 8th International Conference Proceedings*, Springer-Verlag, pp. 1169–1178.

Lin, B. & Zhu, S. 1996, ‘Synthetic 0ptimization of train r0uting and makeup plan in a railway network’, *Journal of the China Railway Society*, vol. 18, no. 1, pp. 1–7.

Liu, G. & Han, J. 1997, ‘Optimality conditions for nonconvex bilevel programming problems’, *Systems science and mathematical sciences*, vol. 10, pp. 183–192.

Lu, J., Shi, C. & Zhang, G. 2006, ‘On bilevel multi-follower decision making: General framework and solutions’, *Information Sciences*, vol. 176, no. 11, pp. 1607–1627.

Lu, J., Shi, C., Zhang, G. & Dillon, T. 2007a, ‘Model and extended kuhn-tucker approach for bilevel multi-follower decision making in a referential-uncooperative situation’, *International Journal of Global Optimization*, vol. 38, no. 4, pp. 597–608.

Lu, J., Shi, C., Zhang, G. & Ruan, D. 2007b, 'An extended branch and bound algorithm for bilevel multi-follower decision making in a referential-uncooperative situation', *International Journal of Information Technology and Decision Making*, vol. 6, no. 2, pp. 371–388.

Lu, J., Wu, F. & Zhang, G. 2007c, 'On a generalized fuzzy goal optimization for solving fuzzy multi-objective linear programming problems', *Journal of Intelligent and Fuzzy Systems*, vol. 18, no. 1, pp. 83–97.

Lu, J., Wu, F. & Zhang, G. 2007d, 'On generalized fuzzy goal optimization for solving fuzzy multi-objective linear programming problems', *Journal of Intelligent and Fuzzy Systems*, vol. 18, no. 1, pp. 83–97.

Lu, J., Zhang, G., Ruan, D. & Wu, F. 2007e, *Multi-objective group decision making: methods, software and applications with fuzzy set technology*, Imperial College Press, London.

Lukac, Z., Soric, K. & Rosenzweig, V. V. 2006, 'Production planning problem with sequence dependent setups as a bilevel programming problem', *European Journal of Operational Research*.

Luksan, L. & J, V. 2000, 'Test problems for nonsmooth unconstrained and linearly constrained optimization', *Tech. rep.*, Institut of Computer Science, Academy of Sciences of the Czech Republic.

Malhotra, N. & Arora, S. R. 1999, 'Optimality conditions for linear fractional bilevel programs', *Indian jouranl of pure and applied mathematics*, vol. 30, pp. 373–384.

Mathieu, R., Pittard, L. & Anandalingam, G. 1994, 'Genetic algorithm based approach to bi-level linear programming', *Recherche Operationnelle*, vol. 28, no. 1, pp. 1–21.

Munz, U., P.Schumm, A.Wiesebrock & Allgower, F. 2007, 'Motivation and learning progress through educational games', *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 3141–3144.

Murata, T. & Ishibuchi, H. 1995, 'Moga: multi-objective genetic algorithms', in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, Perth, WA, Australia, pp. 289–294.

Mustajoki, J. & Hämäläinen, R. P. 2007, 'Smart-swaps - a decision support system for multicriteria decision analysis with the even swaps method', *Decision support systems*, vol. 44, pp. 313–325.

Nie, P. 2007, 'Discrete time dynamic multi-leader-follower games with stage-depending leaders under feedback information', *Nonlinear analysis: hybrid systems*, vol. 1, no. 4, pp. 548–559.

Okuguchi, K. 1976, *Expectations and stability in oligopoly models*, Springer-Verlag, New York.

Okuguchi, K. 1978, 'The stability of price adjusting oligopoly with conjectural variation', *J. Econ.*, vol. 38, pp. 55–60.

Onaland, H., Darmawan, D. H. & Johnson, S. H. 1995, 'A multilevel analysis of agriculture credit distribution in east java, indonesia', *Computers and Operations Research*, vol. 22, pp. 227–236.

Pang, J. & Fukushima, M. 2005, 'Quasi-variational inequalities, generalized nash equilibria, and multi-leader-follower games', *Computational Management Science*, vol. 2, pp. 21–56.

Parsopoulos, K. E. & Vrahatis, M. N. 2002, 'Recent approaches to global optimization problems through particle swarm optimization', *Natural Computing*, vol. 1, pp. 235–306.

Pei, Z., Tian, S. & Huang, H. 2006, 'A novel method for solving nonlinear bilevel programming based on hybrid particle swarm optimization', in *The 8th International Conference on Signal Processing*, vol. 3, vol. 3.

Pramanik, S. & Roy, T. K. 2007, 'Fuzzy goal programming approach to multilevel programming problems', *European Journal of Operational Research*, vol. 176, no. 2, pp. 1151–1166.

Rudolph, G. 1994, *Parallel Problem Solving from Nature*, Springer, chap. An evolutionary algorithm for integer programming, pp. 139–148.

Saad, O. M. 2005, 'An iterative goal programming approach for solving fuzzy multiobjective integer linear programming problems', *Applied Mathematics and Computation*, vol. 170, no. 1, pp. 216–225.

Sakawa, M. 1993, ‘Fuzzy linear programming’, in *Fuzzy sets and interactive multiobjective optimization*, Plenum Press, New York.

Sakawa, M., Nishizaki, I. & Uemura, Y. 2000, ‘Interactive fuzzy programming for multilevel linear programming problems with fuzzy parameters’, *Fuzzy Sets and Systems*, vol. 109, no. 1, pp. 3–19.

Schaffer, J. D. 1984, *Multiple objective optimization with Vector Evaluated Genetic Algorithms*, Ph.D. thesis, Vanderbilt University, Nashvile.

Sherali, H. D. 1984, ‘A multiple leader stackelberg model and analysis’, *Operations research*, vol. 32, no. 2, pp. 390–404.

Shi, C. 2005, *Linear Bilevel Programming Technology - Models and Algorithms*, Ph.D. thesis, Universtiy of Technology, Sydney, Sydney.

Shi, C., Lu, J. & Zhang, G. 2005a, ‘An extended kth-best approach for linear bilevel programming’, *Applied Mathematics and Computation*, vol. 164, no. 3, pp. 843–855.

Shi, C., Lu, J. & Zhang, G. 2005b, ‘An extended kuhn-tucker approach for linear bilevel programming’, *Applied Mathematics and Computation*, vol. 162, pp. 51–63.

Shi, C., Lu, J., Zhang, G. & Zhou, H. 2006, ‘An extended branch and bound algorithm for linear bilevel programming’, *Applied Mathematics and Computation*, vol. 180, no. 2, pp. 529–537.

Shi, C., Zhang, G. & Lu, J. 2005c, ‘On the definition of linear bilevel programming solution’, *Applied Mathematics and Computation*, vol. 160, no. 1, pp. 169–176.

Shi, F., Fang, Q., Li, X., Mo, H. & Huang, Y. 2003, ‘A bi-level programming optimization method for the layout of technical service station’, *Journal of the China Railway Society*, , no. 2, pp. 1–4.

Shi, X. & Xia, H. 1997, ‘Interactive bilevel multi-objective decision making’, *Journal of the operational research society*, vol. 48, pp. 943–949.

Shih, H. S., Lai, Y. J. & Lee, E. S. 1996, ‘Fuzzy approach for multi-level programming problems’, *Computers & Operations Research*, vol. 23, pp. 73–91.

Stackelberg, H. V. 1952, *Theory of the Market Economy*, Oxford University Press, New York.

Unlu, G. 1987, 'A linear bilevel programming algorithm based on bicriteria programming', *Computers & Operations Research*, vol. 14, pp. 173–179.

Vicente, L., Savard, G. & Judice, J. 1994, 'Desent approaches for quadratic bilevel programming', *Journal of optimization theory and applications*, vol. 81, pp. 379–399.

W., C. & R., N. 1977, 'Multilevel programming', *Tech. rep.*, World Bank Development Research Center, Washington D.C.

Wang, C. 2004, 'Research on the model and algorithm of dynamic wagon-flow allocating in a marshalling station', *Journal of the China Railway Society*, vol. 26, no. 1, pp. 1–6.

Weber, J. D. & Overbye, T. J. 2002, 'An individual welfare maximization algorithm for electricity markets', *IEEE Transactions on Power Systems*, vol. 17, pp. 590–596.

Wen, F. & Kumar, A. D. 2001, 'Optimal bidding strategies and modeling of imperfect information among competitive generators', *IEEE Transactions on Power Systems*, vol. 16, pp. 15–21.

Wen, U. P. & Hsu, S. T. 1991, 'Linear bilevel programming problems-a review', *Journal of Operational Research Society*, vol. 42, pp. 123–133.

White, D. & Anandalingam, G. 1993, 'A penalty function approach for solving bi-level linear programs', *Journal of Global Optimization*, vol. 3, no. 4, pp. 397–419.

Wu, F., Lu, J. & Zhang, G. 2005, 'Development and implementation on a fuzzy multiple objective decision support system', in *Knowledge-Based Intelligent Information and Engineering Systems. 9th International Conference, KES 2005*, Springer-Verlag, Melbourne, Vic., Australia, pp. 261–267.

Yang, P. C., Wee, H. M. & Yu, J. 2007, 'Collaborative pricing and replenishment policy for hi-tech industry', *Journal of the Operational Research Society*, vol. 58, pp. 894–909.

Ye, J. J. 1997, 'Optimal strategies for bilevel dynamic problems', *SIAM journal on control and optimization*, vol. 35, pp. 512–531.

Yu, H., Dang, C. & Wang, S. 2006, 'Game theoretical analysis of buy-it-now price auctions', *International Journal of Information Technology and Decision Making*, vol. 5, no. 3, pp. 557–581.

Yu, J. & Wang, H. 2007, 'An existence theorem for equilibrium points for multi-leader-follower games', *Nonlinear analysis*.

Zadeh, L. A. 1963a, 'Optimality and non-scalar valued performance criteria', *IEEE Transactions on Automatic Control*, vol. AC-8, pp. 59–60.

Zadeh, L. A. 1963b, 'Optimality and non-sclar valued performance criteria', *IEEE Transactions on Automatic Control*, vol. AC-8, pp. 59–60.

Zadeh, L. A. 1965, 'Fuzzy sets', *Information & Control*, vol. 8, pp. 338–353.

Zhang, G. & Lu, J. 2007a, 'Model and approach of fuzzy bilevel decision making for logistics planning problem', *Journal of Enterprise Information Management*, vol. 20, no. 2, pp. 178–197.

Zhang, G. & Lu, J. 2007b, 'Models and algorithms for fuzzy multi-objective multi-follower linear bilevel programming in a partial cooperative situation', in *2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE2007)*, Chengdu.

Zhang, G., Lu, J. & Dillon, T. 2006a, 'An approximation branch-and-bound algorithm for fuzzy bilevel decision making problems', in *Proceedings of The 1st International Symposium Advances in Artificial Intelligence and Applications*, Poland.

Zhang, G., Lu, J. & Dillon, T. 2006b, 'Kth-best algorithm for fuzzy bilevel programming', in *Proceedings of International Conference on Intelligent Systems and Knowledge Engineering*, Shanghai.

Zhang, G., Lu, J. & Dillon, T. 2007a, 'An approximation kuhn-tucker approach for fuzzy linear bilevel decision making problems', in L. Jain & G. Wren (eds), *Intelligent Decision Making*, Springer.

Zhang, G., Lu, J. & Dillon, T. 2007b, 'Decentralized multi-objective bilevel decision making with fuzzy demands', *Knowledge-Based System*, vol. 20, no. 5, pp. 495–507.

Zhang, G., Lu, J. & Dillon, T. 2007c, 'Fuzzy linear bilevel optimization: Solution concepts, approaches and applications', in Wang, Ruan & Kerre (eds), *Fuzzy Logic - A Spectrum of Theoretical & Practical Issues*, Springer, pp. 351–379.

Zhang, G., Lu, J. & Dillon, T. 2007d, 'Solution concepts and an approximation kuhu-tucker approach for fuzzy multi-objective linear programming', in P. Pardalos & A. Chinchuluun (eds), *Pareto Optimality, Game Theory and Equilibria*, Springer, pp. 467–490.

Zhang, G., Wu, Y., Remias, M. & Lu, J. 2003, 'Formulation of fuzzy linear programming problems as four-objective constrained optimization problems', *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 383–399.

Zhang, J.-R., Zhang, J., Lok, T.-M. & Lyu, M. R. 2007e, 'A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training', *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026–1037.

Zhao, Z. & Gu, X. 2006, 'Particle swarm optimization based algorithm for bilevel programming problems', in *2006 6th International Conference on Intelligent Systems Design and Applications*.

Zimmermann, H. J. 1978, 'Fuzzy programming and linear programming with several objective functions', *Fuzzy Sets and Systems*, vol. 1, pp. 45–55.

Index

List of Symbols and Abbreviations

R	the set of all real numbers
R^n	$n-$dimensional Euclidean space
$F(R)$	the set of all finite fuzzy numbers on R
$F^n(R)$	the set of all $n-$dimensional finite fuzzy numbers on R^n
$M(F(R))_{n\times m}$	the set of $n \times m$ fuzzy matrix on R
LB	linear bi-level
FLB	fuzzy linear bi-level
FMOLB	fuzzy multi-objective linear bi-level
MOB	multi-objective bi-level
MOLB	multi-objective linear bi-level
FMMLB	fuzzy multi-follower multi-objective linear bi-level
FLBG	fuzzy linear bi-level goal
MLOFB	multi-leader one-follower bi-level
OLMFB	one-leader multi-follower bi-level
MLMFB	multi-leader multi-follower bi-level
FMOLBG	fuzzy multi-objective linear bi-level bi-level goal
DSS	decision support system
FBDSS	fuzzy bi-level decision support system
PSO	particle swarm optimisation
MLB	multi-leader bi-level
MFB	multi-follower bi-level
MLBG	multi-leader-one-follower bi-level game
TSO	train set organisation
RWFM	railway wagon flow management
VMI	vendor management inventory
UMP	uniform marginal price