Pergamon

0305-0548(95)00007-0

Computers Ops Res. Vol. 23, No. 1, pp. 73–91, 1996

0305-0548/96 $9.50 + 0.00

FUZZY APPROACH FOR MULTI-LEVEL PROGRAMMING PROBLEMS

Hsu-Shih Shih,† Young-Jou Lai‡ and E. Stanley Lee§
Department of Industrial Engineering, Kansas State University, Manhattan, KS 66502, U.S.A.

(Received July 1994; in revised form December 1994)

Scope and Purpose—Multi-level programming techniques are developed to solve decentralized planning problems with multiple decision makers in a hierarchical organization. There are common features of multi-level organization: interactive decision-making units exist within a predominantly hierarchical structure; execution of decisions is sequential, from top to bottom levels; each unit independently maximizes its own net benefits, but is affected by the actions of other units through externalities; the external effect on a decision maker's problem can be reflected in both objective function and the set of feasible decisions. The basic concepts of multi-level programming techniques are as follows: an upper-level decision maker sets his or her goal and/or decisions and then asks each subordinate level of the organization for their optima which are calculated in isolation; the lower-level decision makers' decisions are then submitted and modified by the upper-level decision maker with consideration of the overall benefit for the organization; and the process is continued until a satisfactory solution is reached. This decision-making process is extremely practical to such decentralized systems as agriculture, government policy, economic systems, finance, warfare, transportation, network designs and is especially suitable for conflict resolution. The purpose of this study is to propose a new concept and develop an efficient methodology to solve a general multi-level programming problem.

Abstract—Multi-level programming techniques are developed to solve decentralized planning problems with multiple decision makers in a hierarchical organization. These become more important for contemporary decentralized organizations where each unit or department seeks its own interests. Traditional approaches include vertex enumeration and transformation approaches. The former is in search of a compromise vertex based on adjusting the control variable(s) of the higher level and thus is rather inefficient. The latter transfers the lower-level programming problem to be the constraints of the higher level by its Kuhn–Tucker conditions or penalty function; the corresponding auxiliary problem becomes non-linear and the decision information is also implicit. In this study, we use the concepts of tolerance membership functions and multiple objective optimization to develop a fuzzy approach for solving the above problems. The upper-level decision maker defines his or her objective and decisions with possible tolerances which are described by membership functions of fuzzy set theory. This information then constrains the lower-level decision maker's feasible space. A solution search relies on the change of membership functions instead of vertex enumeration and no higher order constraints are generated. Thus, the proposed approach will not increase the complexities of original problems and will usually solve a multi-level programming problem in a single iteration. To demonstrate our concept, we have solved numerical examples and compared their solutions with classical solutions.

1. INTRODUCTION

Multi-level mathematical programming (MLP) is identified as mathematical programming that solves decentralized planning problems with multiple executors in a multi-level or hierarchical

†H.-S. Shih is a Ph.D. candidate of industrial engineering at Kansas State University. His research interests are in optimization techniques and their applications in decision making. Mr Shih received his B.S. and M.S. from Chung-Cheng Institute of Technology (Taiwan).

‡Y.-J. Lai is an assistant professor of industrial engineering at Kansas State University. His major research interests are in systems engineering, decision analysis, fuzzy systems and optimization techniques. He received a B.S. from National Cheng-Kung University (Taiwan), and M.S. and Ph.D. degrees from Kansas State University. In addition to a dozen journal articles, he is a co-author of the books *Fuzzy Mathematical Programming* and *Fuzzy Multiple Objective Decision Making*.

§E. S. Lee is a professor of industrial engineering at Kansas State University. His major research interests are in optimization theory, applied mathematics, quasilinearization and invariant imbedding, systems engineering, set theory and expert systems. He received a B.S. from Chung-Cheng Institute of Technology (Taiwan), a M.S. from University of North Carolina and a Ph.D. from Princeton University. In addition to about 100 articles, he is an author, co-author and co-editor of eight books.

organization. Multi-level organization has the following common features: (a) interactive decision-making units exist within a predominantly hierarchical structure; (b) execution of decisions is sequential, from the top to a lower-level; (c) each unit independently maximizes its own net benefits, but is affected by the actions of other units through externalities; and (d) the external effect on a decision maker's (DM's) problem can be reflected in both the objective function and the set of feasible decision space. The basic concept of the MLP technique is that an upper-level DM sets his or her goal and/or decisions and then asks each subordinate level of the organization for their optima which are calculated in isolation; the lower-level DM's decisions are then submitted and modified by the upper-level DM with consideration of the overall benefit for the organization; and the process is continued until a satisfactory solution is reached [1, 2]. This decision-making process is extremely practical to such decentralized systems as agriculture, government policy, economic systems, finance, warfare, transportation, network designs, and is especially suitable for conflict resolution.

During the last three decades, many methodologies have been proposed to solve MLP problems. Most of these methods are based on concepts of vertex enumeration and transformation approaches. The former is to seek a compromise vertex by simplex algorithm based on adjusting higher level control variables. It is rather inefficient, especially for large size problems. Although there is a short-cut, generality will be lost. The latter involves transforming the lower-level programming problem to be the constraints of the higher level by its Kuhn–Tucker (K–T) conditions or penalty function. Because non-linear or Lagrangian terms appear in constraints, the auxiliary problems become complex and sometimes unmanageable.

Presently, solvable problems for MLP are bi-level programming problems (BLPPs), bi-level decentralized programming problems (BLDPPs) and three-level programming problems (TLPPs). Even though these problems are as simple as linear BLPPs, they are categorized as non-convex programming and proven to be NP-hard by Ben-Ayed and Blair [3]. Furthermore, no general hypothesis on cost functions which will guarantee a Pareto optimal or efficient solution for a linear BLPP is obtainable, unless both objectives coincide, in which case, both agents completely cooperate and lead to a Pareto optimal solution of BLPP.

While most existing methods are computationally inefficient, we use the concepts of membership functions as well as multiple objective optimization to develop a fuzzy approach for solving the above problems. We extend Lai's satisfactory solutions [4] and propose that the upper-level DM defines his or her objective and decisions with possible tolerances which are described by membership functions of fuzzy set theory and fuzzy decision. This information constrains the lower-level DM's feasible space. A solution search relies on changes of membership functions expressing satisfactory degrees of potential solutions for both upper and lower level decision making, instead of vertex enumeration and no higher order constraints are generated. Unlike vertex enumeration, we do not pre-assume that the optimal solution exists at corner points. On the other hand, we consider a satisfactory concept is more acceptable than optimality because it is difficult to define a solid optimality in a multi-person, decision-making process and it is questionable by definition to restrict the potential solutions at corners. Potential satisfactory solutions are those in the non-dominated region. Thus, the proposed approach is very efficient and will not increase the complexities of the original problems. As Bard's grid search algorithm [5], and Wen and Hsu's bicriteria algorithm [6] and two-phase approach [7], multiple objective optimization concepts are then used to solve our auxiliary MLPP.

In the following pages, we first provide a historical review of traditional approaches. In Section 3, we further discuss the *K*th-best approach and propose our fuzzy concept and approach for solving BLPP. By extending the proposed concept and methodology of BLPP, BLDPP, MLPP and MLDPP are solved in Section 4. To demonstrate our concept, numerical examples are solved and their solutions are compared with classical solutions. Finally, Section 5 draws some concluding remarks and proposes future studies.

2. LITERATURE REVIEW

BLPP is usually thought of as two DMs in two different hierarchical levels and is similar to static Stackelberg games, a special case of two-person, non-zero sum, non-cooperative game, and with full information. It is a nested optimization model involving two problems, an upper one and a lower one. To discuss BLPP, let a vector of decision variables $(\mathbf{x}_1, \mathbf{x}_2)$ be partitioned among two planners where the upper-level DM has control over vector $\mathbf{x}_1$ and the lower-level DM has control over vector $\mathbf{x}_2$, and let the performance functions f_1 and f_2 of the two planners be linear and bounded. Then a linear BLPP can be stated as:

$$\max_{\mathbf{x}_1} f_1(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{c}_{11}\mathbf{x}_1 + \mathbf{c}_{12}\mathbf{x}_2, \quad (1)$$

where $\mathbf{x}_2$ solves:

$$\max_{\mathbf{x}_2} f_2(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{c}_{21}\mathbf{x}_1 + \mathbf{c}_{22}\mathbf{x}_2, \quad (2)$$

$$\text{s.t. } (\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{F}_2 = \{(\mathbf{x}_1, \mathbf{x}_2) | A_1\mathbf{x}_1 + A_2\mathbf{x}_2 \leqslant \mathbf{b} \text{ and } \mathbf{x}_1, \mathbf{x}_2 \geqslant \mathbf{0}\}$$

where $\mathbf{c}_{11}$, $\mathbf{c}_{12}$, $\mathbf{c}_{21}$, $\mathbf{c}_{22}$ and $\mathbf{b}$ are constant vectors, and A_1 and A_2 are constant matrices. For a given $\mathbf{x}_1$, let $\mathbf{G}(\mathbf{x}_1)$ denote the set of optimal solutions to the following lower-level problem:

$$\max_{\mathbf{x}_2 \in \mathbf{F}_1(\mathbf{x}_1)} f_2^*(\mathbf{x}_2) = \mathbf{c}_{22}\mathbf{x}_2, \quad (3)$$

where $\mathbf{F}_1(\mathbf{x}_1) = \{\mathbf{x}_2 | A_2\mathbf{x}_2 \leqslant \mathbf{b} - A_1\mathbf{x}_1\}$ represents the upper-level DM's feasible decision space. The set of rational reactions of f_2 over $\mathbf{F}_2$ can also be defined as:

$$\mathbf{S}_{f2}(\mathbf{F}_2) = \{(\mathbf{x}_1, \mathbf{x}_2) | (\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{F}_2 \text{ and } \mathbf{x}_1 \in \mathbf{G}(\mathbf{x}_1)\}.$$

It can be shown that multiple local optima can exist and an implicit search can be defined [8]. From a mathematical viewpoint, this problem is that of maximizing a piecewise linear function over a polyhedron [9]. The algorithms involving BLPP are developed earliest among the MLPPs. Their procedures are thus well-structured and widely diversified. In general, they can be grouped as vertex enumeration and transformation approaches.

Bialas and Karwan, pioneers for BLPP, present a vertex enumeration approach, *K*th-best method in 1978 and restated it in Refs [10, 11]. Since the decision variable space for two levels is the same (besides one additional equation for control variable from the upper-level), an optimal solution will thus occur at an extreme point of $\mathbf{F}_2$. The extreme point search is the basis for the *K*th-best algorithm which will be solved by a simplex method. Although it is developed to find a global optimal solution, computational and storage requirements of the algorithm increase dramatically with the number of variables. It will result in more serious problems when applied to a general *k*-level problem. Candler and Townsley [12] propose an implicit search algorithm that focuses on generating an enumerating bases from lower-level activities, but no progress has been made for a large system.

Brad's grid search algorithm (GSA) begins by setting up a parameterized linear program whose objective function for the corresponding problem is formulated as a convex combination of the objective functions from each level. He demonstrates that GSA generally outperforms the other four methods. Nevertheless, Haurie *et al.* [13] provide a counter-example for proving that the algorithm doesn't always obtained the design solution, and Ben-Ayed and Blair show that the solution given may even be local. Concurrent to GSA, Bard [14] further suggests a set of necessary conditions for BLPPs and general BLPPs which result in a potentially equivalent mathematical program formulation for the bi-criteria case. A relationship between the solution to a BLPP and Pareto optimality is also discussed. On the other hand, Clark and Westerberg [15] demonstrate with a counter-example that Bard's necessary conditions are incorrect.

Based on Bard's algorithm, Ünlü [16] proposes a technique of bi-criteria programming (BCP) to solve BLPP. The techniques for BCP require less computational effort than those for bi-level programming. However, Candler [17] shows one theorem in Refs [5, 14], used by Ünlü, to be erroneous. Wen and Hsu [6] also propose a bicriteria algorithm and point out that the BCP algorithm is not suitable for all BLPP, and Marcotte and Savard [18] show that it is possible to design a linear bi-level problem whose optimal solution is not Pareto optimal.

When the optimal solution is not Pareto-optimal, Wen and Hsu [7] use multiple objective solution techniques to obtain an efficient solution in the proposed feasible contraction set and suggest three solutions: the threat-point, ideal-point and ideal-threat-point dependent solutions.

Transformation approaches are usually based on replacing the two-level problem with its K–T conditions. In this way, BLPP can be formulated as the following first-level auxiliary problem:

$$\max_{\mathbf{x}_1,\mathbf{x}_2} f_1(\mathbf{x}_1,\mathbf{x}_2) = \mathbf{c}_{11}\mathbf{x}_1 + \mathbf{c}_{12}\mathbf{x}_2, \tag{4}$$

$$\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \leqslant \mathbf{b},$$

$$\mathbf{w}(\mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 - \mathbf{b}) = \mathbf{0},$$

$$\mathbf{w}\mathrm{A}_2 = \mathbf{c}_{22},$$

$$\mathbf{x}_1, \mathbf{x}_2, \mathbf{w} \geqslant \mathbf{0}.$$

where the second set of constraints are complementary slackness conditions (CSC) which make (4) a non-convex programming problem. To solve this nonlinear programming problem, Bialas and Karwan [11] propose a parametric complementary pivot (PCP) algorithm which iteratively solves a slight perturbation of the system. It can be viewed as an implicit enumeration of the lower-level optimal bases, and can be extended to solve TLPP. Jùdice and Faustino in Ref. [9] modify the PCP algorithm to guarantee convergence towards a global optimum; however, it can be low efficiency because of the branch-and-bound basis. Ben-Ayed and Blair show that the PCP method may not converge to optimality.

Fortuny-Amat and McCarl's [19] approach enforces the CSC by transforming the formulation into a much larger mixed integer programming problem. Bard and Falk [20] propose a series of transformations for CSC to transform product terms into a series of equalities without altering the solution, and then use the branch-and-bound technique to partition the feasible region and to obtain a global optimal solution. Bard and Moore [21] modify the branch-and-bound algorithm with a 0–1 variable and extend it to solve quadratic BLPP, and claim that the algorithm's performance and robustness is superior to all contenders. However, Ben-Ayed states that for the larger problem the efficiency of the algorithm is still constrained by the exponential growth of the branch-and-bound tree. Wen and Yang [22] suggest a heuristic procedure for solving the mixed integer linear BLPP.

Recently, Anandalingam and White [23] used penalty function to develop a new approach based on a duality gap for solving a nonlinear convex objective function. Although their approach can be proved to be more efficient than approaches based on vertex enumeration and K–T conditions, and can provide a global optimal solution, the method is illustrated by a small example only and no computational result for large-scale problems is reported. The relationship between each level is rather implicit and may be not correctly represented.

3. FUZZY APPROACH FOR BLPP

To discuss the multi-level programming problem, it is better to start with BLPP for its simplicity. Characteristics of BLPP include the following: (a) interaction: the sequence for choosing strategy is top-down, but an upper-level DM will accept reactions from the lower-level; (b) non-cooperation: each level DM will seek his or her own interests, i.e. optimize their individual objectives, and no one dominates the entire problem; (c) non-zero sum: the loss for the cost of one level is unequal to the gain for the cost of the other level; and (d) full information: each level DM is fully informed about all prior choices when it is his or her turn to move. To meet these strategies and reactions, Bialas and Karwan propose the following *K*th-best algorithm: the solution search starts at the individual optimum of the upper-level DM, and the optimal solution is reached if it matches the lower-level DM's optimality; otherwise, search for the neighboring corner (extreme) points of the previous point until the upper-level DM's proposed decision matches the lower-level DM's optimality. Through this algorithm, we can see how the upper-level DM decreases his or her objective value in order to make a compromise for the lower-level DM's optimality. In fact, an implicit compromise process has been carried out through the solution search. Let us consider an extreme situation where the independent solutions for two DMs are located at two neighboring vertices. The *K*th-best algorithm will force its solution to be either of them, depending on who goes first; i.e. the DM who moves first absolutely dominates the solution. Since the conflict has not been solved yet, the

Kth-best's solution seems less meaningful. A compromise solution between these two extreme points should be more practical. Indeed, this phenomenon happens in all problems. The fundamental of the Kth-best algorithm is that the optimum should exist among corner points and that the corner point search leads to complicated enumeration. As mentioned above (see Refs [24, 25] also), it is difficult to define a solid optimality for multi-person, decision-making problems. Compromise or coordination are usually needed in order to reach a solution, even in a non-cooperative phenomenon.

For a large scale problem, the Kth-best algorithm reaches the desired solution rather slowly because the simplex algorithm will have to search a huge set of vertices. Second, DMs do not know the relationship between levels and the possible effects of individual actions on each other, i.e. lacking explicit information, especially about goal achievements of each DM. In the third point, the procedure presents that yield always happens with the upper-level DM, while the lower-level DM obtains some profits in taking advantage of the loss of the former, i.e. decreasing sequentially from the optimum of the upper-level if it is not satisfied by the lower-level. It obviously violates natural law, boss first, in a hierarchy. Finally, even in a decentralized organization, a non-dominated solution might be more meaningful than classical solutions. In this case, other non-corner, non-dominated solutions may be good enough and, at the same time, computational difficulty caused by enumeration can be avoided. In many other cases, Kth-best solutions are dominated and thus not very attractive for both DMs, especially in practice.

Similar to the Kth-best algorithm, the transformation approach based on K–T conditions requires the upper-level DM's solution satisfy the lower-level DM's optimality conditions. That is, the lower-level DM's optimality conditions become rigid constraints of the upper-level DM's problem, or he or she dominates the solution search process. Thus, K–T conditions may also not be practical for solving real-world problems.

Instead of searching through vertices as the Kth-best algorithm, we here propose a supervised search procedure (supervised by top-level DM) which will generate a (non-dominated) satisfactory solution for a multi-level programming problem. In this solution search, the upper-level DM specifies preferred values of his or her control variables and goals with some leeway. This information is modeled by membership functions of fuzzy set theory and passed to the lower-level DM as his or her additional constraints or boss's requirements. The lower-level DM should not only optimize his or her objective but also try to satisfy the upper-level DM's goal and preference as much as possible. He or she realizes that without seriously considering the boss's goal and preference, the proposed solution will very possibly be rejected and the solution search will be a lengthy one. The lower-level DM then presents his or her solution to the upper-level DM. If the upper-level DM agrees to the proposed solution, a solution is reached and it is called a satisfactory solution here. If he or she rejects this proposal, the upper-level DM will need to re-evaluate and change former goals and decisions as well as their corresponding leeway or tolerances until a satisfactory solution is reached. Figure 1 depicts the proposed decision-making procedure. This strategy does not violate the non-cooperate nature—both level DMs first seek their optimal solutions in isolation. However, it does need some kind of coordination with the classical approaches.

Mathematically, the upper-level DM first solves the following problem:

$$\max f_1(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{c}_{11}\mathbf{x}_1 + \mathbf{c}_{12}\mathbf{x}_2, \tag{5}$$

$$\text{s.t. } (\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{F}_2 = \{(\mathbf{x}_1, \mathbf{x}_2) | \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \leqslant \mathbf{b}, \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0}\},$$

whose solution is assumed to be $(\mathbf{x}_1^U, \mathbf{x}_2^U, f_1^U)$, and the lower-level DM independently solves:

$$\max f_2(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{c}_{21}\mathbf{x}_1 + \mathbf{c}_{22}\mathbf{x}_2, \tag{6}$$

$$\text{s.t. } (\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{F}_2 = \{(\mathbf{x}_1, \mathbf{x}_2) | \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \geqslant \mathbf{b}, \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0}\},$$

whose solution is assumed to be $(\mathbf{x}_1^L, \mathbf{x}_2^L, f_2^L)$. The above solutions are then disclosed to both DMs. If $(\mathbf{x}_1^U, \mathbf{x}_2^U) = (\mathbf{x}_1^L, \mathbf{x}_2^L)$, an optimal solution is reached. However, two solutions are usually different because of conflicts of nature between two objectives. The upper-level DM understands that using the optimal decision $\mathbf{x}_1^U$ as a control factor for the lower-level DM is obviously not practical. It is

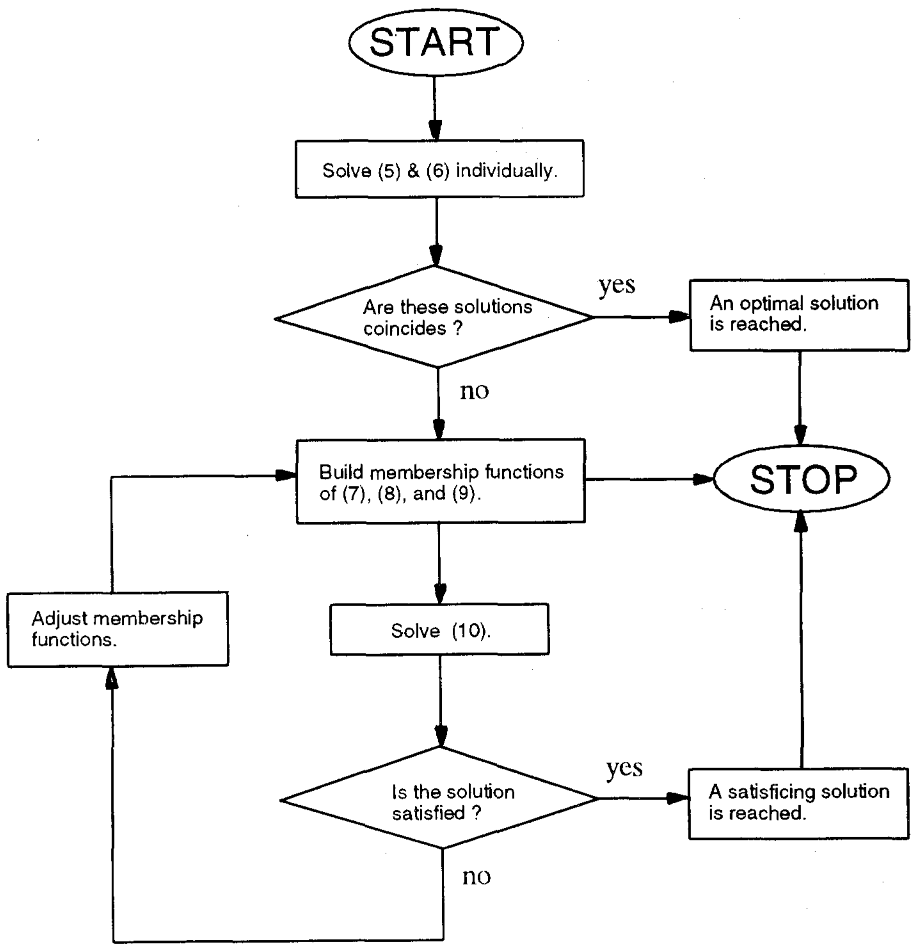

Fig. 1. The flow chart for the proposed fuzzy approach.

more reasonable to have some leeway or tolerances that give the lower-level DM a wider feasible domain to search for his or her optimal solution, and that will significantly reduce searching time or iterations. The range of the decision on $\mathbf{x}_1$ should be "around $\mathbf{x}_1^U$ with its maximum tolerances $\mathbf{p}_1$." The most preferred decision is $\mathbf{x}_1^U$; the worst acceptable decision is at $\mathbf{x}_1^U - \mathbf{p}_1$ and $\mathbf{x}_1^U + \mathbf{p}_1$, and that satisfaction or preference is linearly increasing within the interval of $[\mathbf{x}_1^U - \mathbf{p}_1, \mathbf{x}_1^U]$ and linearly decreasing within $[\mathbf{x}_1^U, \mathbf{x}_1^U + \mathbf{p}_1]$, and other decisions are not acceptable. This information can then be formulated as the following membership functions of fuzzy set theory (see Lai and Hwang [28]):

$$\mu_{\mathbf{x}_1}(\mathbf{x}_1) = \begin{cases} [\mathbf{x}_1 - (\mathbf{x}_1^U - \mathbf{p}_1)]/\mathbf{p}_1, & \text{if } \mathbf{x}_1^U - \mathbf{p}_1 \leqslant \mathbf{x}_1 \leqslant \mathbf{x}_1^U; \\ [(\mathbf{x}_1^U + \mathbf{p}_1) - \mathbf{x}_1)]/\mathbf{p}_1, & \text{if } \mathbf{x}_1^U < \mathbf{x}_1 \leqslant \mathbf{x}_1^U + \mathbf{p}_1; \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

which is also depicted in Fig. 2.

Meanwhile, it is very important that the upper-level DM should specify his goal with his or her tolerance to the lower-level DM in order to direct or supervise him or her to search for solutions in the right direction. The upper-level DM's goal may reasonably consider that all $f_1 \geqslant f_1^U$ are absolutely acceptable and all $f_1 < f'_1[=f_1(\mathbf{x}_1^L, \mathbf{x}_2^L)]$ are absolutely unacceptable, and that the preference within $[f'_1, f_1^U]$ is linearly increasing. The fact that the lower-level DM obtained the optimum at $(\mathbf{x}_1^L, \mathbf{x}_2^L)$, which in turn provides the upper-level DM the objective value of $\mathbf{f}'_1$, makes any $f_1 < f'_1$ unattractive in practice. The following membership function can then be reasonably

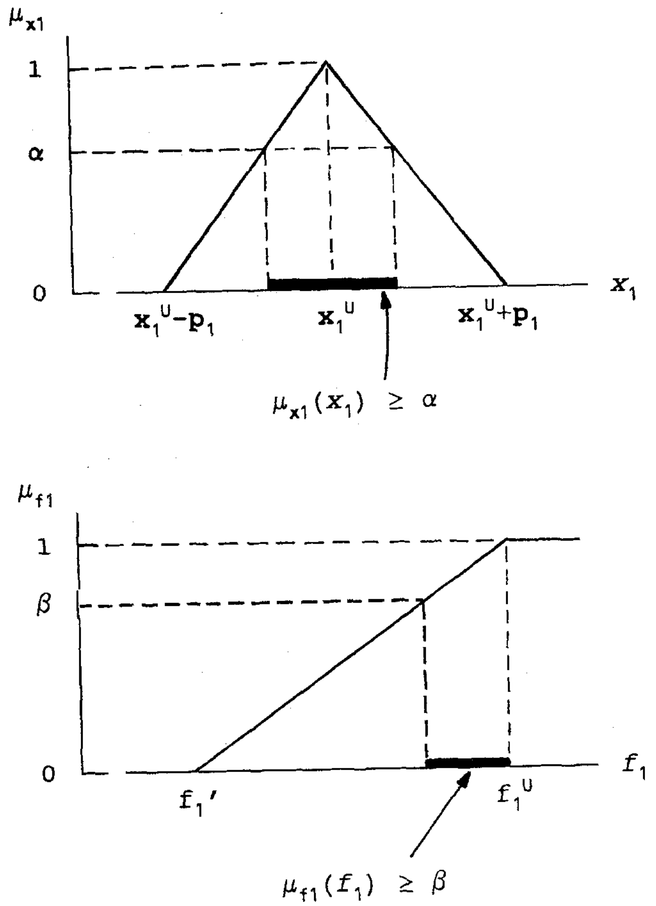

Fig. 2. The membership functions for x_1 and f_1.

assumed:

$$\mu_{f1}[f_1(\mathbf{x})] = \begin{cases} 1, & \text{if } f_1(\mathbf{x}) > f_1^U; \\ [f_1(\mathbf{x}) - f'_1]/[f_1^U - f'_1], & \text{if } f'_1 \leqslant f_1(\mathbf{x}) \leqslant f_1^U; \\ 0, & \text{if } f_1(\mathbf{x}) < f'_1, \end{cases} \tag{8}$$

which is also illustrated in Fig. 2. The lower-level DM now optimizes his objective under the new constraints of "$\mathbf{x}_1$ is about $\mathbf{x}_1^U$" and "f_1 is somehow near to or greater than f_1^U" which are modeled by the membership functions (7) and (8). With (3), (7) and (8), the lower-level DM then obtains the following problem (see Lai and Hwang [26]):

$$\begin{aligned} &\max_{\mathbf{x}_2} f_2(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{c}_{21}\mathbf{x}_1 + \mathbf{c}_{22}\mathbf{x}_2, \\ &\text{s.t. } A_1\mathbf{x}_1 + A_2\mathbf{x}_2 \leqslant \mathbf{b}, \\ &\quad \mathbf{x}_1 = \mathbf{x}_1^U, \\ &\quad f_1(\mathbf{x}) \geqslant f_1^U, \\ &\quad \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0}, \end{aligned}$$

or

$$\begin{aligned} &\max_{\mathbf{x}_2} f_2(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{c}_{21}\mathbf{x}_1 + \mathbf{c}_{22}\mathbf{x}_2, \\ &\quad \text{s.t. } A_1\mathbf{x}_1 + A_2\mathbf{x}_2 \leqslant \mathbf{b}, \\ &\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \alpha\mathbf{I}, \end{aligned}$$

$$\mu_{f_1}[f_1(\mathbf{x})] \geqslant \beta,$$
$$\mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0},$$
$$\alpha \in [\mathbf{0}, \mathbf{1}] \text{ and } \beta \in [0, 1],$$

where α (a row vector) and β are minimum acceptable degrees of satisfaction or preference for the decision $\mathbf{x}_1$ and objective $f_1(\mathbf{x})$, respectively, and $\mathbf{I}$ is a column vector with all elements equal to 1s and the same dimension as $\mu_{\mathbf{x}_1}(\mathbf{x}_1)$ or $\mathbf{x}_1$. The feasible ranges constrained by $\mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \alpha$ and $\mu_{f_1}[f_1(\mathbf{x})] \geqslant \beta$ are depicted in Fig. 2. Obviously, the lower-level DM can analyze various solutions corresponding to the upper-level DM's satisfactory levels α and β.

For each possible solution available to the upper-level DM, the lower-level DM may be willing to build a membership function for his or her objective so that he or she can rate the satisfaction of each potential solution. Here, assume that the lower-level DM has the following membership function for his goal:

$$\mu_{f_2}[f_2(\mathbf{x})] = \begin{cases} 1, & \text{if } f_2(\mathbf{x}) > f_2^{\mathrm{L}}; \\ [f_2(\mathbf{x}) - f_2']/[f_2^{\mathrm{L}} - f_2'], & \text{if } f_2' \leqslant f_2(\mathbf{x}) \leqslant f_2^{\mathrm{L}}; \\ 0, & \text{if } f_2(\mathbf{x}) < f_2'; \end{cases} \tag{9}$$

where $f_2' = f_2(x_1^{\mathrm{T}})$. Obviously, the above membership function μ is a one-to-one mapping within a compact interval of f_2^{L} and f_2'. Because f_2^{L} is the best solution of (6), $f_2(\mathbf{x}) > f_2^{\mathrm{L}}$ is impossible while the upper-level DM gives more constraints to the lower-level DM. The lower-level DM will not accept any $f_2(\mathbf{x}) < f_2'$ for the same reason as the upper-level DM, discussed above. Therefore, the lower-level DM has $\mu_{f_2}[f_2(\mathbf{x})] = [f_2(\mathbf{x}) - f_2']/[f_2^{\mathrm{L}} - f_2']$ and the following auxiliary model:

$$\begin{aligned} &\max \delta = \mu_{f_2}[f_2(\mathbf{x})], \\ &\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \leqslant \mathbf{b}, \\ &\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \alpha\mathbf{I}, \\ &\quad \mu_{f_1}[f_1(\mathbf{x})] \geqslant \beta, \\ &\quad \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0}, \\ &\quad \alpha \in [\mathbf{0}, \mathbf{1}] \text{ and } \beta, \delta \in [0, 1], \end{aligned}$$

or

$$\begin{aligned} &\max \delta, \\ &\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \leqslant \mathbf{b}, \\ &\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \alpha\mathbf{I}, \\ &\quad \mu_{f_1}[f_1(\mathbf{x})] \geqslant \beta, \\ &\quad \mu_{f_2}[f_2(\mathbf{x})] \geqslant \delta, \\ &\quad \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0}, \\ &\quad \alpha \in [\mathbf{0}, \mathbf{1}] \text{ and } \beta, \delta \in [0, 1], \end{aligned}$$

where δ is the satisfactory degree of the lower-level DM who searches for a solution with a higher δ value under the consideration of α and β values. To resolve conflict between both DMs and to avoid the upper-level DM's rejection, the lower-level DM should try to maximize α, β and δ simultaneously, that is:

$$\begin{aligned} &\max\{\delta, \boldsymbol{\alpha}, \beta\}, \\ &\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \leqslant \mathbf{b}, \\ &\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \alpha\mathbf{I}, \\ &\quad \mu_{f_1}[f_1(\mathbf{x})] \geqslant \beta, \end{aligned}$$

$$\mu_{f_2}[f_2(\mathbf{x})] \geqslant \delta,$$
$$\mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0},$$
$$\boldsymbol{\alpha} \in [\mathbf{0}, \mathbf{1}] \text{ and } \beta, \delta \in [0, 1].$$

If the min operator is used to aggregate the satisfactory levels or $\lambda = \min\{\alpha, \beta, \delta\}$, the above problem will become:

$$\begin{aligned}
&\max \lambda,\\
&\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \geqslant \mathbf{b},\\
&\quad \boldsymbol{\mu}_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \lambda\mathbf{I},\\
&\quad \mu_{f_1}[f_1(\mathbf{x})] \geqslant \lambda,\\
&\quad \mu_{f_2}[f_2(\mathbf{x})] \geqslant \lambda,\\
&\quad \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0},\\
&\quad \lambda \in [0, 1],
\end{aligned}$$

or

$$\begin{aligned}
&\max \lambda, \\
&\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_2\mathbf{x}_2 \leqslant \mathbf{b},\\
&\quad [(\mathbf{x}_1^{\mathrm{U}} + \mathbf{p}_1) - \mathbf{x}_1]/\mathbf{p}_1 \geqslant \lambda\mathbf{I},\\
&\quad [\mathbf{x}_1 - (\mathbf{x}_1^{\mathrm{U}} - \mathbf{p}_1)]/\mathbf{p}_1 \geqslant \lambda\mathbf{I},\\
&\quad \mu_{f_1}[f_1(\mathbf{x})] = [f_1(x) - f_1']/[f_1^{\mathrm{T}} - f_1'] \geqslant \lambda,\\
&\quad \mu_{f_2}[f_2(\mathbf{x})] = [f_2(x) - f_2']/[f_2^{\mathrm{L}} - f_2'] \geqslant \lambda,\\
&\quad \mathbf{x}_1 \text{ and } \mathbf{x}_2 \geqslant \mathbf{0},\\
&\quad \lambda \in [0, 1].
\end{aligned} \tag{10}$$

Equation (10) is actually a fuzzy or max–min programming problem by applying Bellman and Zadeh's [27] max–min decision.

If the upper-level DM is satisfied with the solution of (10), a satisfactory solution is reached. Otherwise, he or she should provide new membership functions for the control variable and objective to the lower-level DM until a satisfactory solution is reached. Combined with set of control decisions and goals with tolerance, this solution becomes a satisfactory solution for (1) and (2).

Note: the min operator is noncompensatory and thus may not be good enough to model such situations that trade-offs between δ, α, β are allowable or even unavoidable. In those cases, compensatory aggregation is much more meaningful. Among various operators, the most appropriate compensatory operators for solving MLPP are: algebraic product, algebraic sum, bounded product, bounded sum, Hamacher's min and max, Yager's min and max, Dubois and Prade's min and max, Werners's "fuzzy and" and "fuzzy or" and Zimmermann and Zysno's Γ operator (see Lai and Hwang [26] for details).

As to membership functions, the linear (and triangular) forms are chosen for computational efficiency. Other membership functions such as piecewise, exponential, hyperbolic, inverse hyperbolic or some specific power functions may be needed for pragmatical reasons. Many of these nonlinear functions can be transferred into equivalent linear forms by variable transformations. Therefore, linear forms are only discussed here without losing generality. Indeed, membership functions are essential while applying fuzzy approaches to solve real-world problems. They are generated basically by heuristic determination, reliability concerns, theoretical demand and human perception. It is not the purpose of this study to discuss various function forms and methods to generate membership functions. Concise discussion on these topics has been given by Lai and Hwang [26].

To illustrate this approach, let us consider the following example.

Example 1

One export-oriented country is concentrating on two important products, 1 and 2, which are manufactured by ABC company on given capabilities. Product 1 yields a profit of \$1 per piece and product 2 a profit of \$2 per piece. Product 1 can be exported, yielding a revenue of \$2 per piece from foreign countries, while product 2 needs the imported raw materials of \$1 per piece. There are two level DMs related to this case, i.e. government (upper-level) and the manager of the company (lower-level), and each one can handle one decision variable only, x_1 and x_2, respectively. Two objectives are established respectively: (1) effect on the balance of trade $f_1(\mathbf{x})$, i.e. maximum amount of exports; and (2) profit on the product $f_2(\mathbf{x})$, i.e. maximum profit. The problem can then be formulated as:

$$\max_{x_1} f_1 = 2x_1 - x_2 \text{ (effect on the export trade)},$$

where x_2 solves

$$\begin{aligned} &\max_{x_2} f_2 = x_1 + 2x_2 && \text{(profit on the products)},\\ &\text{s.t. } 3x_1 - 5x_2 \leqslant 15 && \text{(capacity)},\\ &\quad 3x_1 - x_2 \leqslant 21 && \text{(management)},\\ &\quad 3x_1 + x_2 \geqslant 27 && \text{(space)},\\ &\quad 3x_1 + 4x_2 \leqslant 45 && \text{(material)},\\ &\quad x_1 + 3x_2 \leqslant 30 && \text{(labor hours)},\\ &\quad x_1 \text{ and } x_2 \geqslant 0. \end{aligned}$$

whose constraint set is denoted by $\mathbf{X}$. The Kth-best solution is $(x_1, x_2) = (8, 3)$ at $K = 2$. In addition, the optimum for the upper-level objective is $f_1 = 13.5$ at (7.5, 1.5) and for the lower-level objective is $f_2 = 21$ at (3, 9). The decision variable and objective function spaces are shown in Figs 3 and 4, respectively.

The proposed approach first finds individual optimal solutions by solving (5) and (6) and obtains $(\mathbf{x}_1^U, \mathbf{x}_2^U) = (7.5, 1.5)$ and $f_1^U = 13.5$ and $(\mathbf{x}_1^L, \mathbf{x}_2^L) = (3, 9)$ and $f_2^L = 21$. $f_1^T = 13.5$ and let us assume $f'_1 = 0$ (only positive is meaningful here) instead of -3 and $f'_2 = 10.5$. Assume the upper-level DM's control decision x_1 is around 7.5 with the negative and positive-side tolerances 4.5 and 0.5, respectively. By (7), (8) and (9), membership functions $\mu_{x_1}(\cdot)$, $\mu_{f_1}(\cdot)$ and $\mu_{f_2}(\cdot)$ are built. The

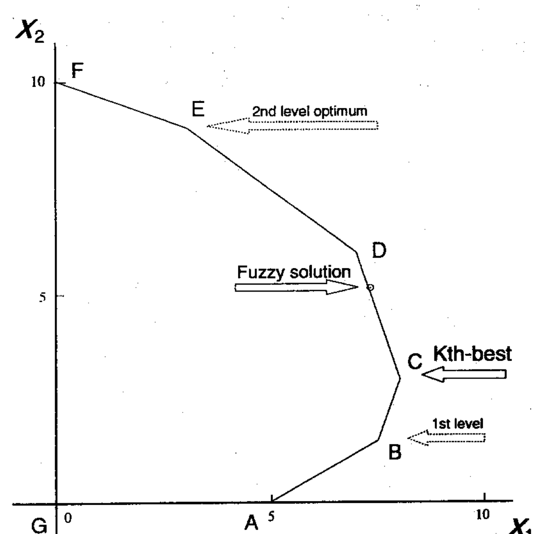

Fig. 3. The decision variable space for Example 1.

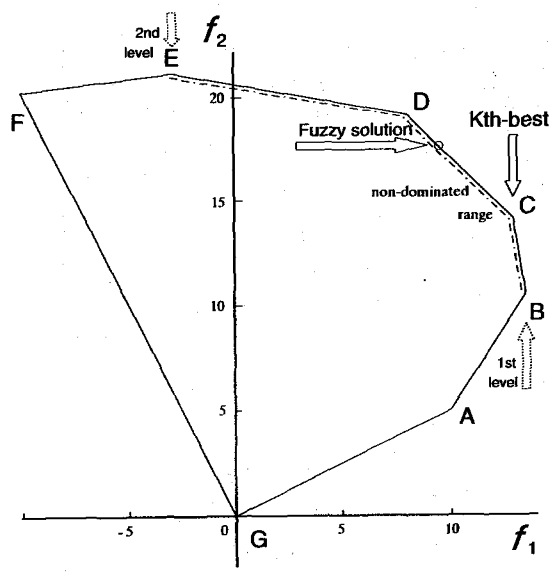

Fig. 4. The objective function space for Example 1.

lower-level DM then solves the following problem of (10):

$$\max \lambda,$$
$$\text{s.t. } \mathbf{x} \in \mathbf{X},$$
$$x_1 \geqslant 4.5\lambda + 3,$$
$$x_1 \leqslant 8 - 0.5\lambda,$$
$$2x_1 - x_2 \geqslant 13.5\lambda,$$
$$x_1 + 2x_2 - 10.5 \geqslant 10.5\lambda,$$
$$\lambda \in [0, 1],$$

whose compromise solution is $f^* = (f_1^*, f_2^*) = (9.29, 17.72)$ at $x^* = (7.26, 5.23)$ with the overall satisfaction of both DMs $\lambda = 0.69$. Realized satisfactory levels are $(\mu_{x_1}^*, \mu_{f_1}^*, \mu_{f_2}^*) = (0.95, 0.69, 0.69)$. If the upper-level DM's total satisfactory level $\lambda_1 = \min\{\mu_{x_1}^*, \mu_{f_1}^*\}$, then our solution provides $\lambda_1 = 0.69$ and λ_2 (of the lower-level DM) $= 0.69$. On the other hand, the Kth-best solution $f = (13, 14)$ at $\mathbf{x} = (8, 3)$ has $(\mu_{x_1}, \mu_{f_1}, \mu_{f_2}) = (0.0, 0.96, 0.33)$ and thus $\lambda_1 = 0.0$ and $\lambda_2 = 0.33$. Obviously, our solution is better than that of the Kth-best in terms of satisfactions of both DMs.

4. EXTENSIONS

The proposed approach can efficiently solve not only BLPP, but also bi-level decentralized programming problems, multi-level programming problems and multi-level decentralized programming problems which will be discussed in Sections 4.1, 4.2 and 4.3.

4.1. BLDPP

A bi-level decentralized programming problem (BLDPP) is characterized by one decision center at the top level and p divisions at the bottom level with the assumption that decision units in the lower level are independent and under control of the higher level. Let us first define the following notations: $f_{ki}(\mathbf{x})$ represents the objective function of the ith division of the kth level; c_{kij} is the cost coefficient of the decision variable x_j for the ith division of the kth-level. According to the definition, in a bi-level decentralized organization, $k = 1$ (the upper-level) and 2 (the lower-level), and $i = 1, 2, \ldots, s_k$. $s_1 = 1$ for the upper level and $s_2 = s$ for s divisions constitute the lower level.

BLDPP can then be defined as:

$$\max_{\mathbf{x}_{11}} f_{11}(\mathbf{x}) = \Sigma_j c_{11j} x_j; \tag{11}$$

where $\mathbf{x}_{21}, \mathbf{x}_{22}, \ldots, \mathbf{x}_{2s}$ solve

$$\begin{aligned} &\max_{\mathbf{x}_{21}} f_{21}(\mathbf{x}) = \Sigma_j c_{21j} x_j; \\ &\ldots \\ &\max_{\mathbf{x}_{2s}} f_{2s}(\mathbf{x}) = \Sigma_j c_{2sj} x_j; \\ &\text{s.t. } \sum_{\forall k, i} \mathbf{A}_{ki}\mathbf{x}_{ki} \leqslant \mathbf{b}, \ k = 1, 2 \text{ and } i = 1, 2, \ldots, s_k, \\ &\quad x_j \geqslant 0, \ j = 1, 2, \ldots, n, \end{aligned}$$

where $j = 1, 2, \ldots, n$ representing the jth decision variable, and the constraint set is denoted by $\mathbf{X}$. In addition, $\cup_{k,i}\{\mathbf{x}_{ki}|\forall k, i\} = \{x_1, x_2, \ldots, x_n\}$ and x_j are decisions of the s divisions in the bottom level.

To solve (11), we can extend the concept and approach discussed in Section 3. We first obtain individual optimal solutions:

$$f^*_{ki} = f_{ki}(\mathbf{x}^*_{ki}) = \max_{x \in X} f_{ki}(\mathbf{x}), \quad \forall k, i. \tag{12}$$

Then, $f'_{ki} = \min_{k,i} f_{ki}(\mathbf{x}^*_{ki})$. And, if the tolerance vector $\mathbf{p}_1$ is given by the upper-level DM, membership functions $\mu_{\mathbf{x}_{11}}(\mathbf{x}_{11})$ and $\mu_{ki}[f_{ki}(\mathbf{x})]$ can then be formulated by using (7), (8) and (9).

The lower-level DMs then have the following problem:

$$\begin{aligned} &\max_{\mathbf{x}_{21}} f_{21}(\mathbf{x}) = \mathbf{c}_{21}\mathbf{x}_{21}; \\ &\ldots \\ &\max_{\mathbf{x}_{2s}} f_{2s}(\mathbf{x}) = \mathbf{c}_{2s}\mathbf{x}_{2s}; \\ &\text{s.t } \mathbf{x} \in \mathbf{X}, \\ &\quad \boldsymbol{\mu}_{\mathbf{x}_{11}}(\mathbf{x}_{11}) \geqslant \boldsymbol{\alpha}, \\ &\quad \mu_{11}[f_{11}(\mathbf{x})] \geqslant \beta, \\ &\quad \alpha \in [\mathbf{0}, \mathbf{1}] \text{ and } \beta \in [0, 1], \end{aligned} \tag{13}$$

which is a multiple objective programming. Note that each division has its own objective and all divisions constitute the lower (or second) level of the organization. Because divisions in the same level should have similar positions in decision processes, multiple objective methodologies (see Lai and Hwang [28]) are then reasonable to model this phenomenon. As in Section 3, we here use max–min programming to solve (13) and obtain the following auxiliary problem:

$$\begin{aligned} &\max \lambda, \\ &\text{s.t. } \mathbf{x} \in \mathbf{X}, \\ &\quad \boldsymbol{\mu}_{\mathbf{x}_{11}}(\mathbf{x}_{11}) \geqslant \lambda \mathbf{I}, \\ &\quad \mu_{11}[f_{11}(\mathbf{x})] \geqslant \lambda, \\ &\quad \mu_{2i}[f_{2i}(\mathbf{x})] \geqslant \lambda, \quad i = 1, \ldots, s, \\ &\quad \lambda \in [0, 1], \end{aligned} \tag{14}$$

whose solution is then a satisfactory solution for the BLDPP problem of (11) if the upper-level DM is satisfied with it. If the upper-level DM is not satisfied with the current solution, he or she will then modify the membership functions until a satisfactory solution is reached.

Example 2

Consider the following BLDPP [29]:

$$\max_{x_1} f_1 = x_1 + y_1 + 2y_2 + y_3,$$

where y_1, y_2 and y_3 solve:

$$\max_{y_1} f_{21} = -x_1 + 3y_1 - y_2 - y_3;$$

$$\max_{y_2} f_{22} = -x_1 - y_1 + 3y_2 - y_3;$$

$$\max_{y_3} f_{23} = -x_1 - y_1 - y_2 + 3y_3;$$

$$\begin{aligned} \text{s.t. } & 3x_1 + 3y_1 \leqslant 30, \\ & 2x_1 + y_1 \leqslant 20, \\ & y_2 \leqslant 10, \\ & y_2 + y_3 \leqslant 15, \\ & y_3 \leqslant 10, \\ & x_1 + 2y_1 + 2y_2 + y_3 \leqslant 40, \\ & x_1, y_1, y_2, y_3 \geqslant 0. \end{aligned}$$

The individual optimal solutions of the top and bottom level DMs are $f^*_{11} = 35$ (or f^{T}_{11} in Section 3) at $\mathbf{x}^{11*} = (x_1, y_1, y_2, y_3) = (10, 0, 10, 0)$ or (5, 5, 10, 3), $f^*_{21} = 30$ at $\mathbf{x}^{21*} = (0, 10, 0, 0)$, $f^*_{22} = 30$ at $\mathbf{x}^{22*} = (0, 0, 10, 0)$ and $f^*_{23} = 30$ at $\mathbf{x}^{23*} = (0, 0, 0, 10)$. Assume that $f'_{11} = 10$ and $f'_{21} = f'_{22} = f'_{23} = 0$ for negative objective values are not preferred, and that for x_1 the upper-level DM fully satisfies all values between 5 and 10, and the negative and positive-side tolerances are 5 and 0, respectively. Its membership function then has the same form as (8).

By (14), we obtain a satisfactory solution $\mathbf{f} = (f_{11}, f_{21}, f_{22}, f_{23}) = (31.07, 6.43, 6.43, 6.43)$ with $\gamma = 0.21$ at $\mathbf{x} = (1.07, 7.5, 7.5, 7.5)$. In comparison with Anandalingam's solution [29] $\mathbf{f} = (35, -5, 15, -5)$ at $\mathbf{x} = (5, 5, 10, 5)$, the proposed solution is much acceptable for all DMs.

4.2. MLPP

A multi-level programming problem (MLPP) can be defined as a k-person, non-zero sum game with perfect information in which each player moves sequentially from top-down. This problem is a nested hierarchical structure in which there are k levels of DMs [30] and can be represented as:

$$\max_{\mathbf{x}_1} f_1(\mathbf{x}) = \Sigma_j c_{1j} x_j, \qquad \text{(1st level)} \qquad (15)$$

where $\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k$ solve

$$\max_{\mathbf{x}_2} f_2(\mathbf{x}) = \Sigma_j c_{2j} x_j, \qquad \text{(2nd level)}$$

where $\mathbf{x}_3, \ldots, \mathbf{x}_k$ solve

...

where $\mathbf{x}_k$ solves

$$\max_{\mathbf{x}_k} f_k(\mathbf{x}) = \Sigma_j c_{kj} x_j, \qquad (k\text{th level})$$

$$\begin{aligned} \text{s.t. } & \Sigma_k \mathbf{A}_k \mathbf{x}_k \leqslant \mathbf{b}, \quad k = 1, 2, \ldots, k, \\ & x_j \geqslant 0, \quad j = 1, 2, \ldots, n, \end{aligned}$$

where $\cup_k \{\mathbf{x}_k | k = 1, 2, \ldots, k\} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{x}_k$ is the control variable vector of the kth level DM. Let us denote the constraint set of (15) as $\mathbf{X}$.

To solve (15), the top-level DM provides his or her preferred ranges of f_1 and $\mathbf{x}_1$ to the second-level DM and the second-level DM solves his or her problem with the additional preference information from the top-level DM as discussed in Section 3. The same decision-making process as that of BLPP proceeds until a satisfactory solution is reached. By following the satisfactory solution, both DMs individually rebuild or build their own (revised) membership functions which become the additional constraints of the third-level DM. The solution of the third-level DM is proposed to the upper levels; if any upper levels are not satisfied with this proposal, the third-level DM will then solve a new problem with new membership functions from the upper-level DMs until a satisfactory solution is reached. This procedure continues until the kth-level DM's solution satisfies all DMs and the final solution will be a satisfactory solution for (15).

With the required membership functions obtained by (7) and (8), the kth-level DM has the following problems:

$$\begin{aligned}
&\max_{\mathbf{x}_k} f_k(\mathbf{x}), \qquad (16)\\
&\text{s.t. } \mathbf{x} \in \mathbf{X},\\
&\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \alpha_1 \mathbf{I} \text{ and } \mu_{f_1}[f_1(\mathbf{x})] \geqslant \beta_1,\\
&\quad \mu_{\mathbf{x}_2}(\mathbf{x}_2) \geqslant \alpha_2 \mathbf{I} \text{ and } \mu_{f_2}[f_2(\mathbf{x})] \geqslant \beta_2,\\
&\quad \ldots\\
&\mu_{\mathbf{x}_{(k-1)}}[\mathbf{x}_{(k-1)}) \geqslant \boldsymbol{\alpha}_{(k-1)} \mathbf{I} \text{ and } \mu_{f_{(k-1)}}[f_{(k-1)}(\mathbf{x})] \geqslant \beta_{(k-1)},\\
&\alpha_1, \ldots, \alpha_{(k-1)} \in [\mathbf{0}, \mathbf{1}] \text{ and } \beta_1, \ldots, \beta_{(k-1)} \in [0, 1],
\end{aligned}$$

for $k = 2, 3, \ldots, n$. A straightforward extension of (10) is:

$$\begin{aligned}
&\max \lambda, \qquad (17)\\
&\text{s.t. } \mathbf{x} \in \mathbf{X},\\
&\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \lambda \mathbf{I} \text{ and } \mu_{f_1}[f_1(\mathbf{x})] \geqslant \lambda,\\
&\quad \mu_{\mathbf{x}_2}(\mathbf{x}_2) \geqslant \lambda \mathbf{I} \text{ and } \mu_{f_2}[f_2(\mathbf{x})] \geqslant \lambda,\\
&\quad \ldots\\
&\quad \mu_{\mathbf{x}_{(k-1)}}(\mathbf{x}_{(k-1)}) \geqslant \lambda \mathbf{I} \text{ and } \mu_{f_{(k-1)}}[f_{(k-1)}(\mathbf{x})] \geqslant \lambda,\\
&\quad \mu_{f_k}[f_k(\mathbf{x})] \geqslant \lambda,\\
&\quad \lambda \in [0, 1],
\end{aligned}$$

where $\lambda = \min\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{(k-1)}, \beta_1, \ldots, \beta_{(k-1)}, \beta_k\}$. If the solution of (17) does not satisfy some of the upper-level DMs, they should then modify their membership functions, and resolve (17) with new preference information described in Section 3.

It is noted that if there is a constraint which makes the problem infeasible (over-restricts the decision space), this constraint should be examined carefully and may be given a tolerance in order to generate a reasonable, feasible domain.

Example 3

Consider the following TLPP [23]:

$$\max_{x_1} f_1 = 7x_1 + 3x_2 - 4x_3,$$

where x_2 and x_3 solve

$$\max_{x_2} f_2 = x_2.$$

where x_3 solves

$$\max_{x_3} f_3 = x_3,$$

$$\begin{aligned} \text{s.t. } & x_1 + x_2 + x_3 \leqslant 3, \\ & x_1 + x_2 - x_3 \leqslant 1, \\ & x_1 + x_2 + x_3 \geqslant 1, \\ & -x_1 + x_2 + x_3 \leqslant 1, \\ & x_3 \leqslant 0.5, \\ & x_1, x_2 \text{ and } x_3 \geqslant 0, \end{aligned}$$

whose constraint set is depicted as **X**.

The individual optimal solutions are: $f_1^* = 8.5$ (or f_1^{T} in Section 3) at $\mathbf{x}^{1*} = (1.5, 0, 0.5)$, $f_2^* = 1$ at $\mathbf{x}^{2*} = (0, 1, 0)$ or (0.5, 1, 0.5) and $f_3^* = 0.5$ at $\mathbf{x}^{3*} = (1.5, 0, 0.5)$, (0.5, 1, 0.5) or (0, 0.5, 0.5). $f_1' = 3$, $f_2' = 0$ and $f_3' = 0$. Also, assume that control decision x_1 should be around 1.5 with negative and positive-side tolerances 1.5 and 0, respectively. The satisfactory solution of (17) with $k = 2$ is $f = (f_1, f_2) = (6.18, 0.58)$ at $\mathbf{x} = (0.92, 0.58, 0.5)$ with the satisfaction level $\lambda = 0.58$. Assume that the top-level DM satisfies this solution. Then we go to the third level.

If the above solution is not acceptable, the top-level DM may modify his/her membership functions as follows: $f_1^* = 6.18$ and $f_1' = 0$; and x_1 should be around 0.95 with negative- and positive-side tolerances 0.95 and 0, respectively. The 2nd-level DM provides: $f_2^* = 0.58$ and $f_2' = 0$; and x_2 should be around 0.58 with negative and positive-side tolerances 0.58 and 0, respectively. Then, the 3rd-level DM has the problem of (17) with $k = 3$ whose solution is $f = (f_1, f_2, f_3) = (6.18,\ 0.58,\ 0.5)$ at $\mathbf{x} = (0.92, 0.58, 0.5)$ with the satisfactory level $\lambda = 1.00$. Because of $\lambda = 1.00$, this solution is acceptable under prior preference information or membership functions. However, upper-level DMs may want to change their membership functions, then problem (17) with $k = 3$ will be resolved when a satisfactory solution which satisfies all upper-level DMs is reached.

In comparison with Anandalingam's solution $f = (4.5, 1, 0.5)$ at $\mathbf{x} = (0.5, 1, 0.5)$ [5], our solution shows that the top-level DM has a leading position.

4.3. MLDPP

The multi-level decentralized programming problem (MLDPP) is a general decentralized planning problem which includes a multi-level structure and more than one division in each lower level. The hierarchical structure can be represented in Fig. 5. Thus, the proposals in Sections 4.1 and 4.2 can be extended to solve some specific MLDPPs. Because of its complexity, it is difficult to propose a general methodology to solve a general MLDPP, especially when the constraint set has various effects on all levels' DMs. If the constraints can be decomposed to individual levels and divisions, then we can decompose the original structure into substructures which are composed of $(k-1)$ single upper-level DMs and many divisions or DMs forming the lower level in the kth-level. For example, if we have a three-level decentralized programming problem (TLDPP) (see Fig. 5 too), we will have s subproblems and each subproblem is composed of {center $f_1 \rightarrow$ division $f_{2i} \rightarrow$ subdivision, f_{3t}, $t = 1, 2, \ldots, t_i$}, $i = 1, 2, \ldots, s$, if the constraint set can be decomposed into s individual (independent) subsets. The concepts discussed in the previous sections can be applied straightforward to solve these s subproblems as:

$$\max_{\mathbf{x}_1} f_1(\mathbf{x}) = \Sigma_j c_{1j} x_j, \qquad \text{(1st level)} \quad (18)$$

where $\mathbf{x}_{2i}, \mathbf{x}_{3i1}, \ldots, \mathbf{x}_{3it_i}$ solve

$$\max_{\mathbf{x}_{2i}} f_{2i}(\mathbf{x}) = \Sigma_j c_{2ij} x_j, \qquad \text{(2nd level)}$$

where $\mathbf{x}_{3i1}, \ldots, \mathbf{x}_{3it_i}$ solve

$$\max_{\mathbf{x}_{3i1}} f_{3i1}(\mathbf{x}) = \Sigma_j c_{3i1j} x_j; \qquad \text{(3rd level)}$$

$\ldots$

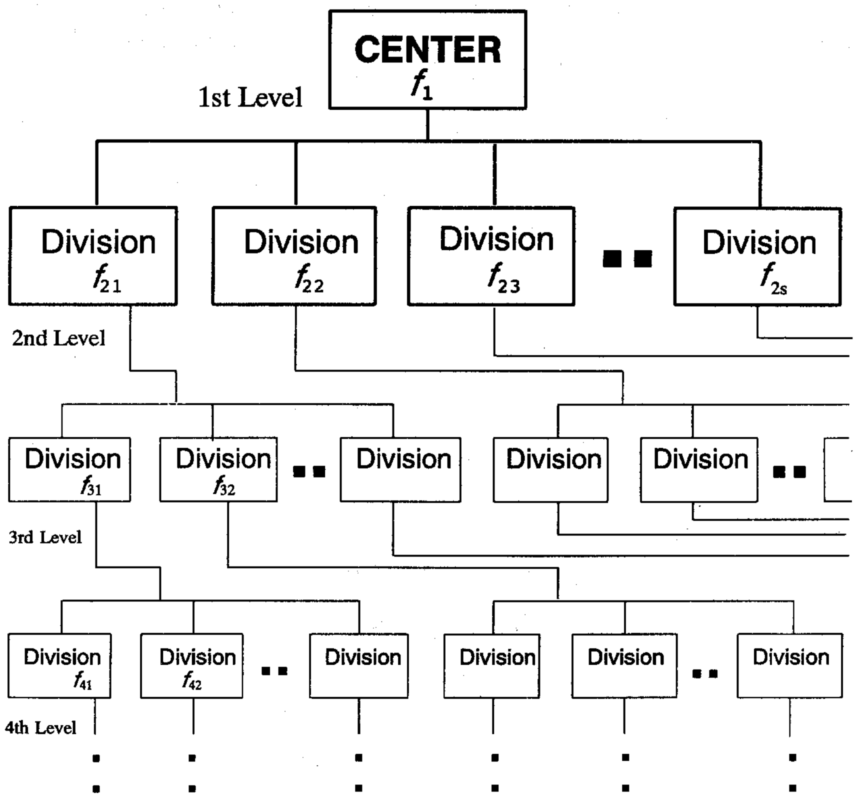

Fig. 5. Decentralizd hierarchical structure.

$$\max_{\mathbf{x}_{3it_i}} f_{3it_i}(\mathbf{x}) = \Sigma_j c_{3it_i j} x_j;$$
$$\text{s.t. } \mathrm{A}_1\mathbf{x}_1 + \mathrm{A}_{2i}\mathbf{x}_{1i} + \mathrm{A}_{3i1}\mathbf{x}_{3i1} + \cdots + \mathrm{A}_{3it_i}\mathbf{x}_{3it_i} \leqslant \mathbf{b},$$
$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n,$$

for $i = 1, 2, \ldots, s$, which can be solved by the following auxiliary problems [a combination of (14) and (17)]:

$$\begin{aligned} &\max \lambda, \\ &\text{s.t. } \mathbf{x} \in \mathbf{X}, \\ &\quad \mu_{\mathbf{x}_1}(\mathbf{x}_1) \geqslant \lambda \mathbf{I}, \\ &\quad \mu_{f_1}[f_1(\mathbf{x})] \geqslant \lambda, \\ &\quad \mu_{\mathbf{x}_{2i}}(\mathbf{x}_{1i}) \geqslant \lambda \mathbf{I}, \\ &\quad \mu_{f_{2i}}[f_{1i}(\mathbf{x})] \geqslant \lambda, \\ &\quad \mu_{f_{3ik}}[f_{2ik}(\mathbf{x})] \geqslant \lambda, \quad k = 1, \ldots, t_i, \\ &\quad \lambda \in [0, 1], \end{aligned} \tag{19}$$

where $\mathbf{X}$ denotes the constraint set of (18).

On the other hand, in many cases the constraint set must be considered altogether for DMs in all levels and divisions. We might then consider all the DMs in the same level as a group. In this way, TLDPP is equivalent to a TLPP with one DM in the first level, s DMs in the second level and $t = \Sigma_i t_i$ DMs in the third level. To discuss this problem, let us first define $f_{ki}(\mathbf{x})$ and $\mathbf{x}_{ki}$ [$i = 1$ when $k = 1$, $i = 1, \ldots, s$ when $k = 2$ and $i = 1, \ldots, t$ $(= t_1 + t_2 + \cdots t_s)$ when $k = 3$] as the objective function and control variable vectors of the kth-level ith-division DM, respectively, and c_{kij} as the cost coefficient of the decision variable $x_j, j = 1, 2, \ldots, n$, for the ith-division of the kth-level. Then, by extending (11) and (15), the TLDPP can be formulated as:

$$
\begin{aligned}
&\max_{\mathbf{x}_{11}} f_{11}(\mathbf{x}) = \Sigma_j \mathbf{c}_{11j}\mathbf{x}_j; && \text{(1st level)} \quad (20)\\
&\text{where } \mathbf{x}_{21}, \mathbf{x}_{22}, \ldots, \mathbf{x}_{2s}, \mathbf{x}_{31}, \mathbf{x}_{32}, \ldots, \mathbf{x}_{3t} \text{ solve}\\
&\quad \max_{\mathbf{x}_{21}} f_{21}(\mathbf{x}) = \Sigma_j \mathbf{c}_{21j}\mathbf{x}_j; && \text{(2nd level)}\\
&\quad \ldots\\
&\quad \max_{\mathbf{x}_{2s}} f_{2s}(\mathbf{x}) = \Sigma_j \mathbf{c}_{2sj}\mathbf{x}_j;\\
&\quad \text{where } \mathbf{x}_{31}, \mathbf{x}_{32}, \ldots, \mathbf{x}_{3t} \text{ solve}\\
&\quad \max_{\mathbf{x}_{31}} f_{31}(\mathbf{x}) = \Sigma_j \mathbf{c}_{31j}\mathbf{x}_j; && \text{(3rd level)}\\
&\quad \ldots\\
&\quad \max_{\mathbf{x}_{3t}} f_{3t}(\mathbf{x}) = \Sigma_j \mathbf{c}_{3tj}\mathbf{x}_j;\\
&\quad \text{s.t. } \sum_{\forall k,i} \mathrm{A}_{ki}\mathbf{x}_{ki} \leqslant \mathbf{b},\\
&\qquad x_j \geqslant 0, \quad j = 1, 2, \ldots, n.
\end{aligned}
$$

For the second level, we have the following auxiliary problem:

$$
\begin{aligned}
&\max \lambda, && (21)\\
&\text{s.t. } \mathbf{x} \in \mathbf{X},\\
&\quad \mu_{\mathbf{x}_{11}}(\mathbf{x}_{11}) \geqslant \lambda \mathbf{I},\\
&\quad \mu_{f_{11}}[f_{11}(\mathbf{x})] \geqslant \lambda,\\
&\quad \mu_{f_{2i}}[f_{2i}(\mathbf{x})] \geqslant \lambda, \quad i = 1, \ldots, s,\\
&\quad \lambda \in [0, 1],
\end{aligned}
$$

which is similar to (14). By considering the satisfactory solution, DMs in the first and second levels then refine their membership functions which are further passed to the third level. The third-level DMs have:

$$
\begin{aligned}
&\max \lambda, && (22)\\
&\text{s.t. } \mathbf{x} \in \mathbf{X},\\
&\quad \mu_{\mathbf{x}_{11}}(\mathbf{x}_{11}) \geqslant \lambda \mathbf{I} \text{ and } \mu_{f_{11}}[f_{11}(\mathbf{x})] \geqslant \lambda,\\
&\quad \mu_{\mathbf{x}_{21}}(\mathbf{x}_{21}) \geqslant \lambda \mathbf{I} \text{ and } \mu_{f_{21}}[f_{21}(\mathbf{x})] \geqslant \lambda,\\
&\quad \ldots\\
&\quad \mu_{\mathbf{x}_{2s}}(\mathbf{x}_{2s}) \geqslant \lambda \mathbf{I} \text{ and } \mu_{f_{2s}}[f_{2s}(\mathbf{x})] \geqslant \lambda,\\
&\quad \mu_{f_{3i}}[f_{3i}(\mathbf{x})] \geqslant \lambda, \quad i = 1, 2, \ldots, t,\\
&\quad \lambda \in [0, 1],
\end{aligned}
$$

whose solution will be a satisfactory solution for the TLDPP under presumed phenomenon. If the solution does not satisfy some of the DMs, an interactive process as discussed above should be considered.

Because the constraint set or feasible domain is not separable, DMs in the same level may need to reach solutions simultaneously. This problem, or a multi-objective decision-making problem, does not necessarily become a single-person decision-making problem as pointed out by Bard [1]. If a parallel computer system is available, which is not uncommon nowadays, multi-person decision-making processes can be carried out as a single-person decision-making problem in terms of techniques. Philosophically, it is also natural to use multi-objective decision-making methods to model multi-person decision-making problems if their feasible domain is mutually dependent and inseparable. However, the general hierarchical structure as shown in Fig. 5 can be achieved during interactive processes. While re-examining and changing membership (preference) functions, only direct upper and lower-level DMs current satisfactory objective values and decisions are considered. That is, the relationship of the hierarchy structure is explicitly reflected in (re)forming membership functions. In this way, our proposal of (21) and (22) indeed solves a general MLDP problem.

5. CONCLUDING REMARKS

Bard's grid search algorithm first uses a parametric approach to obtain an efficient solution and then checks if this solution satisfies Stackelberg optimal conditions until an efficient Stackelberg solution is reached. On the other hand, Wen and Hsu's two-phase approach first locates the Stackelberg solution and then checks if it is Pareto-optimal. If not, a multiple objective optimization technique is further used to obtain efficient solutions as final solutions. Both methods basically follow a classical multiple-level decision-making process with some modifications from multiple objective decision-making concepts. By following Lai's concepts, the proposed decision-making process proceeds from top to bottom in a natural and straightforward manner—a boss supervises the solution search direction, and explicitly considers and displays preference information through membership functions. Preference information is delivered from upper levels to lower levels sequentially, and the lower-level DM solves his problem under restrictions of the upper level DMs' requirements.

Unlike lexicographic methods where higher-level objectives are only used to restrict lower-level feasibilities, both decision variables and objective functions are considered, and thus the proposed satisfactory solution should be more practical and reasonable. Since the solution search is based on the change of membership function instead of vertex enumeration, even a large-scale problem can be solved with little computation. For a non-linear programming problem, the proposed approach at least will not increase the order of non-linearity.

Since different membership functions and operations as mentioned in Section 3 provide different satisfactory solutions, it is important to explore various functions and operators, as well as to allow DMs to change function forms and operators in the above discussed interactive processes; thereby, we can build a more complete decision support system for solving MLDPP. At the same time, we should also extend our approach to solve non-linear, integer or mixed integer (non-linear) multi-level programming problems.

Finally, we would also like to mention that input data or parameters are often imprecise or fuzzy (see Lai and Hwang [26, 28]) in a wide variety of hierarchical optimization problems such as defense problems, transportation network designs, economical analysis, financial control, energy planning, government regulation, equipment scheduling, organizational management, quality assurance, conflict resolution and so on. Developing methodologies and new concepts for solving fuzzy and possibilistic multi-level programming problems is a practical and interesting direction for future studies.

Acknowledgement—We would like to thank Professor John C. H. Wu at Kansas State University and three anonymous referees for their valuable suggestions and comments.

REFERENCES

1. J. F. Bard, Coordination of a multidivisional organization through two levels of management. *Omega* **11**, 457–468 (1983).
2. R. M. Burton, The multilevel approach to organizational issues of the firm—a critical review. *Omega* **5**, 395–414 (1977).
3. O. Ben-Ayed and C. E. Blair, Computational difficulties of bilevel linear programming. *Ops Res.* **38**, 556–560 (1990).
4. Y. J. Lai, Hierarchical optimization: A satisfactory solution. *Fuzzy Sets and Systems* (forthcoming).
5. J. F. Bard, An efficient point algorithm for a linear two-stage optimization problem. *Ops Res.* **31**, 670–684 (1983).

6. U. P. Wen and S. T. Hsu, A note on a linear bilevel programming algorithm based on bicriteria programming. *Computers Ops Res.* **16**, 79–83 (1989).
7. U. P. Wen and S. T. Hsu, Efficient solutions for the linear bilevel programming problem. *Eur J. Opl Res.* **62**, 354–362 (1991).
8. U. P. Wen and S. T. Hsu, Linear bi-level programming problems—a review. *J. Opl Res. Soc.* **42**, 125–133 (1991).
9. O. Ben-Ayed, Bilevel linear programming. *Computers Ops Res.* **20**, 485–501 (1993).
10. W. F. Bialas and M. H. Karwan, On two-level optimization. *IEEE Trans. Autom. Control* **AC-27**, 211–214 (1982).
11. W. F. Bialas and M. H. Karwan, Two-level linear programming. *Mgmt Sci.* **30**, 1004–1020 (1984).
12. W. Candler and R. Townsley, A linear two-level programming problem. *Computers Ops Res.* **9**, 59–76 (1982).
13. A. Haurie, G. Savard and D. J. White, A note on: an efficient point algorithm for a linear two-stage optimization problem. *Ops Res.* **38**, 553–555 (1990).
14. J. F. Bard, Optimality conditions for the bilevel programming problem. *Nav. Res. Logist. Q.* **31**, 13–26 (1984).
15. P. A. Clark and A. W. Westerberg, A note on the optimality conditions for the bilevel programming problem. *Nav. Res. Logist. Q.* **35**, 413–418 (1988).
16. G. Ünlü, A linear bilevel programming algorithm based on bicriteria programming. *Computers Ops Res.* **14**, 173–179 (1987).
17. W. Candler, A linear bilevel programming algorithm: a comment. *Computers Ops Res.* **15**, 297–298 (1988).
18. P. Marcotte and G. Savard, A note on the Pareto optimality of solutions to the linear bilevel programming problem. *Computers Ops Res.* **18**, 355–359 (1991).
19. J. Fortuny-Amat and B. McCarl, A representation and economic interpretation of a two-level programming problem. *J. Opl Res. Soc.* **32**, 783–792 (1981).
20. J. F. Bard and J. E. Falk, An explicit solution to the multi-level programming problem. *Computers Ops Res.* **9**, 77–100 (1982).
21. J. F. Bard and J. T. Moore, A branch and bound algorithm for the bilevel programming problem. *SIAM J. Sci. Stat. Comput.* **11**, 281–292 (1990).
22. U. P. Wen and Y. H. Yang, Algorithms for solving the mixed integer two-level linear programming problem. *Computers Ops Res.* **17**, 133–142 (1990).
23. G. Anandalingam and D. J. White, A solution for the linear static Stackelberg problem using penalty functions. *IEEE Trans. Autom. Control* **35**, 1170–1173 (1990).
24. T. Basar and G. J. Olsder, *Dynamic Noncooperative Game theory*. Academic Press, New York (1982).
25. J.-P. Aubin, *Optima and Equilibria: An Introduction to Nonlinear Analysis*. Springer-Verlag, Berlin (1993).
26. Y. J. Lai and C. L. Hwang, *Fuzzy Mathematical Programming—Methods and Applications*. Springer-Verlag, Berlin (1993).
27. R. Bellman and L. A. Zadeh, Decision-making in a fuzzy environment. *Mgmt Sci.* **17**, B141–164 (1970).
28. Y. J. Lai and C. L. Hwang, *Fuzzy Multiple Objective Decision Making—Methods and Applications*. Springer-Verlag, Berlin (1994).
29. G. Anandalingam, A mathematical programming model of decentralized multi-level systems. *J. Opl Res. Soc.* **39**, 1021–1033 (1988).
30. G. Anandalingam and T. L. Friesz, Hierarchical optimization: an introduction. *Annals of Operations Research* (Edited by G. Anandalingam and T. L. Friesz), Vol. 34, pp. 1–11 (1992).