F L I P
Multiobjective
Fuzzy Linear Programming
Package

Piotr Czyżak, Roman Słowiński

CP-91-008
June 1991

Foreword

This Collaborative Paper is one of a series which presents the different software packages designed and implemented for interactive decision support. These packages constitute the outcome of the contracted study agreement between the System and Decision Sciences Program at IIASA and several Polish scientific institutions. The theoretical part of these results is presented in the IIASA Collaborative Paper CP-90-008 entitled *Contributions to Methodology and Techniques of Decision Analysis (First Stage)*, edited by Andrzej Ruszczyński, Tadeusz Rogowski and Andrzej P. Wierzbicki.

The distributable versions of the software are usually tailored for the illustration of methodology and possible applications. However, for most of these software packages there exists a version made for a specific application and it is possible to modify each software package for a specific real-life application (if the corresponding mathematical programming model is of the type for which a particular package has been designed).

All software developed within the scientific cooperation mentioned above is available either at distribution cost or free of charge for scientific non-commercial usage by institutions and individuals from the countries which are members of IIASA. Inquiries about more detailed information and requests for the software should be addressed to the Leader of the MDA Project.

This volume contains the theoretical and methodological backgrounds as well as the User's Guide for a FLIP package designed to help in analysis of multiobjective linear programming (MOLP) problems in an uncertain environment.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program

Abstract

FLIP (Fuzzy LInear Programming) is a package designed to help in analysis of multiobjective linear programming (MOLP) problems in an uncertain environment. The uncertainty of data is modeled by L-R type fuzzy numbers. They can appear in the objective functions as well as on the both sides of the constraints.

The input data to the FLIP package include the characteristics of the analyzed fuzzy MOLP problem, i.e., the number of criteria, constraints and decision variables, fuzzy cost coefficients for every objective and fuzzy coefficients of LHS and RHS for all constraints. The data loading is supported by a graphical presentation of fuzzy coefficients. The calculation is preceded by a transformation of the fuzzy MOLP problem into a multiobjective linear fractional program. It is then solved with an interactive method using a linear programming procedure as the only optimiser. In every iteration, one gets a series of solutions that are presented very clearly in a graphical and numerical form.

In FLIP, interaction with the user takes place at two levels: first, when safety parameters have to be defined in the transformation phase, and second, when the associate deterministic problem is solved.

The package is written in TURBO-Pascal and can be used on microcomputers compatible with IBM-PC XT/AT with hard disc and a graphic card.

Contents

F L I P
Multiobjective
Fuzzy Linear Programming
Package

*Piotr Czyżak, Roman Słowiński**

Part I
Methodological guide

1 Formulation of the problem

FLIP solves the following MOLP problem with fuzzy coefficients:

$$(P1) \qquad minimize \begin{bmatrix} \tilde{z}_1 = \underline{\tilde{c}}_1\underline{x} \\ \vdots \\ \tilde{z}_k = \underline{\tilde{c}}_k\underline{x} \end{bmatrix}$$

$$s.t. \quad \begin{array}{llll} \underline{\tilde{a}}_i\underline{x} & \leq & \tilde{b}_i & i = 1,...,m_1 & (A) \\ \underline{\tilde{a}}_i\underline{x} & \geq & \tilde{b}_i & i = m_1 + 1,...,m & (B) \\ \underline{x} & \geq & \underline{0} & & \end{array}$$

where $\underline{x}$ is a column vector of n decision variables, $\underline{\tilde{c}}_1$,...,$\underline{\tilde{c}}_k$ are row vectors of fuzzy cost coefficients corresponding to the objective functions $\tilde{z}_1$,...,$\tilde{z}_k$, $\underline{\tilde{a}}_i$ is the i-th row of the matrix of fuzzy coefficients, and $\tilde{b}_i$ is its corresponding fuzzy right-hand-side. The equality constraints are excluded from the above formulation since they can be obviously represented by pairs of inequality constraints. It is assumed, moreover, that for the particular objectives, the decision maker (DM) is in a position to define fuzzy aspiration levels, thought of as goals, denoted by $\tilde{g}_1,...,\tilde{g}_k$. All fuzzy coefficients are given as fuzzy numbers, i.e. normal and convex fuzzy subsets of the real line.

FLIP has been proposed to solve a water supply planning problem under uncertainty. Its presentation in [11] has been preceded by a brief survey of approaches to fuzzy mathematical programming proposed before 1984. The survey has shown that there was no method which would deal with a multiobjective linear programming problem with fuzzy coefficients in the objective functions and on the both sides of the constraints. Since then, a lot of new work has been done; a short updated review has been made in [12], together with some refinements of FLIP. More comprehensive surveys have been done recently by Dubois [6], Luhandjula [8], and Inuiguchi, Ichihashi and Tanaka [7]. Others, like Delgado, Verdegay and Vila [5], Rommelfanger [9], and Sakawa and Yano [10], presented their new methods on the background of existing ones.

*Technical University of Poznań, Institute of Computing Science, Poznań, Poland

The idea of FLIP relies on an observation that for a given $\underline{x}$, the main question to be answered consists in the comparison of the left- and right-hand-sides in objectives and constraints which are fuzzy numbers. Assuming an L-R representation of fuzzy coefficients, a comparison principle has been proposed [11], which allows a transformation of the fuzzy MOLP problem into a multiobjective linear fractional programming problem. The best compromise solution of the latter problem ensures the "best consistency" between the goals and the objective functions, and satisfies the constraints with a given "safety".

In the next section, we present the comparison principle of fuzzy numbers which is used in the transformation of the fuzzy MOLP problem into a deterministic mathematical program. The transformation is described in chapter 3, and in chapter 4, the way of solving the associate deterministic problem is outlined with a special emphasis on interactive steps.

2   Comparison of L-R fuzzy numbers

A convenient representation of a fuzzy number $\tilde{a}$ is a triple of parameters (a, α, β) of its membership function

$$\mu_a(x) = \begin{cases} L((a-x)/\alpha) & \text{if } x \le a \\ R((x-a)/\beta) & \text{if } x \ge a \end{cases}$$

where a is a "most possible" middle value, α and β are nonnegative left and right "spreads" of $\tilde{a}$, respectively, and L, R are symmetric bell-shaped reference functions that are decreasing in $(-\infty, \infty)$ and L(0)=R(0)=1, L(1)=R(1)=0; $\tilde{a}$ is said to be an L-R fuzzy number. When the spreads are zero, then $\tilde{a}$ is a nonfuzzy (crisp) number equal to a.

Let $\tilde{a} = (a, \alpha, \beta)_{LR}$ and $\tilde{b} = (b, \gamma, \delta)_{L'R'}$ be fuzzy numbers. In order to evaluate the possibility for $\tilde{b}$ to be greater than $\tilde{a}$, we use two different indices: σ and π, called optimistic and pessimistic, respectively.

(a)Optimistic index σ

Let us suppose a situation presented in Fig.1. Optimistic index $\sigma\left(\tilde{b} > \tilde{a}\right)$ is defined as an ordinate of the intersection point of R' and L, i.e. the right slope of $\tilde{b}$ with the left slope of $\tilde{a}$:

$$\sigma(\tilde{b} > \tilde{a}) = R'((d-b)/\delta) = L((a-d)/\alpha) = \omega \tag{1}$$

where d is an abscissa of the intersection point. The smaller $\sigma(\tilde{b} > \tilde{a})$ is, the less true is the assertion that $\tilde{b}$ is greater than $\tilde{a}$.

Let us observe that (2.1) is equivalent to

$$b + \delta R'^{-1}(\omega) = a - \alpha L^{-1}(\omega) \tag{2}$$

and

$$\delta R'^{-1}(\omega) + \alpha L^{-1}(\omega) = a - b \tag{3}$$

For $F(\omega) = \left(\delta R'^{-1}(\omega) + \alpha L^{-1}(\omega)\right)^{-1}$, we have

$$F^{-1}(\omega) = a - b \tag{4}$$
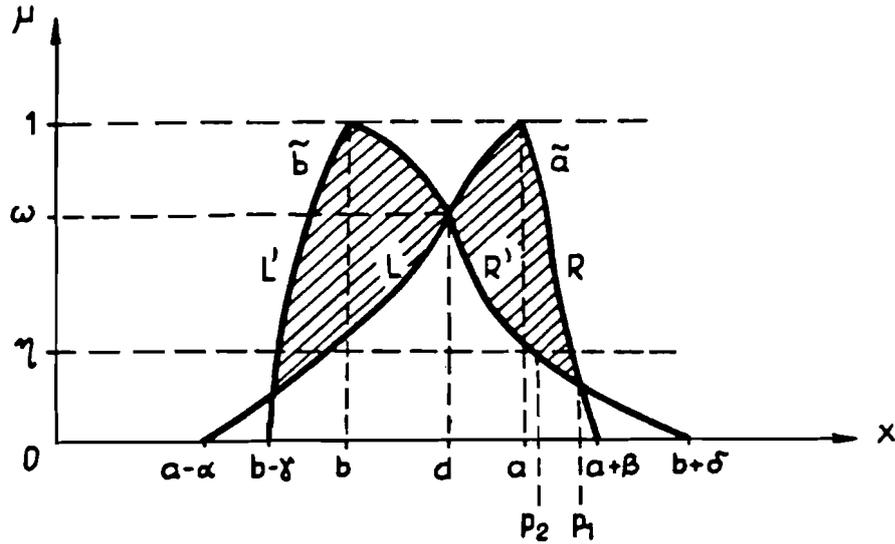
Fig.1. Comparison of fuzzy numbers $\tilde{a}$ and $\tilde{b}$

which implies

$$\sigma(\tilde{b} > \tilde{a}) = \omega = F(a - b) \tag{5}$$

Similarly, $\sigma(\tilde{a} > \tilde{b})$ is defined as an ordinate of the intersection point of R and L', i.e. the right slope of $\tilde{a}$ with the left slope of $\tilde{b}$:

$$\sigma(\tilde{a} > \tilde{b}) = \psi = G(b - a) \tag{6}$$

where $G(\psi) = (\gamma L'^{-1}(\psi) + \beta R^{-1}(\psi))^{-1}$.

If L=R' and L'=R then (2.5) and (2.6) take the form

$$\sigma(\tilde{b} > \tilde{a}) = L((a - b)/(\alpha + \delta)) \tag{7}$$

$$\sigma(\tilde{a} > \tilde{b}) = R((b - a)/(\beta + \gamma)) \tag{8}$$

In order to say that $\tilde{b} > \tilde{a}$, we need both $\sigma(\tilde{a} > \tilde{b})$ and $\sigma(\tilde{b} > \tilde{a})$. If for instance $\sigma(\tilde{b} > \tilde{a}) \geq 1$, we know that either $\tilde{b}$ is greater than $\tilde{a}$, or both fuzzy numbers are too close to be separated. Then, we may choose a threshold $0 \leq \tau \leq 1$ and admit that $\tilde{b}$ is greater than $\tilde{a}$ at level τ as soon as $\sigma(\tilde{a} > \tilde{b}) \leq \tau$. If $\min\{\sigma(\tilde{a} > \tilde{b}), \sigma(\tilde{b} > \tilde{a})\} \geq \tau$, we say that $\tilde{a}$ and $\tilde{b}$ are approximately equal.

In the case of weak inequality $\tilde{b} \geq \tilde{a}$, we only need $\sigma(\tilde{b} > \tilde{a})$. Indeed, for $\sigma(\tilde{b} > \tilde{a}) \geq 1$ the weak inequality is satisfied for any value of $\sigma(\tilde{a} > \tilde{b})$. Then we may choose a credibility constant $0 \leq \tau \leq 1$ and admit that $\tilde{b} \geq \tilde{a}$ at credibility level τ as soon as $\sigma(\tilde{b} > \tilde{a}) \geq \tau$.

The comparison index σ can be seen as optimistic because even for a high value of $\sigma(\tilde{b} > \tilde{a})$, the possibility for $x \in \mathcal{R}$, fuzzily restricted to belong to $\tilde{a}$, to be greater than $y \in \mathcal{R}$, fuzzily restricted to belong to $\tilde{b}$, may be quite big. As a measure of this possibility one can consider the hatched area marked in Fig.1. Thus, in order to make the comparison more credible, we should use σ conjointly with the pessimistic index π.

(b) Pessimistic index π

The pessimistic index follows from the comparison of the right slopes of $\tilde{a}$ and $\tilde{b}$ at some level $0 \leq \eta \leq 1$:

$$\pi(\tilde{b} >_{\eta} \tilde{a}) = b + \delta R'^{-1}(\eta) - a - \beta R^{-1}(\eta) = p_2 - p_1 \tag{9}$$

$$\pi(\tilde{a} >_\eta \check{b}) = a + \beta R^{-1}(\eta) - b - \delta R'^{-1}(\eta) = p_1 - p_2 \tag{10}$$

For $\eta = 0$ we have

$$\pi(\check{b} >_0 \tilde{a}) = b + \delta - a - \beta \tag{11}$$

$$\pi(\tilde{a} >_0 \check{b}) = a + \beta - b - \delta \tag{12}$$

Thus, a joint use of $\sigma(\check{b} > \tilde{a})$ and $\pi(\check{b} >_\eta \tilde{a})$ consists in the comparison of both slopes of a which is supposed to be smaller with the right slope of $\check{b}$ which is supposed to be greater.

Now, we can admit that $\check{b} \geq \tilde{a}$ at credibility levels τ and η if and only if

$$\sigma(\check{b} > \tilde{a}) \geq \tau \tag{13}$$

and

$$\pi(\check{b} >_\eta \tilde{a}) \geq \theta \tag{14}$$

where $\tau, \eta \in [0,1]$ and $\theta \in (-\infty, \infty)$. $\theta \geq 0$ means that for any pair (x,y) such that $x \geq a$, $y \geq b$ and $0 \leq \mu_a(x) = \mu_b(y) \leq \eta$, inequality y≥x is true. A negative value of θ makes possible that inequality y≥x is not true.

The constants τ, η and θ are called "safety parameters" because they are responsible for the safety of the assertion that $\check{b}$ is greater than $\tilde{a}$. Instead of safety, we could speak of course about risk which is a complementary term. Let us remark that using τ, η and θ one can control the surface of the hatched area marked in Fig.1 which corresponds to the risk of the above assertion.

3   The associate deterministic problem

Conditions (2.13) and (2.14) can be used directly to transform the fuzzy MOLP problem into an associate deterministic one. Let us separately analyze the constraints and the objective functions of the fuzzy MOLP problem.

Let us suppose that fuzzy coefficients of the constraints are given as L-R fuzzy numbers:

$$\tilde{a} = (a, \alpha, \beta)_{LR}, \quad \check{b} = (b, \gamma, \delta)_{RL}, \qquad\qquad \text{i=1,...m.}$$

For the sake of clarity, we assume that the reference functions of all fuzzy coefficients are of two kinds only: L and R. It can be seen from the preceding considerations that this assumption is not necessary for the calculation of σ and π (cf. [5], [6] and [9], [10]).

It should be specified that all arithmetic operations on fuzzy numbers taking place in (P1) are extended operations in the sense of Zadeh's principle. For any $\underline{x} \geq \underline{0}$, the left side of the i-th constraint can be summarized to the fuzzy number:

$$\underline{\tilde{a}_i x} = (\underline{a_i x}, \underline{\alpha_i x}, \underline{\beta_i x})_{LR}, \qquad\qquad \text{i=1,...,m.}$$

It is easy to verify that for i=1,...,m_1

$$\sigma(\check{b}_i > \underline{\tilde{a}_i x}) = L((\underline{a_i x} - b_i)/(\underline{\alpha_i x} + \delta_i))$$

$$\pi(\check{b}_i >_{\eta_i} \underline{\tilde{a}_i x}) = b_i + \delta_i L^{-1}(\eta_i) - \underline{a_i x} - \underline{\beta_i x} R^{-1}(\eta_i)$$

and for $i = m_1 + 1, ..., m$

$$\sigma(\underline{\tilde{a}}_i\underline{x} > \tilde{b}_i) = R((b_i - \underline{a}_i\underline{x})/(\underline{\beta}_i\underline{x} + \gamma_i))$$

$$\pi(\underline{\tilde{a}}_i\underline{x} >_{\eta_i} \tilde{b}_i) = \underline{a}_i\underline{x} + \underline{\beta}_i\underline{x}R^{-1}(\eta_i) - b_i - \delta_iL^{-1}(\eta_i)$$

For given values of safety parameters τ_i, η_i and θ_i , i=1,...,m, the constraints (A) and (B) of (P1) may be transformed to:

$$L((\underline{a}_i\underline{x} - b_i)/(\underline{\alpha}_i\underline{x} + \delta_i)) \geq \tau_i \qquad\qquad \text{i=1,...,m}_1 \qquad\qquad (15)$$

$$R((b_i - \underline{a}_i\underline{x})/(\underline{\beta}_i\underline{x} + \gamma_i)) \geq \tau_i \qquad\qquad \text{i=m}_1\text{+1,...,m} \qquad\qquad (16)$$

$$b_i + \delta_iL^{-1}(\eta_i) - \underline{a}_i\underline{x} - \underline{\beta}_i\underline{x}R^{-1}(\eta_i) \geq \theta_i \qquad\qquad \text{i=1,...,m}_1 \qquad\qquad (17)$$

$$\underline{a}_i\underline{x} + \underline{\beta}_i\underline{x}R^{-1}(\eta_i) - b_i - \delta_iL^{-1}(\eta_i) \geq \theta_i \qquad\qquad \text{i=m}_1\text{+1,...,m} \qquad\qquad (18)$$

Basing on the property of strict monotonicity of reference functions, one can transform (3.1) and (3.2) into the following equivalent conditions:

$$\underline{a}_i\underline{x} - b_i \leq L^{-1}(\tau_i)(\underline{\alpha}_i\underline{x} + \delta_i) \qquad\qquad \text{i=1,...,m}_1 \qquad\qquad (19)$$

$$b_i - \underline{a}_i\underline{x} \leq R^{-1}(\tau_i)(\underline{\beta}_i\underline{x} + \gamma_i) \qquad\qquad \text{i=m}_1\text{+1,...,m} \qquad\qquad (20)$$

In the case of fuzzy objective functions, σ can be used to evaluate the degree of consistency between fuzzy objectives and fuzzy goals. Let the fuzzy cost coefficients and fuzzy goals be:

$$\tilde{\underline{c}}_l = (\underline{c}_l, \underline{\epsilon}_l, \underline{\kappa}_l)_{LR}, \quad \tilde{g}_l = (g_l, 0, \nu_l)_{LL}, \qquad\qquad \text{l=1,...,k}$$

where the left spread of $\tilde{g}_l$, l=1,...,k, is equal to zero because it is immaterial for the evaluation of consistency between goals and objectives. Here again, the equality of the reference functions is not a necessary assumption. For any $\underline{x}$, the components of the l-th objective function can be summarized with the extension principle to the flat fuzzy number:

$$\tilde{\underline{c}}_l\underline{x} = (\underline{c}_l\underline{x}, \underline{\epsilon}_l\underline{x}, \underline{\kappa}_l\underline{x})_{LR}, \qquad\qquad \text{l=1,...,k.}$$

On the basis of (2.7), we can write for l=1,...,k

$$\sigma(\tilde{g}_l > \tilde{\underline{c}}_l\underline{x}) = L((\underline{c}_l\underline{x} - g_l)/(\underline{\epsilon}_l\underline{x} + \nu_l)) \qquad\qquad (21)$$

In order to ensure the best consistency between goals and objectives, $\sigma(\tilde{g}_l > \tilde{\underline{c}}_l\underline{x})$, l=1,...,k, should be maximized. In consequence, we arrive to the following deterministic mathematical programming problem equivalent to problem (P1):

$$(P2) \quad maximize \begin{bmatrix} f_1(\underline{x}) = L((\underline{c}_1\underline{x} - g_1)/(\underline{\epsilon}_1\underline{x} + \nu_1)) \\ \vdots \\ f_k(\underline{x}) = L((\underline{c}_k\underline{x} - g_k)/(\underline{\epsilon}_k\underline{x} + \nu_k)) \end{bmatrix}$$

$$s.t. \quad \begin{aligned} \underline{a}_i\underline{x} - b_i &\leq L^{-1}(\tau_i)(\underline{\alpha}_i\underline{x} + \delta_i) & \text{i=1,...,m}_1 \quad &(A1) \\ b_i - \underline{a}_i\underline{x} &\leq R^{-1}(\tau_i)(\underline{\beta}_i\underline{x} + \gamma_i) & \text{i=m}_1\text{+1,...,m} \quad &(B1) \\ b_i + \delta_iL^{-1}(\eta_i) - \underline{a}_i\underline{x} - \underline{\beta}_i\underline{x}R^{-1}(\eta_i) &\geq \theta_i & \text{i=1,...,m}_1 \quad &(A2) \\ \underline{a}_i\underline{x} + \underline{\beta}_i\underline{x}R^{-1}(\eta_i) - b_i - \delta_iL^{-1}(\eta_i) &\geq \theta_i & \text{i=m}_1\text{+1,...,m} \quad &(B2) \\ \underline{x} &\geq \underline{0} & &(C) \end{aligned}$$

If reference function L of fuzzy cost coefficients is linear, i.e. L(y)=1-y, than problem (P2) becomes a multiobjective linear fractional programming (MLFP) problem:

$$(P3) \qquad maximize \begin{bmatrix} f_1(\underline{x}) = 1 - (\underline{c}_1\underline{x} - g_1)/(\underline{\epsilon}_1\underline{x} + \nu_1) \\ \vdots \\ f_k(\underline{x}) = 1 - (\underline{c}_k\underline{x} - g_k)/(\underline{\epsilon}_k\underline{x} + \nu_k) \end{bmatrix}$$

s.t. (A1), (B1), (A2), (B2), (C).

In order to be sure that the denominators of $f_1(\underline{x}), ..., f_k(\underline{x})$ are greater than zero for any feasible $\underline{x}$, we may admit that $\nu_l > 0$, l=1,...,k, which is quite natural.

4   Interactive steps

FLIP can be summarized in the following steps:

1. Formulation of problem (P1) and definition of fuzzy coefficients.

2. Definition of fuzzy aspiration levels $\tilde{g}_l$ (l=1,...,k) on objectives.

3. Definition of safety parameters τ_i, η_i and θ_i (i=1,...,m) for fuzzy constraints.

4. Formulation of the associate multiobjective deterministic problem (P2) or (P3).

5. Application of an interactive method for solving the associate deterministic problem formulated in step 4.

6. If a best compromise solution has been found then stop, otherwise return to step 3 for revision of safety parameters. Retraction to steps 1 and 2 for redefinition of fuzzy coefficients and/or aspiration levels is also possible.

Our experience indicates that an interaction with the DM in searching of the best compromise solution is very beneficial for the final decision. So, in step 5, in order to solve the associate multiobjective deterministic problem, we propose to use an interactive method (cf.[16]). If the associate deterministic problem has the form of the MLFP problem (P3), a very convenient interactive procedure is that of Choo and Atkins [1].

Let us recall its general idea informally.

In order to find a starting efficient point of the MLFP problem, the objective functions $f_1, ..., f_k$ are aggregated by the Chebyshev norm which is the maximum weighted deviation from the ideal point $\underline{u}^*$:

$$(P4) \qquad minimize \ \max_l\{\phi_l(u_l^* - f_l(\underline{x}))\}$$

s.t. (A1), (B1), (A2), (B2), (C)

where $\underline{\Phi} = (\Phi_1, ..., \Phi_k)$ is a weighting vector defining a "direction" of the Chebyshev norm, i.e. a line passing through an ideal and a nadir point, $\underline{u}^*$ and $\underline{u}_*$, respectively (cf. Fig.2).

The Chebyshev norm minimization chooses the "corner" closest to $\underline{u}^*$ and still in contact with the feasible region. This final point of contact ensures weak efficiency. In the case of the MLFP, (P4) can be transformed to a linear program with a single parameter. One can thus use any convenient univariate search method over this parameter to find
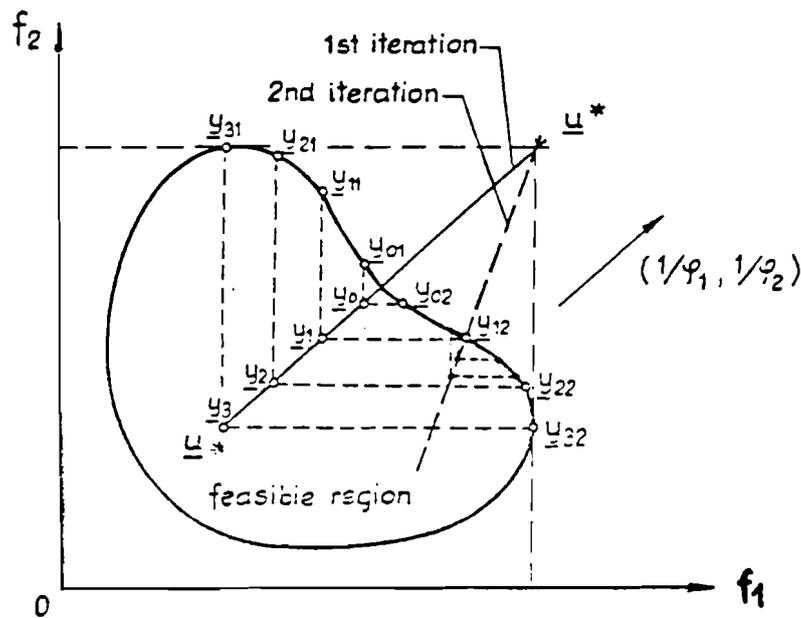
Fig.2. The idea of the interactive procedure by Choo and Atkins [1]

point $\underline{y}_0$ as close to efficient as we like. The "closeness" here is not critical and even a rough approximation will be quite sufficient. This point is taken as starting point for the interactive part of the algorithm. This part crucially involves the DM. The search direction from $\underline{u}^*$ to $\underline{y}_0$ is then extended to include several more points $\underline{y}_1, \underline{y}_2, ..., \underline{y}_p$. Next, taking each criterion in turn, say f_1 first, f_1 is maximized subject to all other criteria being at least equal to their values at $\underline{y}_0$, then $\underline{y}_1$, all the way to $\underline{y}_p$. This gives a sequence of (weakly) efficient points $\underline{y}_{01}, \underline{y}_{11}, ..., \underline{y}_{p1}$ for criterion 1 and $\underline{y}_{20}, \underline{y}_{21}$, etc. for criterion 2, and so on for all the criteria. The minimization of $f_l(\underline{x})$ for l=1,...,k is of a linear fractional subject to linear constraints, so the Charnes and Cooper transformation into a linear program can be used. Thus, at each step a single-objective linear programming problem has to be solved. The calculated sample of the efficient border is then presented to the DM who is asked to select the one that best fits his needs. The selected point becomes the new starting point and the procedure is repeated, but now the search space is focused on a part of the efficient border of most interest. The interactive process ceases when the most satisfactory efficient point is reached.

An important advantage of the above algorithm is that the only optimization procedure to be used is a linear programming one. Moreover, it has a simple scheme and allows retractions to the points which have been found uninteresting in previous iterations.

In a microcomputer implementation of FLIP, the efficient points proposed to the DM are presented both numerically, in terms of $\underline{x}$ and middle values of $\tilde{z}_l(\underline{x})$, l=1,...,k, and graphically, in terms of mutual positions of fuzzy numbers corresponding to objectives and aspiration levels on the one hand, and to left- and right-hand-sides of constraints, on the other hand (cf. [3], [4],[14]). In this way, the DM gets quite a complete idea about the quality of each proposed solution. The quality is evaluated taking into account:

● scores of fuzzy objectives in relation to the goals,

● dispersion of values of the fuzzy objectives due to uncertainty,

- safety of the solution or, using a complementary term, the risk of violation of the constraints (cf. the hatched areas in Fig.1).

So, the definition of the best compromise involves an analysis of the compromise among criteria and an evaluation of the safety of the corresponding solution.

It steems from the above considerations that the DM intervenes in two steps of FLIP. First, when fixing the safety parameters (step 3), and then in the course of the guided generation and evaluation of the efficient points of the associate deterministic problem (step 5). Thus, the interaction with the DM takes place at two levels. As to the first one (step 3), it is worth noting that there are two practical ways of controlling the safety of solutions using parameters τ_i , η_i and θ_i :

(a) fix $\eta_i = 0$, i=1,...,m, and control the optimistic safety with τ_i , and the pessimistic safety with θ_i , i=1,...,m, or

(b) fix $\theta_i = 0$, i=1,...,m, and control the optimistic safety with τ_i , and the pessimistic safety with η_i , i=1,...,m.

The safety parameters are defined taking into account their interval of variation and the knowledge acquired in preceding iterations about the dependency between safety and the quality of the compromise among criteria.

If way (a) is chosen, an (approximate) interval of variation of θ_i at level $\eta_i = 0$, i=1,...,m, can be calculated in the following way. Let the interval be denoted by $[\theta_i^L, \theta_i^U]$, and the set of feasible solutions by D.

It is clear that

$$\theta_i^U = \gamma_i + \delta_i \qquad\qquad \text{for i=1,...,m}_1 \qquad\qquad (22)$$

and

$$\theta_i^U = \max_{\underline{x} \in D}\{\underline{\alpha_i x} + \underline{\beta_i x}\} \qquad\qquad \text{for i=m}_1\text{+1,...,m} \qquad\qquad (23)$$

which corresponds to a perfect safety of solutions. The lower bound of the interval is defined as:

$$\theta_i^L = \min_{\underline{x} \in D}\{b_i + \delta_i - \underline{a_i x} - \underline{\beta_i x}\} \qquad\qquad \text{for i=1,...,m}_1 \qquad\qquad (24)$$

and

$$\theta_i^L = \min_{\underline{x} \in D}\{\underline{a_i x} + \underline{\beta_i x} - b_i - \delta_i\} \qquad\qquad \text{for i=m}_1\text{+1,...,m} \qquad\qquad (25)$$

In practice, an exact calculation of $[\theta_i^L, \theta_i^U]$, i=1,...,m, may lead to too large intervals with very low utility. The following procedure permits "killing two birds with one stone". One substitutes all fuzzy coefficients of problem (P1) by their middle values and calculate the pay-off table of the multiobjective deterministic problem obtained from (P1). It needs solving k single-objective linear programming problems. In this way, a set S of k solutions is obtained. The result can be used in two ways. First, one can substitute D by S in formulas (4.2),(4.3),(4.4). Then $[\theta_i^L, \theta_i^U]$, i=1,...,m, become approximate intervals having often a greater practical utility than exact ones. Second, the individual optima from the diagonal of the pay-off table may serve as suggestions for middle values of fuzzy goals. The columns of the pay-off table may also give an idea about the right spreads of the goals. This option is useful when the DM has no his own definition of the aspiration levels.

To conclude, let us remark that since its appearance, FLIP has been successfully applied to several real-world problems from the field of water supply systems ([15], [12]) and agriculture ([2], [4], [3]).

Part II
User's manual

5   Executive summary

The FLIP package supports the following general functions:

- the definition and edition of a source model in the form of a multiobjective linear programming problem with fuzzy coefficients; in this phase, particular attention is paid to modeling of fuzzy coefficients and their graphical representation.

- interactive analysis of the problem, with user-friendly graphical and numerical representation of generated solutions, and various facilities for comparison of those solutions in the process of searching for the best compromise.

The FLIP package is recorded on one diskette in a compiled code. After installing it in the user's directory, it can be activated by the command FLIP $< CR >$.

6   User's reference manual

6.1   Installing and activating the program

There are two main versions of FLIP package. One version requires a math coprocessor (8087,80287,80387) while another one does not. Each version is recorded on one diskette that should be installed on an IBM-PC XT/AT or a compatible computer preferably with a hard disc and with Hercules or color graphic card (CGA,EGA,VGA). The diskette contains the compiled code of FLIP with a test example.

While it is possible to use the program from floppy disc it is preferable to install it on a hard disc.

If we run the program from a floppy disc we must remember that on the distribution disc there is no room for big problems and for saving partial or final results. If you need more free space you can use another disc for data (change disc before the problem is created). System starts with command FLIP $< CR >$.

To install FLIP package on the hard disc make the following steps:

a. create a new user directory (e.g. md FLIP);

b. copy the contents of a distribution disc to this directory;

c. run FLIP program by the command: FLIP $< CR >$.

A moment after giving a starting command for FLIP package, the invitation screen is displayed (see Fig.3). Pressing any key we pass to main menu of the FLIP program.

6.2   Main menu

Main menu (Fig.4) provides the following functions:

Exit FLIP to leave the program and return to the operating system;

```
                        VERSION 5x50x50

                    F   L   I   P
           (Fuzzy LInear Programming)

                 This software has been developed by
                       P.Czyzak,R.Slowinski
                   Institute of Computing Science
                   Technical University of Poznan
          ul.Piotrowo 3A     60-965 Poznan     Poland
               within the scientific cooperation with
               the System and Decision Sciences Program of
          the International Institute for Applied Systems Analysis
                     A-2361 Laxenburg, Austria
                 This copy has been licensed to:
      ##################################################################
      ##################################################################
               to be used only for nonprofitable research or educational
             purposes. Any other use of this software requires written
          permision from the authors. This also include redistribution
                   FLIP uses GRADIENT LP of EURO - DECISION
   press any key
```

Fig.3 Invitation screen.

Data Input – to create a new problem and save it into a disc file;

Calculations – after activating this function, the fuzzy problem is transformed into a deterministic one and an interactive procedure starts to search for the best compromise solution;

Retrieving Previous Solutions – one can retrieve from disc files some series of previously got solutions;

Data Edit – with this function we can review and/or change both source model and definition of fuzzy aspiration levels;

Printing the Data – this function allows to get a copy of data file on a printer;

Choose an Active File – we can choose from directory a data file for further consideration;

Help – offers the information about FLIP method and how to make use of the FLIP program.

```
┌─────────────────────────────────────────────────┐
│              ▐Fuzzy Linear Programming Method▌     │
│                                                   │
│              Exit FLIP                            │
│                                                   │
│              ▐Data input▌                          │
│                                                   │
│              Calculation                          │
│                                                   │
│              Ready solution                       │
│                                                   │
│              Data editing                         │
│                                                   │
│              Printing the data                    │
│                                                   │
│              Choose an active file                │
│                                                   │
│              HELP                                 │
│                                                   │
│                          Active file:             │
│   Select item with▐SPACE▌/▐BACKSPACE▌, then hit▐ENTER▌│
└─────────────────────────────────────────────────┘
```
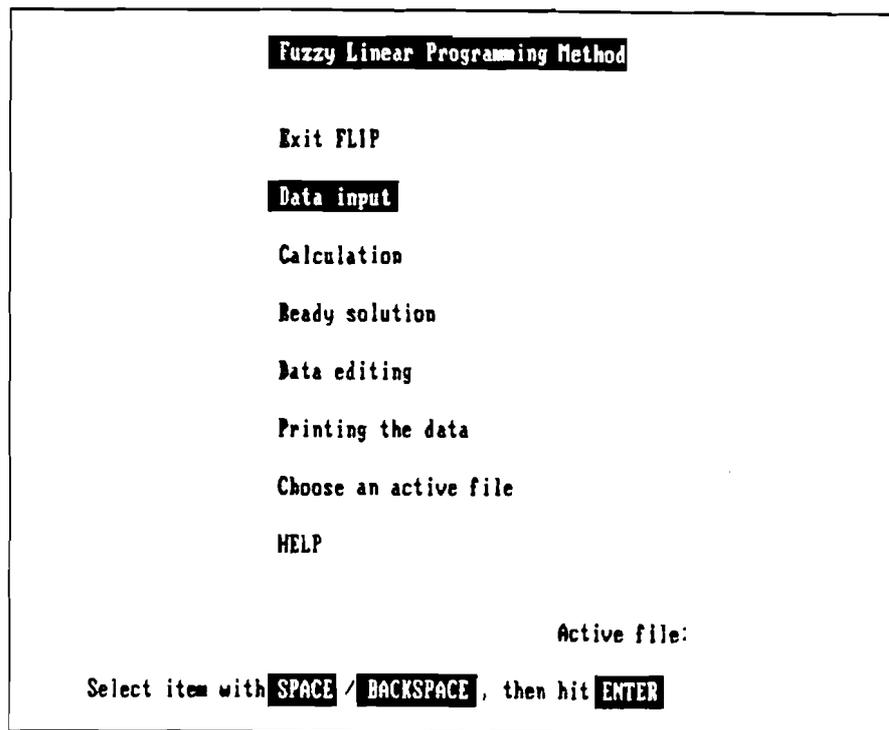
Fig.4 Main menu of the FLIP program.

Selection of any of these functions is made by moving a highlighted window to the required function and acceptance with $< CR >$ key.

Moving of the highlighted window can be performed in one of three ways:

- using arrow keys;

- pressing space bar or backspace;

- pressing appropriate number keys e.g. 0 for "Exit FLIP", 1 - for "Data Input", 2 for "Calculation" and so on.

6.3 Data input

This function allows to create a new disc file containing a new problem formulation. At the beginning, we enter the name of data file we create. Then, we define sizes of a new problem, that are: number of criteria, number of decision variables and number of constraints. We accept this values pressing function key - F10.

Now we pass to the definition of criterion number one. First, we choose a type of cost coefficients we want to use in this criterion. There are three alternatives:

real numbers – if we want to have nonfuzzy coefficients;

linear fuzzy numbers – if we want to define simple fuzzy numbers, giving only the middle value and left and right ranges of fuzzy number (see fig.5a);

piecewise linear fuzzy numbers – similarily to linear fuzzy numbers we have to define the middle value, left and right ranges and additionally to construct left and right reference functions consisting of maximum three linear pieces each (Fig.5b).
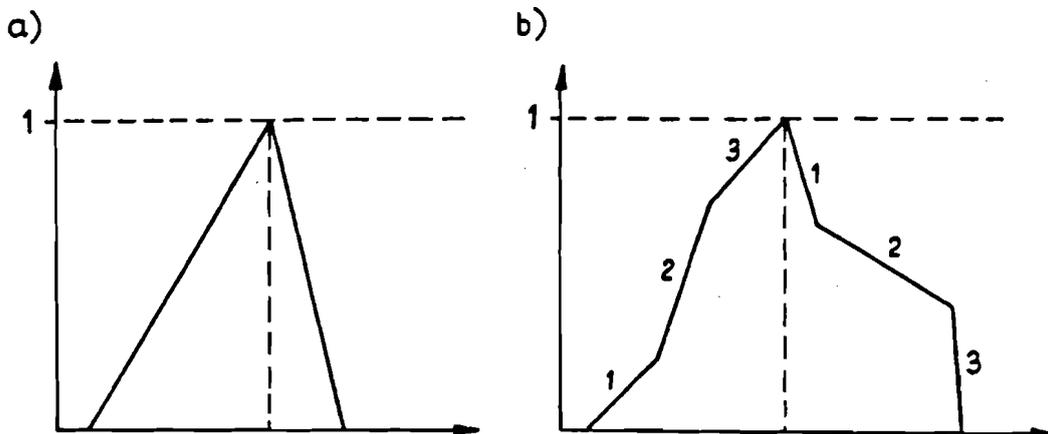

Fig.5 Definition of a fuzzy number.

In the next step, we declare if we want to maximize or to minimize the considered criterion.

When choosing the type of coefficient and maximization or minimization, we move a highlighted window using Space or Backspace keys and then we accept our choice with function key F10.

Before explaining in detail how to built two types of fuzzy numbers, we must make some important remarks.

In the definition of the fuzzy number (see chapt. 2), we spoke about left and right "spreads" of the fuzzy number; the left "spread" means a distance between the middle value and the left range of the fuzzy number and the right "spread" means a distance between the middle value and the right range of the fuzzy number. In our implementation of FLIP, we don't use the values that express the "distance", but we give directly the real values of the left and right ranges of the fuzzy number. It means that we give values of abscissae of the appropriate points.

Next, we start defining the successive coefficients. At this moment, on the monitor screen we have:

- general form of the considered criterion with a highlighted coefficient which can be introduced or changed now;

- two windows containing an information about the highlighted coefficient: in the left window, numerical representation of the coefficient is displayed and in the right window, the graphical representation, i.e. the shape of the membership function.
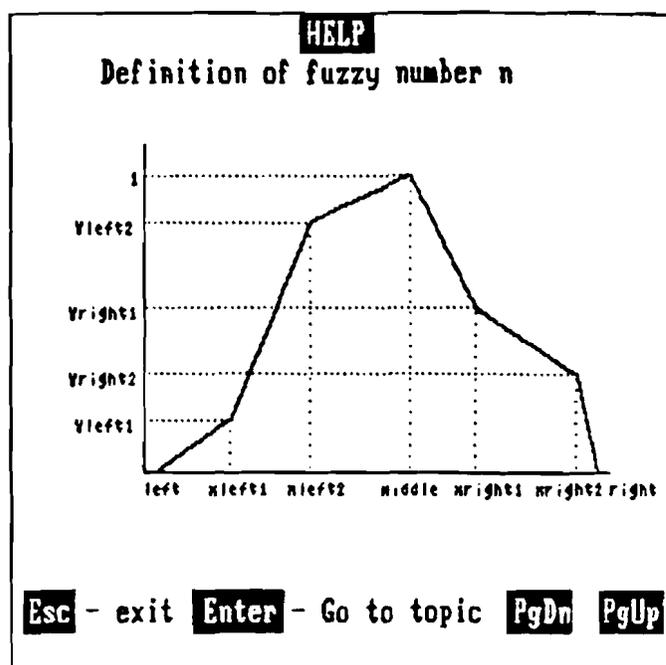
Fig.6 Definition of piecewise linear fuzzy number (Help).

The menu is displayed in the right bottom corner of the screen, offering the following functions:

number $< ENTER >$ data input;
    if we define coefficient of type:

- real number, we type a value of the coefficient and press ENTER;

- linear fuzzy number, we type first the middle value of the coefficient, i.e. the "most possible" and press ENTER; then, we type values of left an right ranges of the fuzzy number; We must remember that:
left range $\leq$ middle value $\leq$ right range.

- piecewise linear fuzzy number (Fig.6), we first type the middle value, left and right ranges such that again:
left range $\leq$ middle value $\leq$ right range.
If the left range $<$ the middle value, we define the left reference function in the following steps:

    a. We are asked about the value of Xleft1. This is the abscissa of the point ending the first linear piece and beginning the second. If we press ENTER without any number, we resign from construction of the left reference function, otherwise, we type value for XLeft1 such that:
left range $\leq$ XLeft1 $\leq$ middle value
and then, we define YLeft1 *in* [0,1], that is value of the ordinate of the first breakpoint.

    b. In the same way, we define the second breakpoint [XLeft2,Yleft2] ending the second linear piece and beginning the third such that:
XLeft1 $\leq$ XLeft2 $\leq$ middle value and YLeft1 $\leq$ YLeft2 $\leq$ 1.

If we press "ENTER" without any number, the left function will consist of two linear pieces only.

Next, if the middle value < right range, we start definition of the right reference function. We have to proceed it in the following steps:

a. We are asked about the value of XRight1. If we answer by "ENTER", we end the definition of the fuzzy number, otherwise, we define the first breakpoint $[XRight1, YRight1]$ such that:
middle value $\leq$ XRight1 $\leq$ right range
and YRight1 *in* [0,1].

b. We define the second breakpoint [XRight2,YRight2] of the right reference function. If we don't type any value for XRight2 and press "ENTER", we have the second linear piece between points [XRight1,YRight1] and [right range,0].

next coefficient – pressing right arrow key we can move to the next coefficient (right from presently highlighted coefficient), if the last coefficient is presently pointed we moved back to the first coefficient of the criterion or to the RHS coefficient of the constraints;

previous coefficient – pressing left arrow key we can move to the previous coefficient (left from presently highlighted coefficient); if we are at the first coefficient, then we go back to the last coefficient of the criterion or to the RHS coefficient of the constraint;

Choose coefficient – this function allows to jump to a selected coefficient; after pressing "C" we are asked about the number of the coefficient we want to consider; this function can be very useful for problems with a great number of decision variables and a low density of the matrix coefficients.

F1 (help) – in particular, we can see a general definition of a fuzzy number; it can be helpful to understand how to define fuzzy numbers;

F10 next criterion or

F10 next constraint – with this function we end definition of the presently displayed criterion or constraint, and we go to the next criterion or constraint; if it was already the last constraint, we pass to the definition of fuzzy aspiration levels.

The procedure described above is just the same for criteria and constraints, with an obvious exception that for constraints we don't have maximization or minimization.

"Input Data" procedure ends with definition of a fuzzy aspiration level for every criterion. To support the decision maker the individual optima are calculated for every criterion under assumption, that fuzzy coefficients become crisp ones, equal to middle values of the corresponding fuzzy numbers. The calculated optimum is displayed on the screen as a suggestion. The decision maker chooses the type of an aspiration level (real number or fuzzy number - with linear or piecewise linear membership function) and then defines the value of the aspiration level (Fig.7).
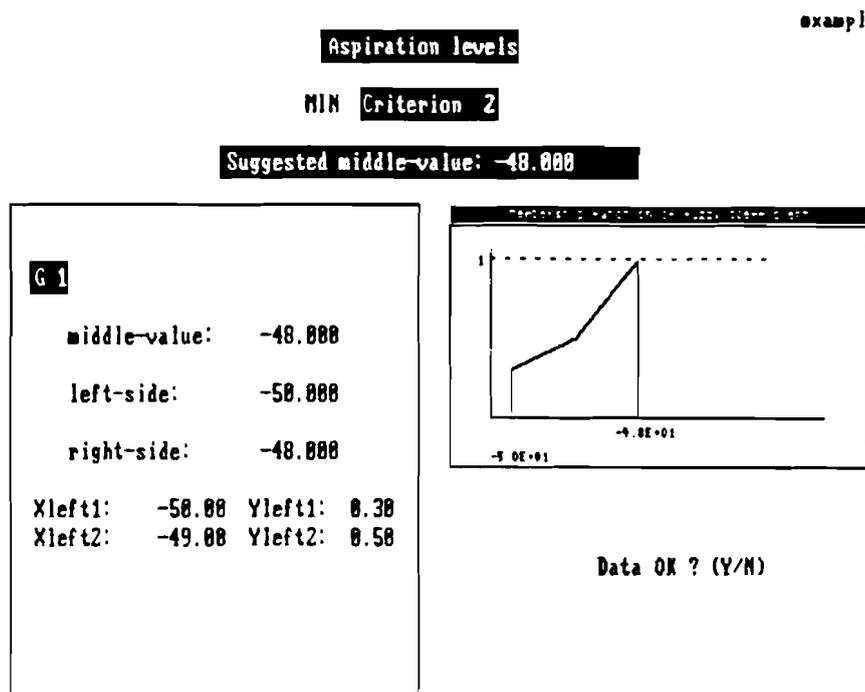
Fig.7 Definition of an aspiration level.

6.4   Data editing

This function gives a possibility of reviewing and changing a problem created using function "Data Input".

If an active file has been already choosen (with function "Choose an Active File" or other functions, e.g. "Data Input", "Calculations"), then, the active file will be edited, otherwise, the directory with data files will be displayed and we have to choose a data file we want to edit.

Then, an edit menu is displayed on the screen, which consists of ten functions (Fig.8). At the bottom of the screen the sizes of the analyzed problem are given.

Now we have at our disposal the following menu:

Return to main menu – with this function we can go back to the main menu of the FLIP program. If we have performed any editing function on an active file we are asked if we want to save the problem. If we don't save the changes they are lost and the source active file remains with no change.

Write the current problem to disc storage – this function allows, at any moment, to save the currently edited problem with all changes we have made up to now. It is of course possible to change the name of the data file storing the changed problem and; if we do so, the source data file remains unchanged.

Add more criteria to the current problem – using this function, we can add new criterion to the current problem. The procedure of defining a new criterion is just the same as in "Data Input" function. So we define consequently: type of coefficients, maximization or minimization and nonzero coefficients. After accepting definition of the new criterion by pressing F10 we are asked if we want to define next criterion; if not, we return to edit menu.

Add more constraints to the current problem – with this function we can add new constraints to the current problem. The way of entering new constraints is the same as for criteria.

Add more variables to the current problem – it allows to increase the number of decision variables in the current problem. First, we are asked "How many variables do you want to add ?". If we answer "0", we return to edit menu, otherwise, the new variables are added to the problem and we can define coefficients corresponding to the new variables. New variables are added as the last variables.

Delete some criterion from the current problem – with this function we can remove some criterion from the current problem. If we choose this function, we examine all criteria and point out the criteria which we want to delete.

Delete some constraints from the current problem – it allows to delete any constraint in a similar way to the previous function;

Delete some variables from current problem – we are asked about the number of variable we want to delete. If we answer "0" then no variable is deleted.

Examine and/or change the current problem – it is very useful function that allows reviewing and changing the current problem. We have at our disposal the same menu as when entering the data (see chapt. 2). Our reviewing/changing procedure starts from the first criterion. We can move along the coefficients using left and right arrow keys or with function "Choose coefficient". We can change any coefficient. Pressing F10 we skip to the next criterion or constraint.

Examine and/or change aspiration levels – this function gives the possibility of changing or reviewing aspiration levels. Numerical and graphical presentation of the current state of aspiration levels starts from the first criterion. The DM is asked, if he wants to change the value of the aspiration level. If answer is "No" then procedure goes to the next aspiration level of the next criterion, otherwise, the following procedure is realized:

    a. individual optimization problem is solved for the considered criterion with all fuzzy coefficients fixed on their middle value and a suggestion about possible value of the aspiration level is displayed,

    b. the DM chooses the type of the aspiration level: real number or fuzzy number (linear or piecewise linear membership function),

    c. the DM types a new value of the aspiration level and the program goes to the next criterion.

```
┌─────────────────────────────────────────────────────────┐
│                        ▐SETUP MENU▌                        │
│                                                           │
│          Return to main menu                              │
│                                                           │
│          ▐Write the current problem to disk storage▌      │
│                                                           │
│          Add more criteria to the current problem         │
│                                                           │
│          Add more constraints to the current problem      │
│                                                           │
│          Add more variables to the current problem        │
│                                                           │
│          Delete some criterion from the current problem   │
│                                                           │
│          Delete some constraints from the current problem │
│                                                           │
│          Delete some variables from the current problem   │
│                                                           │
│          Examine and/or change the current problem        │
│                                                           │
│          Examine and/or change aspiration levels          │
│                                                           │
│     Select item with▐SPACE▌/▐BACKSPACE▌, then hit▐ENTER▌  │
│                                                           │
│      ____Size of PLAN3: 5 Crit.,   4 Constr.,  7 Dec.var.____  │
└─────────────────────────────────────────────────────────┘
```

Fig.8 Data editing menu.

6.5   Calculation

Calculation procedure can be divided into three parts:

- transformation of the fuzzy problem into an associate deterministic fractional linear programming problem;

- searching for a starting point of an interactive procedure;

- interactive search for the best compromise solution.

If the active file isn't defined till now, the calculation procedure starts from choosing a data file from a directory.

6.5.1   Transformation of the fuzzy problem

Transformation of the fuzzy problem into an associate deterministic fractional linear programming problem is based on conception of comparison of fuzzy numbers proposed by Słowiński [11] (see chapt. 3).

At the beginning, we have to define safety parameters: τ_i, η_i and θ_i (i=1,...,m) for fuzzy constraints. In our implementation of FLIP, in order to support the DM, standard

values are proposed for safety parameters: τ_i =0.6, η_i =0, and θ_i equal to the middle value of an approximate range of its variation for i=1,...,m.

The DM is asked if he accepts this proposal or not. If not, these values can be changed in two ways:

- simultaneously for all fuzzy constraints;

- independently for each fuzzy constraint.

After fixing the final value for safety parameters, the fuzzy problem is transformed as it was shown in chapt. 3.

Calculation   DXl

Data Loading

Calculation of the pay-off table to the DMFLP problem

Calculation of the starting point

   0.000000000E+00

STARTING POINT FOUND

press any key

Fig.9 Search for a starting point.

6.5.2   Searching for a starting point

The second step of calculation procedure consists in searching for a starting point of the interactive method by Choo and Atkins. First, utopia and nadir points are calculated, (the following information is displayed on the screen: "Pay-off table is calculated") and then the procedure looks for a point lying in the proximity of an efficient border on the line connecting utopia and nadir points. This search consist in solving a series of LP problems and it can take some time, particularly, for big problems. During this

calculation, information about the distance between current solution and the efficient border is displayed.

It may happen that certain LP problems are contradictory (ERROR 2). In this situation, we should try to change safety parameters or, eventually, reformulate the analyzed problem. When the starting point will be found, an appropriate information will be displayed on the screen, together with the order "press any key" (Fig.9). After pressing a key, we pass to the most important, interactive phase of searching for the best compromise solution.

6.5.3   Interactive phase

This phase starts with a question to the DM how many compromise solutions he wants to get. At this moment possible numbers of solutions appear on the screen; that are, obviously, multiples of the number of criteria. The DM chooses the number of compromise solutions by moving a highlighted window (using arrow keys) and accepting the choice by ENTER. Then, we can see, that single objective optimizations are performed and, finally, we get required number of efficient solutions (points) to be analyzed.

So, we can pass to the stage of comparison of compromise solutions and searching for the best one. The obtained solutions are presented both graphically and numerically. We have on the screen four windows with graphical representation of fuzzy criterion, constraints and values of decision variables (Fig.10).

In this moment, the DM has at his disposal the following menu:

arrow keys – change active window (highlighted) - functions F1 to F4 can be performed in the active window;

F1 next display;

F2 previous display;

F3 setting display in the active window; allows to see selected criterion or constraint or variables for a given solution (point);

F4 enlarge/reduce the active window;

F5 numerical display of middle values of criteria; it gives to the DM the first impression about the generated compromise solutions (Fig.11);

F7 gives the options:

> **C ontinue** to restart review of the generated set of compromise solutions.

> **N ew starting point** to point the most interesting solution at the current stage which becomes a new starting point for the next iteration. In the next iteration, a new set of compromise solutions is generated from the neghbourhood of the starting point.

> **P revious solution** to come back to review of solutions generated in the previous iteration.

> **S ave solution** to save the currently generated set of compromise solution in a disc file. Using function "Retrieving previous solutions" from Main Menu, we can restore these solutions and continue their review.
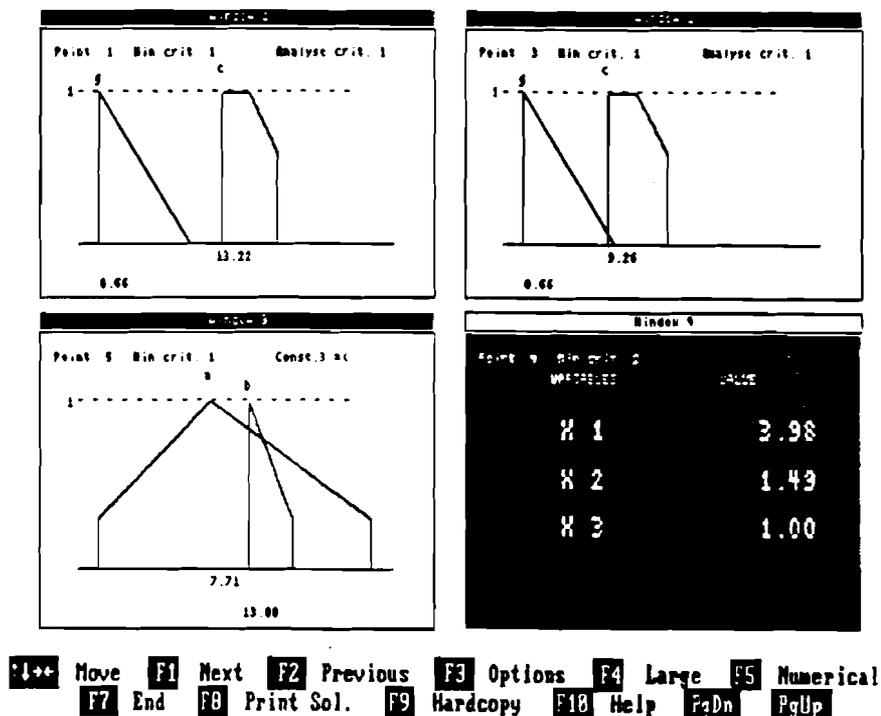
Fig.10 Comparison of compromise solutions.

E nd of survey to return to Main Menu.

F8 print selected solution; we can get on a printer numerical information - values of criteria and decision variables - for selected solution.

F9 screen copy on the printer; this function gives a hardcopy with a present contents of the monitor screen.

6.6 Retrieving previous solutions

This function displays previously saved solutions for a selected problem. We can examine a chosen set of solutions and, possibly, start generation of a new set of solutions. So, we pass to the interactive phase (described above) where a number of options is offered.

6.7 Printing the data

If no file is active we have to choose an active file. Then the DM is asked about the state of a printer. If the printer is ready then, after pressing any key, we get a copy of source data file on a printer.

Middle values of criteria   ΣΧΊ

|Points|	1		2		3		4		5		6		7		8		9	
|Crit.1|	13.2		13.8		18.8		15.4		8.5		17.8		6.1 !		2.2		3.7	
|Crit.2|	5.1		5.2		5.8		4.6		6.4		3.9		7.8		3.3		7.7	

|Points|	18		11		12	
|Crit.1|	22.6		1.3		24.9	
|Crit.2|	2.7		8.3		2.8	

Fig.11 Middle values of compromise solutions.

7   User's training manual

7.1   General form of the problem to be solved

FLIP package allows solving the MOLP problem with fuzzy coefficients which can be written in the following general form:

$$minimize \begin{bmatrix} \tilde{z}_1 = \tilde{\underline{c}}_1 \underline{x} \\ \vdots \\ \tilde{z}_k = \tilde{\underline{c}}_k \underline{x} \end{bmatrix}$$

$$s.t. \quad \begin{array}{ll} \tilde{\underline{a}}_i \underline{x} \leq \tilde{b}_i & i = 1, ..., m_1 \\ \tilde{\underline{a}}_i \underline{x} \geq \tilde{b}_i & i = m_1 + 1, ..., m \\ \underline{x} \geq \underline{0} \end{array}$$

where $\tilde{\underline{c}}_l$ (l=1,...,k) is a row vector of fuzzy cost coefficients of objective l, $\tilde{\underline{a}}_i$ is the i-th row of the matrix of fuzzy left-hand-side coefficients and $\tilde{b}_i$ is a right-hand-side coefficient of the i-th constraint (i=1,...,m).

To complete the above problem formulation, the decision maker must be able to define fuzzy aspiration levels thought of as goals, denoted by $\tilde{g}_1$ to $\tilde{g}_k$.

All coefficients, denoted with ˜, and aspiration levels can be defined in one of three following ways:

- as real numbers;

- as linear fuzzy numbers - defined by three real numbers: middle value, left and right range;

- as piecewise linear fuzzy numbers - defined by maximum 11 real numbers:middle value, left and right range and, eventually, $[Xleft1, Yleft1]$, $[Xleft2, Yleft2]$, $[Xright1, Yright1]$, $[Xright2, Yright2]$.

7.2   Illustrative example

Let us consider the following simple MOLP problem with fuzzy coefficients:

$$minimize \begin{bmatrix} \tilde{z}_1 = \tilde{c}_{11}x_1 + \tilde{c}_{12}x_2 \\ \tilde{z}_2 = \tilde{c}_{21}x_1 + \tilde{c}_{22}x_2 \end{bmatrix}$$

$$s.t. \quad \begin{array}{rcrcl} -2x_1 & + & 2x_2 & \leq & 12 \\ 3x_1 & + & x_2 & \leq & 25 \\ x_1 & & & \leq & 7 \\ \tilde{a}_{41}x_1 & + & \tilde{a}_{42}x_2 & \leq & \tilde{b}_4 \end{array}$$

$$x_1, x_2 \geq 0$$
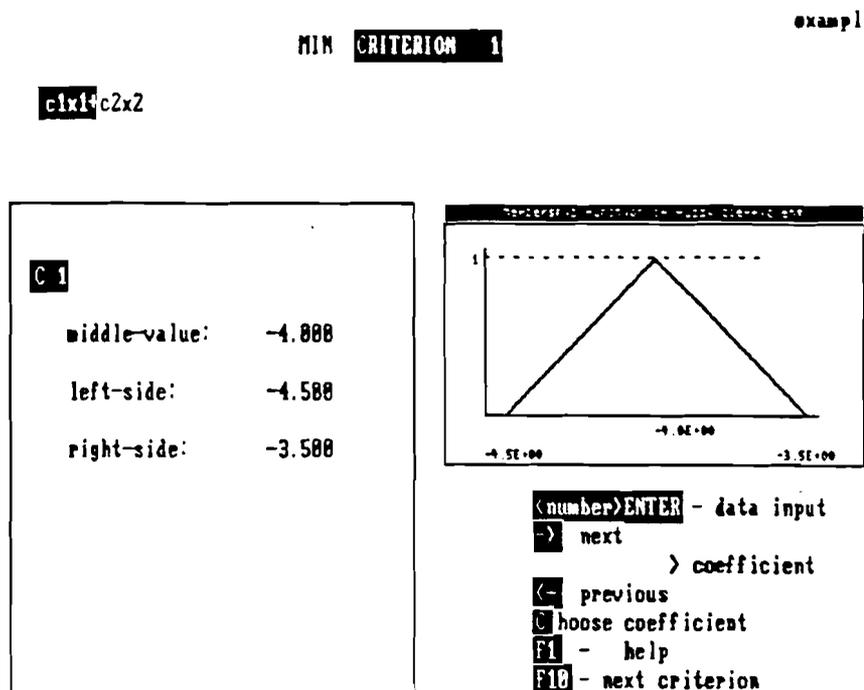
Fig.12 Definition of coefficient $\tilde{c}_{11}$.

where $\tilde{c}_{11}$=(-4,-4.5,-3.5)$_{LR}$, $\tilde{c}_{12}$=(1,1,1.5)$_{LR}$, $\tilde{c}_{21}$=(2,2,2.5)$_{LR}$, $\tilde{c}_{22}$=(-6,-6.5,-5.5)$_{LR}$, $\tilde{a}_{41}$=(2,1.5,2)$_{LR}$, $\tilde{a}_{42}$=(3,2.8,3.5)$_{LR}$, $\tilde{b}_4$=(33,32,36)$_{LR}$ and function L, R are linear.

The above problem is introduced to the program using the "Data Input" function. In criteria no.1 and 2 and in constraint no.4, we use fuzzy numbers with their linear form (Fig.12).

Then aspiration levels are defined. On the monitor screen the suggestion is displayed: '-28' for criterion no.1 and '-48' for criterion no.2. The decision maker takes as aspiration levels the following fuzzy numbers (Fig.13):
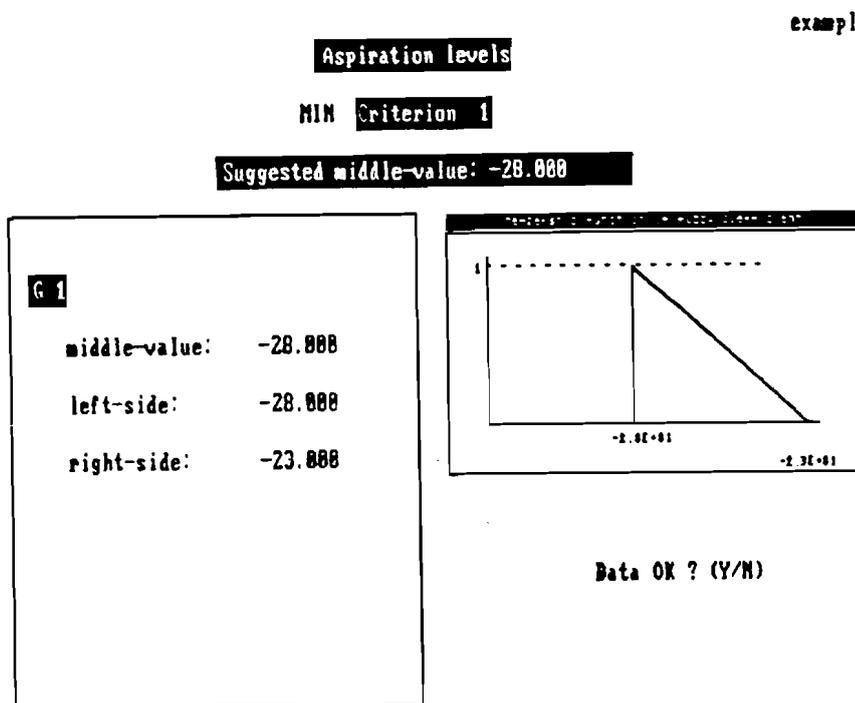
Fig.13 Definition of aspiration level $\tilde{g}_1$.

$$\text{for } z_1 - \tilde{g}_1 = (-28,-28,-23) \; _{LR}$$
$$\text{for } z_2 - \tilde{g}_2 = (-48,-48,-18) \; _{LR}$$

At this moment the data input is completed and the new problem is saved into a disc file. After returning to the Main Menu of FLIP we choose the function "Calculation".

At the beginning, we define safety coefficients that allow to transform fuzzy problem into deterministic one. The program makes a proposal: $\tau=0.6$, $\eta=0$ and $\theta=0.85$. The value proposed for θ is a middle value of approximated range of variability for θ_4 that is [-1.5,3]. The decision maker accepts this values without any changes.

Next, the fuzzy problem is transformed into a deterministic problem according to the rules explained in part I. Then, an appropriate pay-off table is calculated and the search for a starting point begins. When it is completed, we can pass to the interactive part of looking for the best compromise solution. Let l_j denote the l-th solution obtained in the j-th iteration and S, the number of compromise solutions required.

Let's take S=8. In the first iteration, we have got a series of 8 compromise solutions for which middle values of criteria are presented in Tab.1.

TABLE 1.

	1_1	2_1	3_1	4_1	5_1	6_1	7_1	8_1
z_1	-18.2	-18	-23.2	-11.7	-25.9	-6.3	-28	-1.8
z_2	-26.5	-27.1	-12	-34.6	1.5	-41	14.3	-46.4

When comparing them, we come to conclusion that the most interesting solutions are solutions 1_1 , 2_1 and 4_1, because they are not far from aspiration levels.

Fig.14 shows a mutual position of particular criteria and their aspiration levels for solution 4_1 - upper windows - and for solution 2_1 -lower windows. Let's suppose that we want to improve the value of criterion 1 (left windows). We choose solution 2_1 as a new starting point for the second iteration.
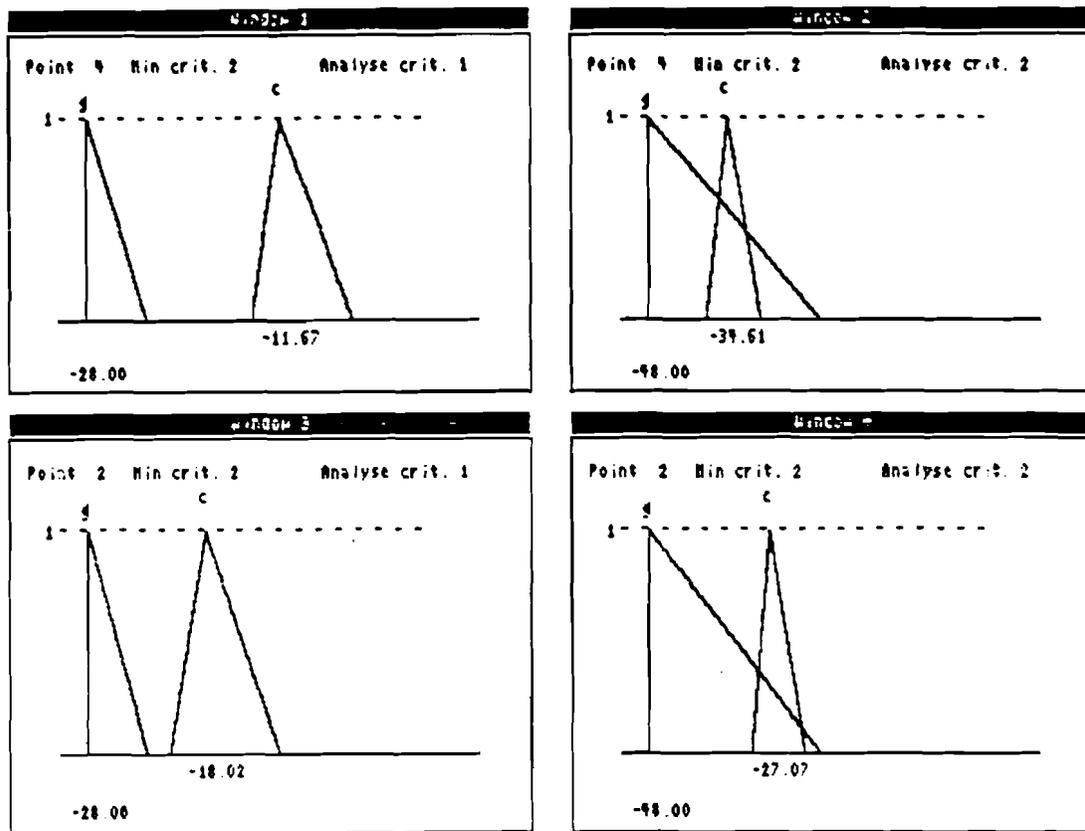
Fig.14 Values of criteria for solutions 4_1 and 2_1 .

For S=10 we obtain solutions listed in Tab. 2.

TABLE 2.

	1_2	2_2	3_2	4_2	5_2	6_2	7_2	8_2
z_1	-18	-18	-19.9	-15.5	-21.8	-13.1	-23.6	-10.9
z_2	-27.1	-27.1	-21.6	-30.1	-16.3	-32.9	-11.2	-35.5

	9_2	10_2
z_1	-24.6	-8.8
z_2	-6.2	-38

As we see, the most interesting solutions from the viewpoint of criterion 1 are solutions 3_2, 5_2, 7_2 and 9_2. Among those four solutions only solution 3_2 has a satisfactory value of criterion 2 (see Fig.15, upper window).
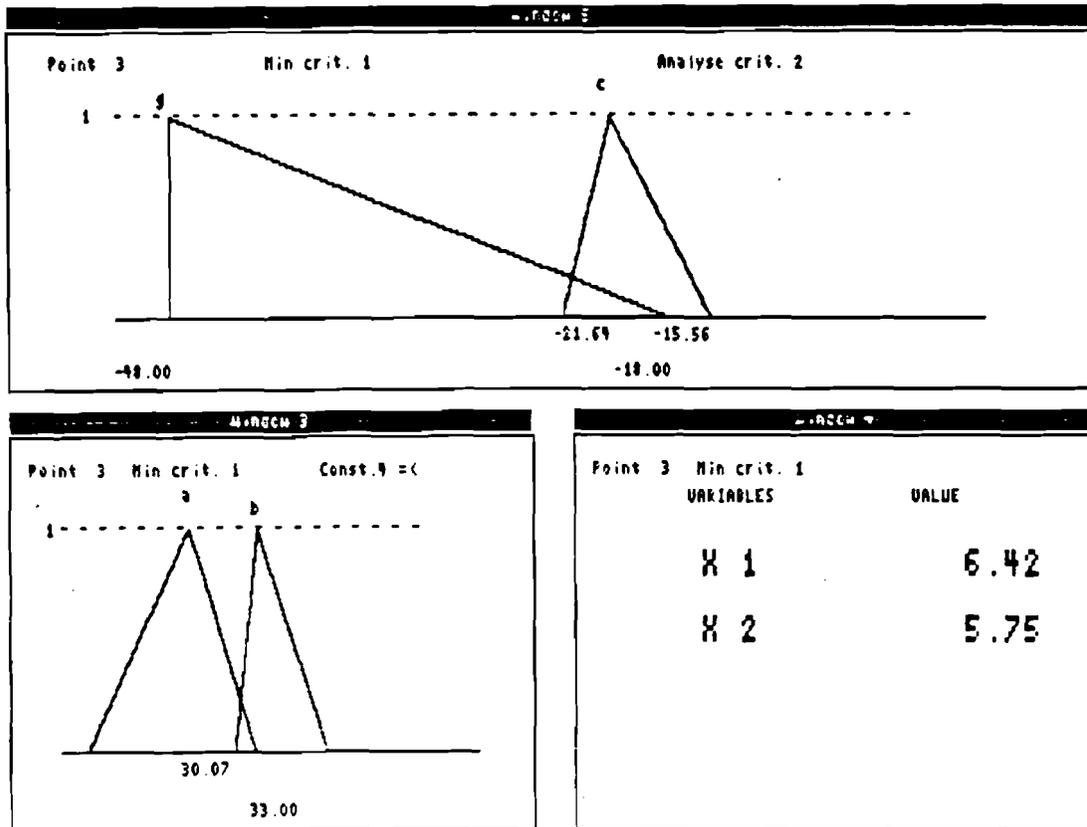
Fig.15 Solution 3_2

To accept this solution as the best compromise solution, we must analyze the state of the fuzzy constraint for this solution. By the state we mean information about a risk of violation of the constraint.

In the lower, left window in Fig.15, we present the state of constraint no. 4. As we can see, there is no risk of violation of this constraint.

So finally, we choose solution 3_2 as the best compromise solution with following values of decision variables (Fig.15):

$$x_1 = 6.42 \text{ and } x_2 = 5.75.$$

Part III
Solution of a real farm structure optimization problem

Let us consider a real-life problem of searching for the best structure of a typical Polish private farm. The considered farm has 20 hectares (ha) of arable land and 4 ha of permanent grassland. The farmer possesses 6 sows and 4 cows. We take into account 26 activities which can be divided into 5 groups:

- plant production for sale: winter wheat, winter barley, triticale, spring wheat, spring barley, rape, pea, potato, sugar beet;

- plant production for fodders consumed in the farm: winter barley, spring barley, triticale, spring wheat, potatoes, fodder beet, lucerne, clover, corn;

- permanent grassland cultivation;

- purchase of fertilizers: phosphorus, nitrogen, potassium;

- purchase of a nutritive fodder;

- manpower hire: in the spring, summer and autumnal period.

All these activities correspond to decision variables $x_1, x_2, ..., x_{26}$ defining their dimension, e.g. number of hectares of winter barley for sale, number of kilograms of phosphorus to be bought, number of hours of manpower hire in the spring period.

We take into consideration the following constraints:

- balance of arable land;

- 2 balances of crop succession: for spring crops and for rape;

- 3 constraints on the area of group of plants: for crops, for winter crops and for root crops;

- 3 balances of manpower: for spring, summer and autumnal periods of extend manpower demand;

- 3 balances of artificial fertilizers: for phosphorus, nitrogen and potassium;

- 4 balances of fodders: for green fodder, for potatoes, for fodder beet and for nutritive fodder;

- balance of permanent grassland.

The matrix of constraints is presented in Tab.3. Definition of fuzzy coefficients is given in Tab.4.

TABLE 3. Set of constraints.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
1	1	1	1	1	1	1	1	1	1	1	1	1	1
			-1	-1				1		-1		-1	
			1		-1					1			
1	1	1	1	1					1	1	1	1	1
								1	1				
1	1	1							1		1		
$\tilde{a}_1^7$	$\tilde{a}_2^7$	$\tilde{a}_3^7$	$\tilde{a}_4^7$	$\tilde{a}_5^7$	$\tilde{a}_6^7$	$\tilde{a}_7^7$	$\tilde{a}_8^7$	$\tilde{a}_9^7$	$\tilde{a}_{10}^7$	$\tilde{a}_{11}^7$	$\tilde{a}_{12}^7$	$\tilde{a}_{13}^7$	$\tilde{a}_{14}^7$
$\tilde{a}_1^8$	$\tilde{a}_2^8$	$\tilde{a}_3^8$	$\tilde{a}_4^8$	$\tilde{a}_5^8$	$\tilde{a}_6^8$	$\tilde{a}_7^8$	$\tilde{a}_8^8$	$\tilde{a}_9^8$	$\tilde{a}_{10}^8$	$\tilde{a}_{11}^8$	$\tilde{a}_{12}^8$	$\tilde{a}_{13}^8$	$\tilde{a}_{14}^8$
$\tilde{a}_1^9$	$\tilde{a}_2^9$	$\tilde{a}_3^9$	$\tilde{a}_4^9$	$\tilde{a}_5^9$	$\tilde{a}_6^9$	$\tilde{a}_7^9$	$\tilde{a}_8^9$	$\tilde{a}_9^9$	$\tilde{a}_{10}^9$	$\tilde{a}_{11}^9$	$\tilde{a}_{12}^9$	$\tilde{a}_{13}^9$	$\tilde{a}_{14}^9$
160	140	120	120	120	180	20	120	160	10	15	35	15	160
120	120	120	100	100	120	100	120	120	120	100	120	100	120
100	100	90	80	80	180	80	120	120	90	80	90	80	120
								300					
													250
									45	40	43	40	

x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	sg.	RHS
1	1	1	1									$=$	20
1			1									$\leq$	0
												$\leq$	0
												$\geq$	14
1												$\geq$	5
												$\geq$	10
$\tilde{a}_{15}^7$	$\tilde{a}_{16}^7$	$\tilde{a}_{17}^7$	$\tilde{a}_{18}^7$	$\tilde{a}_{19}^7$					-1			$\geq$	$\tilde{b}_7$
$\tilde{a}_{15}^8$	$\tilde{a}_{16}^8$	$\tilde{a}_{17}^8$	$\tilde{a}_{18}^8$	$\tilde{a}_{19}^8$						-1		$\geq$	$\tilde{b}_8$
$\tilde{a}_{15}^9$	$\tilde{a}_{16}^9$	$\tilde{a}_{17}^9$	$\tilde{a}_{18}^9$	$\tilde{a}_{19}^9$							-1	$\geq$	$\tilde{b}_9$
160	30	30	200	200	-1							$\geq$	0
120	120	120	140	120		-1						$\geq$	0
120	100	100	120	100			-1					$\geq$	0
	400	350	550	350								$\leq$	3000
												$\leq$	480
900												$\leq$	450
								1				$\leq$	360
				1								$=$	4

TABLE 4. Fuzzy coefficients of constraints.

$\tilde{a}_j^i = (a_j^i, \alpha_j^i, \beta_j^i)$								
j	a_j^7	α_j^7	β_j^7	a_j^8	α_j^8	β_j^8	a_j^9	α_j^9	β_j^9
1	15	1	1	40	2	2	35	2	3
2	15	1	1	40	2	2	10	1	1
3	15	1	1	40	2	2	35	2	3
4	25	2	3	40	2	2	15	1	1
5	25	2	3	40	2	2	15	1	1
6	30	3	3	90	4	5	10	1	1
7	30	2	3	75	3	3	20	2	3
8	110	3	5	15	1	1	120	4	5
9	80	4	4	20	2	2	160	5	6
10	30	2	3	40	2	3	10	1	1
11	25	2	2	40	2	3	15	2	2
12	15	1	1	40	2	3	35	2	3
13	25	2	3	40	2	3	15	1	1
14	110	3	5	15	1	1	120	4	4
15	80	2	3	20	2	2	160	5	6
16	30	2	2	34	2	3	10	1	1
17	30	2	2	34	2	3	10	1	1
18	25	2	2	10	1	1	60	3	3
19	10	1	1	40	2	3	40	2	2
$\tilde{b}_i = (b_i, \gamma_i, \delta_i)$								
b_7	γ_7	δ_7	b_8	γ_8	δ_8	b_9	γ_9	δ_9
600	0	50	580	30	30	930	0	90

TABLE 5. Fuzzy cost coefficients.

$\tilde{c}_i^s = (c_i^s, \lambda_i^s, \rho_i^s)$, s=1 for i¡10, s=2 for i $\geq$ 10						
i	c_i^s	λ_i^s	ρ_i^s	i	c_i^s	λ_i^s	ρ_i^s
1	500	0	20	14	780	30	0
2	400	0	20	15	300	20	0
3	380	0	16	16	140	5	0
4	350	0	20	17	150	10	0
5	405	0	20	18	250	10	0
6	565	0	25	19	100	5	0
7	500	0	30	20	0.1	0.01	0
8	1400	0	50	21	0.75	0.25	0
9	600	0	30	22	0.04	0.02	0
10	180	20	0	23	13	2	0
11	170	10	0	24	0.8	0.1	0
12	180	10	0	25	0.9	0.2	0
13	175	10	0	26	1	0.2	0

Three criteria are used to evaluate solutions:
- **gross profit**

$$f_1 = \tilde{c}_1^1 x_1 + ... + \tilde{c}_9^1 x_9 - \tilde{c}_{10}^2 x_{10} - ... - \tilde{c}_{26}^2 x_{26} + \sum_{s=1}^{S} \tilde{T}_s h_s$$

where fuzzy cost coefficients are presented in Tab.5 and constant $\sum_{s=1}^{S} \tilde{T}_s h_s$ is equal to (22000,0,0).
- **structure-forming plants area**

$$f_2 = c_6^3 x - 6 + c_7^3 x_7 + c_{16}^3 x_{16} + c_{17}^3 x_{17}$$

where $c_6^3 = 0.5$, $c_7^3 = c_{16}^3 = c_{17}^3 = 1$.
- **manpower hire**

$$f_3 = c_{24}^4 x_{24} + c_{25}^4 x_{25} + c_{26}^4 x_{26}$$

where $c_{24}^4 = c_{25}^4 = c_{26}^4 = 1$. After entering the data we get individual optima for the criteria that are calculated with fuzzy coefficients of the problem fixed on their middle values:

$$\text{Max } \tilde{f}_1 = 20716,$$

$$\text{Max } \tilde{f}_2 = 17.580,$$

$$\text{Min } \tilde{f}_3 = 180.52.$$

According to the farmer suggestion, we take the following aspiration levels[1]:

$$\text{for } f_1 - \tilde{g}_1 = (20716,1716,0),$$

$$\text{for } f_2 - \tilde{g}_2 = (4, 0.5, 0.5),$$

$$\text{for } f_3 - \tilde{g}_3 = (500, 50, 50),$$

Now we must define safety coefficients. Let's take them in conformity with program suggestion $\tau_i = 0.6, \eta_i = 0$ and $\theta_i = 16.7$ for i=1,....,m. Approximate ranges of variation of θ_i are: $\theta_7 \in$ [-20.1,50],$\theta_8 \in$ [-47.7,20] and $\theta_9 \in$ [13.4,90].

Then, the fuzzy problem is transformed into the deterministic one. For this new problem the pay-off table is calculated and a starting point of the Choo-Atkins method is searched for. The first starting point is y_0=(-2.64,13.65,-0.72).

Now we start with examination of the efficient border of the transformed problem looking for the best compromise solution. In each iteration, the number of compromise solutions S to be compared must be a multiple of the number of criteria. Let's l_j denote the l-th solution obtained in the j-th iteration.

Iteration 1.
S=15. We obtain 15 solutions listed in Tab.6. In this table, only middle values of criteria for particular efficient points are shown. We have got a wide range of criteria values.

---

[1]Although we use here a representation off fuzzy numbers consistent with their definition, in FLIP we introduce directly the values of left and right ranges insted of left and right spreads

After a first brief analysis we can point out solutions that are not attractive because of too high deviation of values of particular objectives from their aspiration levels, e.g for gross profit - solutions $1_1, 2_1, 3_1, 5_1, 6_1, 8_1, 9_1, 11_1, 12_1, 14_1, 15_1$.

From among solutions $4_1, 7_1, 10_1, 13_1$, solution 10_1 is chosen as a new starting point for the next iteration. Now we are looking for a solution with a lower value of criterion z_3.

TABLE 6.

	1_1	2_1	3_1	4_1	5_1	6_1	7_1	8_1
z_1	15618	15645	15644	18891	14022	14126	20088	12506
z_2	10.8	10.8	10.8	8.5	14.9	8.5	6.1	17.6
z_3	536	536	538	648	648	373	760	690

	9_1	10_1	11_1	12_1	13_1	14_1	15_1
z_1	12539	20568	11178	13339	20763	9940	14605
z_2	6.1	3.7	17.6	3.8	1.3	17.6	1.4
z_3	278	872	600	241	983	516	215

Iteration 2.

Let's take S=12. We have got 8 solutions (Tab.7) and 4 of generated problems are contradictory. When we are searching through the feasible region, we make successive reductions of it and it can lead, in some cases, to contradictory problems.

In this iteration we have got three very interesting solutions: $1_2, 7_2$ and 10_2. For our DM, solution 7_2 is the best (Fig.16).

TABLE 7.

	1_2	3_2	4_2	6_2	7_2	9_2	10_2	12_2
z_1	19045	20567	19647	20557	19878	20547	19952	20537
z_2	3.7	3.7	3.1	1.9	4.0	0.2	4.0	0.1
z_3	503	937	539	905	575	897	611	891

As we can see, for solution 7_2, the value of the gross profit and the structure-forming plants area are within the range of the fuzzy aspiration levels. To get full evaluation of solution 7_2, let us also analyze the state of fuzzy constraints. In the upper windows of Fig.17, we have constraints corresponding to the demand of manpower in the spring and summer periods. As we can see, this demand has been satisfied with some surplus and no risk of violation of these constraints. For a constraint in the lower window that corresponds to the demand of manpower in the autumnal period, the risk of violation is very small. So, we can try to relax constraints 7, 8, 9 by changing safety coefficients.

Let's assume $\tau_i = 0.6, \eta_i = 0$ and $\theta_i = 0$ for all i. **Iteration 1'.**
For S=15 we obtain solutions listed in Tab.8.

TABLE 8.

	$1_{1'}$	$2_{1'}$	$3_{1'}$	$4_{1'}$	$5_{1'}$	$6_{1'}$	$7_{1'}$	$8_{1'}$
z_1	15642	15652	15651	18969	14039	14133	20167	12524
z_2	10.6	10.6	10.6	8.3	14.7	8.3	5.9	17.6
z_3	492	492	493	602	602	333	713	656
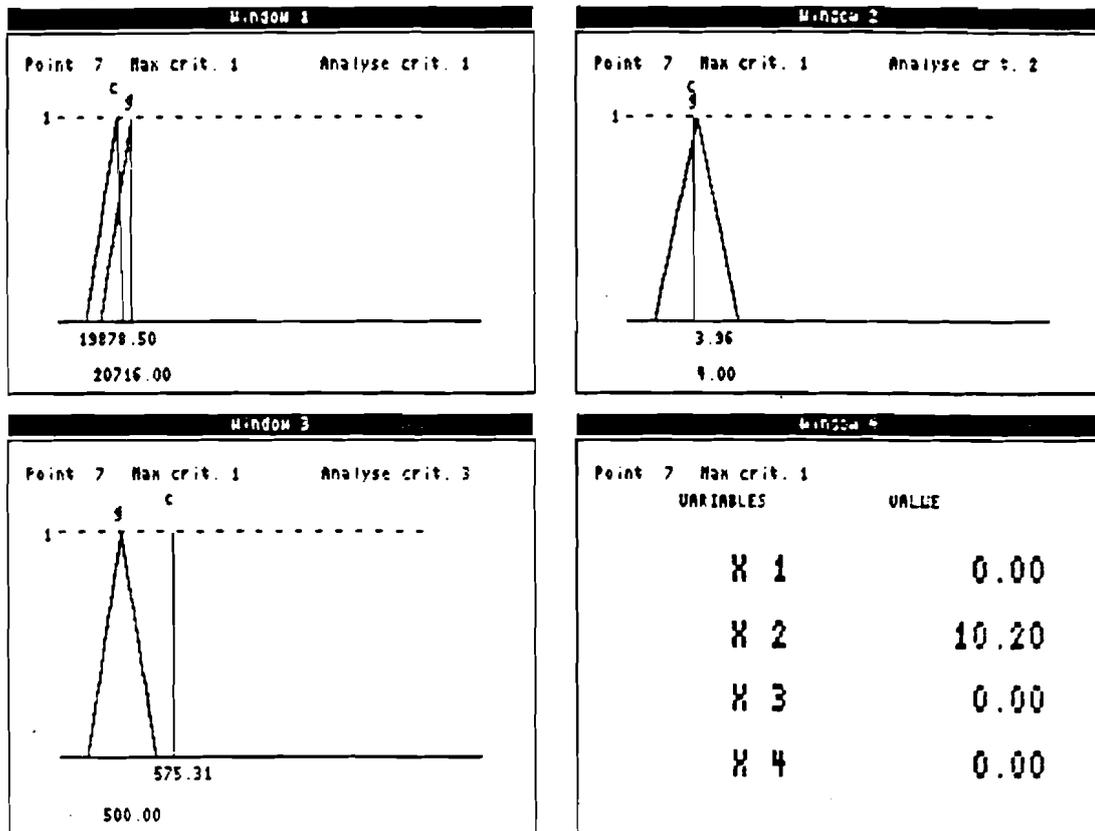
Figure 16. Values of criteria for solution 7_2.

	$9_{1'}$	$10_{1'}$	$11_{1'}$	$12_{1'}$	$13_{1'}$	$14_{1'}$	$15_{1'}$
z_1	12554	20595	11202	13319	20692	9969	14596
z_2	5.9	3.4	17.6	3.5	2.1	17.6	1.1
z_3	236	823	566	198	933	483	171

Iteration 2'.

Taking solution $10_{1'}$ as a starting point and S=12, we get 8 new solutions (Tab.9).

TABLE 9.

	$1_{2'}$	$3_{2'}$	$4_{2'}$	$6_{2'}$	$7_{2'}$	$9_{2'}$	$10_{2'}$	$12_{2'}$
z_1	19707	20594	19954	20582	20058	20570	20164	20558
z_2	3.2	3.4	4.0	1.7	4.0	0.0	4.0	0.3
z_3	503	862	541	841	584	833	624	826

For solution $4_{2'}$ the middle values of all objectives are within the range of their fuzzy aspiration levels. Fig.18 shows the state of fuzzy constraints for this solution. As we can see, there is no risk of violation of constraints 7 and 8 (lower windows) and a little bigger risk for constraint 9, then in solution 7_2.

Finally, the DM has accepted solution $4_{2'}$ as the best compromise. The values of corresponding criteria are presented in Fig.19.
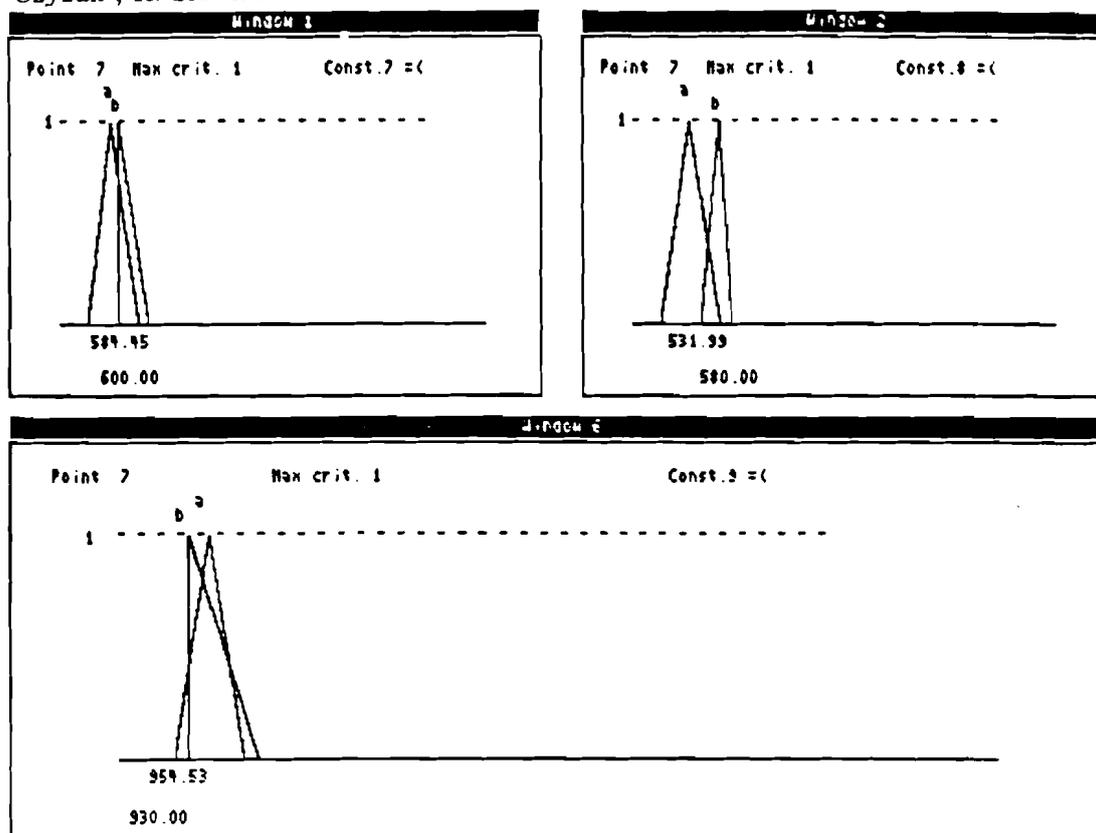
Figure 17. Fuzzy constraints for solution T_2.

So, we have got the following definition of the best-compromise farm structure:

a) plant production for sale:

- 0.80 ha of winter wheat,
- 9.20 ha of winter barley,
- 0.67 ha of spring barley,
- 2.75 ha of potatoes;

b) plant production for fodder consumed in the farm:

- 1.92 ha of potatoes,
- 0.5 ha of fooder beat,
- 4 ha of lucerne;

c) 4 ha of permanent grassland under cultivation;

d) purchase of fertilizers:

- 3267 kg of phosphorus,
- 2866 kg of nitrogen,
- 2397 kg of potassium;

e) manpower hire:

- 281 hour's in the spring period,
- 260 hour's in the summer period.
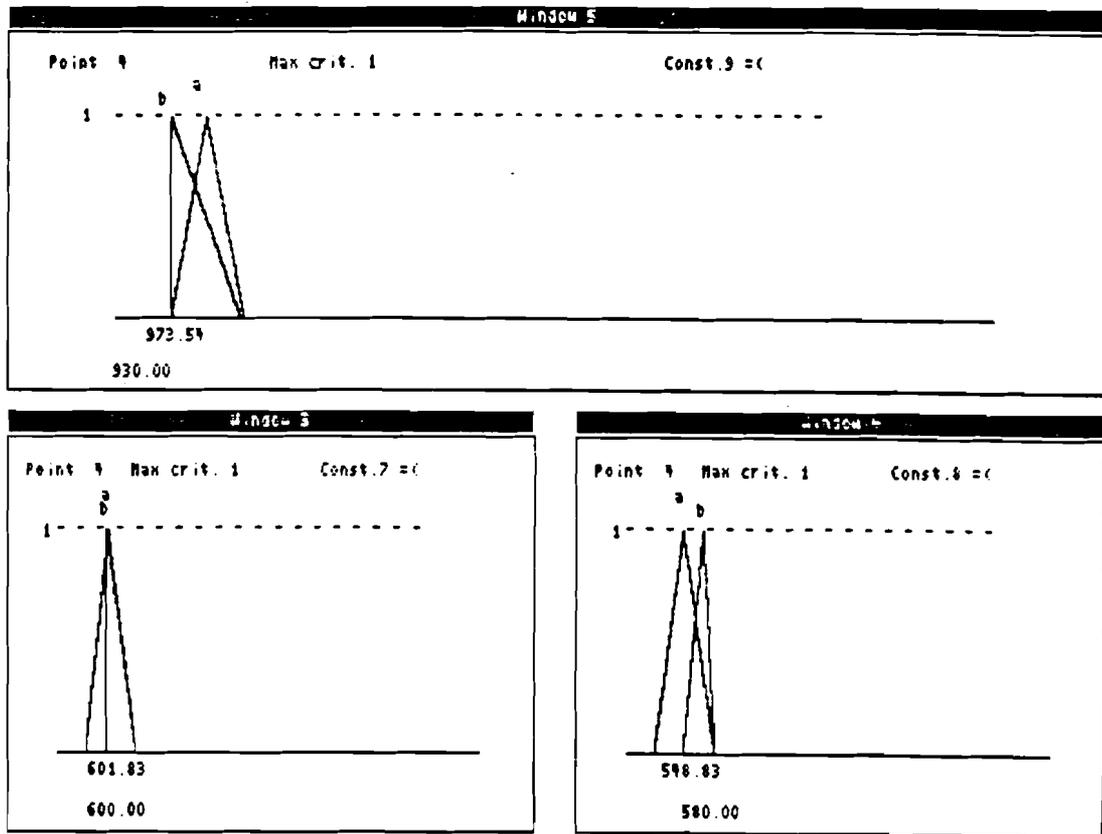
Figure 18. Fuzzy constraints for solution $4_{2'}$.
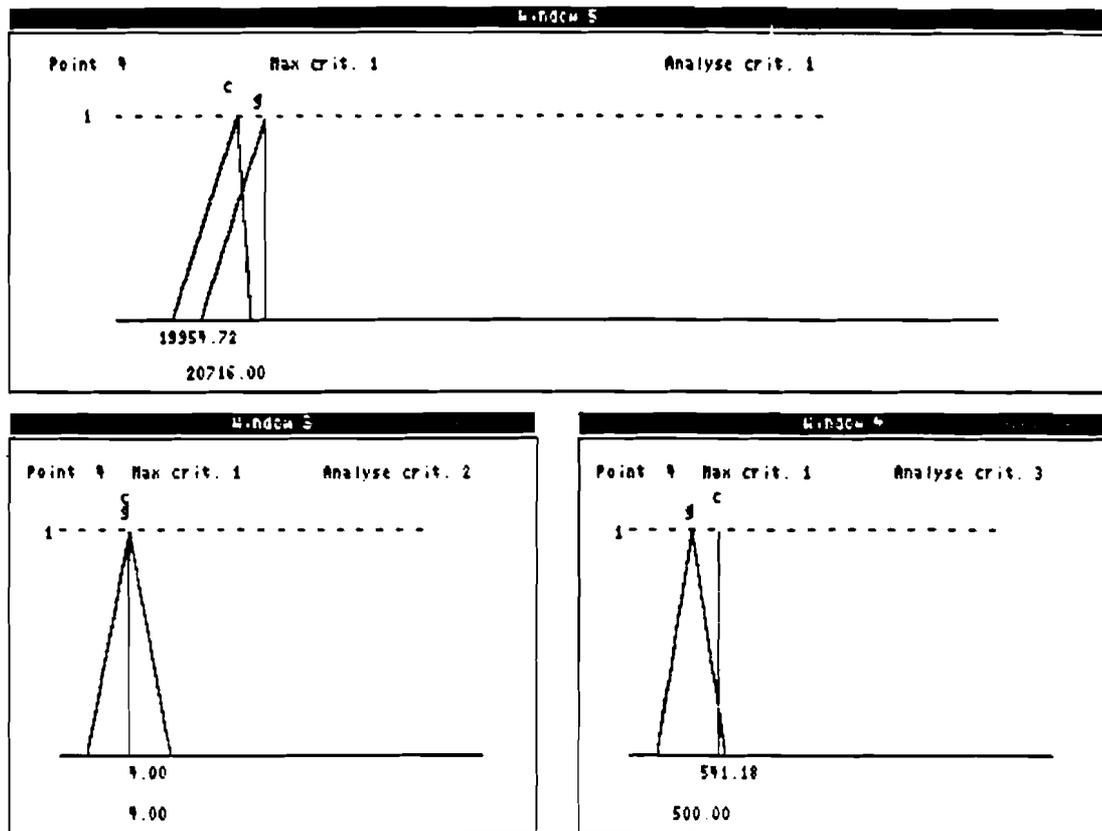

Figure 19. Values of criteria for solution $4_{2'}$.

References

[1] Choo, E.U., Atkins, D.R. (1980). *An interactive algorithm for multicriteria programming.* Computers and Operations Research 7, pp. 81-87.

[2] Czyżak, P. (1989). *Multicriteria agricultural problem solving under uncertainty* Foundations of Control Engineering 14, no.3, pp. 61-80.

[3] Czyżak, P. (1990). *Application of the FLIP method to farm structure optimization problems.* In R.Słowiński, J.Teghem (eds.), Stochastic vs. Fuzzy Approaches to Multiobjective Mathematical Programming, Kluwer Academic Publishers, pp. 263-278.

[4] Czyżak, P. Słowiński, R. (1989). *Multiobjective diet optimization problem under fuzziness.* In J.L. Verdegay, M. Delgado (eds.), The Interface between Artificial Intelligence and Operations Research in Fuzzy Environment, Verlag TUV Rheinland, Koln, pp.85-103.

[5] Delgado, M., Verdegay, J.L., Vila, M.A. (1989). *A general model for fuzzy linear programming* Fuzzy Sets and Systems 29, 21-30.

[6] Dubois, D. (1987). *Linear programming with fuzzy data.* In J.C. Bezek (ed.), Analysis of Fuzzy Information, Vol.3:Applications in Engineering and Science, CRC Press, Boca Raton, pp. 241-263.

[7] Inuiguchi, M., Ichihashi, H., Tanaka, H. (1990). *Fuzzy programming: a survey of recent developments.* In R. Słowiński, J. Teghem (eds.), Stochastic vs. Fuzzy Approaches to Multiobjective Mathematical Programming, Kluwer Academic Publishers, pp. 45-68.

[8] Luhandjula, M.K. (1989). *Fuzzy optimization: an appraisal* Fuzzy Sets and Systems 30, 257-282.

[9] Rommelfanger, H. (1989). *Interactive decision making in fuzzy linear optimization problems* European J. Operational Res. 41, 210-217.

[10] Sakawa, M., Yano, H. (1989). *Interactive fuzzy decision making for generalized multiobjective linear programming problems with fuzzy parameters* Fuzzy Sets and Systems 29, 315-326.

[11] Słowiński, R. (1986). *A multicriteria fuzzy linear programming method for water supply system development planning* Fuzzy Sets and Systems 19, 217-237.

[12] Słowiński, R. (1987). *An interactive method for multiobjective linear programming with fuzzy parameters and its application to water supply planning.* In J. Kacprzyk, S.A. Orlovski (eds.), Optimization Models using Fuzzy Sets and Possibility Theory, D.Reidel, Dordrecht, pp.396-414.

[13] Słowiński, R., Teghem, J. (1990). *Comparison study of STRANGE and 'FLIP'.* In R. Słowiński, J. Teghem (eds.), Stochastic vs. Fuzzy Approaches to Multiobjective Mathematical Programming, Kluwer Academic Publishers, pp. 365-393.

[14] Słowiński, R., Teghem, J., eds. (1990). *Stochastic vs. Fuzzy Approaches to Multiobjective Mathematical Programming,*Kluwer Academic Publishers, Dordrecht.

[15] Słowiński, R., Urbaniak, A., W eglarz, J. (1986). *Probabilistic and fuzzy approaches to capacity expansion planning of a water supply system.* In L.Valadares Tavares, J.Evaristo da Silva (eds.), Systems Analysis Applied to Water and Related Land Resources, Pergamon Press, Oxford, pp.93-98.

[16] Vanderpooten, D. (1990). *Multiobjective programming: basic concepts and approaches.* In R. Słowiński, J. Teghem (eds.), Stochastic vs. Fuzzy Approaches to Multiobjective Mathematical Programming, Kluwer Academic Publishers, pp. 7-22.