A PARAMETRIC APPROACH TO FUZZY LINEAR PROGRAMMING

Christer CARLSSON
Department of Business, Åbo Academy, 20500 Åbo 50, Finland

Pekka KORHONEN
Helsinki School of Economics, 00100 Helsinki, Finland

Received July 1984
Revised October 1985

In this paper we discuss how to deal with decision problems that are described with LP models and formulated with elements of imprecision and uncertainty. More precisely, we will study LP models in which the parameters are not fully known but only with some degree of precision.

Even with incomplete information the model builder (or model user) is normally able to give a realistic interval for the parameters of an LP model. For the constraint vector this is combined with some wishes or some leeway on the constraints. Even with ambiguity in the objective function, there is normally some preference ordering to be found among alternative ways of action. We will demonstrate that these modelling complications can be handled with the help of some results developed in the theory of fuzzy sets.

After an overview of some central contributions to fuzzy linear programming, we will develop an LP model in which the parameters are not fully known, only with some degree of precision, and show that the model can be parametrised in such a way that the optimal solution becomes a function of the degree of precision. The fuzzy LP model derived in this way appears to be fairly easy to handle computationally, which is demonstrated with a numerical example.

Keywords: Fuzzy LP, Parametric programming.

1. Introduction

The traditional way to evaluate any imprecision in the parameters of an LP model is through a post-optimization analysis, with the help of sensitivity analysis, shadow prices and parametric programming. However, none of these methods is suitable for an overall analysis of the effects of imprecision in the parameters. Sensitivity analysis can be used to generate alternatives in the environment of the optimum. Shadow prices indicate how much the optimal solution will improve as a function of the constraint vector. By using parametric programming it is possible to analyze overall changes in the constraints and the objective function. But none of these methods offers any indication of an optimal policy for variations or for best designs.

Another way to handle imprecision is to model it with uncertainty and to apply stochastic programming (see, e.g. Wets [9]). Stochastic programming, however, is built on the assumption that some of the parameters used in describing the

North-Holland

problem are random variables, which are difficult to determine exactly. Stochastic
programming models are often very complex in structure and obviously therefore
rarely used in practice.

A third way to come to terms with imprecision is to apply some of the results
achieved in the theory of fuzzy sets, which proposes to give the conceptual and
theoretical framework for handling complexity, imprecision and vagueness (cf.
Bellman and Zadeh [1]). As this is a new, flexible and powerful approach we will
try it here, and find out how imprecision in the parameters of an LP model could
be handled.

In this paper we discuss how to deal with decision problems that are described
with LP models and formulated with elements of imprecision and uncertainty.
More precisely, we will study LP models in which the parameters are not fully
known but only with some degree of precision.

Even with incomplete information the model builder (or model user) is
normally able to give a realistic interval for the parameters of an LP model. For
the constraint vector this is combined with some wishes or some leeway on the
constraints. Even with ambiguity in the objective function there is normally some
preference ordering to be found among alternative ways of action. We will
demonstrate that these modelling complications can be handled with the help of
some results which have been developed in the theory of fuzzy sets.

After an overview of some contributions to fuzzy linear programming – aimed
at finding ways to handle imprecision – we will develop an LP model in which the
parameters are known with only some degree of precision, and show that the
model can be parametrized in such a way that the optimal solution becomes a
function of the grades of precision. The fuzzy LP model derived in this way
appears to be fairly easy to handle computationally, which is demonstrated with a
numerical example.

2. Fuzzy linear programming

Let us consider two versions of the LP model, the standard and the fuzzy:

\[
\text{min } z = cx, \quad (\text{standard formulation}),
\]

\[
\text{s.t. } Ax \geq b, \quad x \geq 0
\]

\[
\text{find } x, \quad (\text{fuzzy formulation}).
\]

\[
\text{s.t. } cx \approx z_0, \quad Ax \geq b_0, \quad x \geq 0
\]

In the fuzzy formulation the strict optimization of the standard formulation is
replaced with a gradual attainment of aspiration levels \((z_0, b_0)\). This principle was
introduced by Bellman and Zadeh [1], and makes use of (i) the properties of the
membership function, and (ii) the operators for combining membership functions
with the logical connective ‘and’. As we replace strict optimization and strict
constraints with aspiration levels, the constraints and the objective function serve identical purposes for solving the decision problem; we do not get a solution which is 'optimal within given constraints', but rather a 'good enough compromise' in terms of the aspiration levels given to the objective and the constraints.

This 'good enough compromise' is found by solving (2.1b). For that purpose the membership functions should be made explicit and defined in some proper way. The 'proper way' is open for the model builder (or the model user) to decide, but clearly he should represent his aspirations relative to the objective, the constraints and some interdependence between them – to the extent these factors are known. The membership functions offer the means for trials and experiments with various aspiration levels, and for trying out various combinations of objectives and constraints.

In an LP formulation the most practical way to construct membership functions is to use a linear form (cf. Zimmermann [11]):

$$\mu_i((A'x)) = \begin{cases} 
1 & \text{for } (A'x)_i \leq b'_i, \\
1 - \frac{(A'x)_i - b'_i}{\delta'_i} & \text{for } b'_i < (A'x)_i \leq b'_i + \delta'_i, \\
0 & \text{for } (A'x)_i > b'_i + \delta'_i 
\end{cases} \quad (2.2)$$

where $A'$ is the matrix $A$ into which the objective function $c'x$ has been inserted, $b'$ the constraint vector $b$ with $z_0$, and $\delta'_i$ an admissible deviation from the objective or the constraints ($\delta'_i$ includes $\delta_n$); the index $i$ denotes the $i$-th row of the matrix. Now the original LP formulation is transformed to

$$\mu_c(A'x) = \max_{x \geq 0} \min_i \mu_i((A'x)_i) \quad (2.3)$$

where $\mu_c$ denotes the 'good enough compromise'. The formulation in (2.3) is an implementation of the Bellman-Zadeh principle of forming the conjunction of the fuzzy sets representing the objective and the constraints,

$$\forall x \in X, \quad \mu_c(x) = \min \left[ \min_i \mu_i(x), \min \mu_j(x) \right],$$

where $b_i, (i = 1, \ldots, m)$ are constraints and $G$ is an objective; this is easily expanded to multiple objectives by replacing $G$ with $G_j (j = 1, \ldots, n)$ and constructing a corresponding number of membership functions. For the conjunction they suggested the min-operator, which is a safe and conservative approach, and propose then to find the maximum as a means for selecting a 'best' alternative. This can be done with an equivalent formulation to (2.3):

$$\max_{x \geq 0} \min_i \left( \frac{b'_i - (A'x)_i}{\delta'_i} \right). \quad (2.4)$$
which, in turn, is equivalent to solving (cf. Charnes and Cooper [3])

\[
\begin{align*}
\max & \quad \lambda_i \\
\text{s.t.} & \quad \lambda_i \leq \left( \frac{b_i - (A'x)_i}{\delta_i} \right), \quad (i = 0, \ldots, m) \\
& \quad x \geq 0
\end{align*}
\]

but which is treatable with the Simplex algorithm. When formulated and solved in this fashion, the fuzzy LP model does not give more information, but probably a better insight in the decision problem than a systematic sensitivity analysis.

In the fuzzy LP model the linear membership functions support the implicit assumption that the aspiration levels assigned to the objective and the constraints are equally important, and the min-operator that the ‘good enough compromise’ is rather poor. Some influence on the compromise can be achieved through the vector \( \delta' \), but the method is nearly as static as the standard LP formulation. Through nonlinear and other forms for membership functions it might be possible to work with more advanced assumptions on the interdependences and relative importance of the objective and the constraints – but there is no easy way to solve the resulting optimization problem. The fuzzy LP model is no better for problems in multiple objectives: as shown by Zimmermann [11] the objectives are handled as if they were independent, which is pretty unrealistic for a case with multiple objectives.

Another source for inflexibility seems to be the min-operator, which in some cases produces counter-intuitive results (cf. Zeleny [10]). Hamacher (1975) has suggested a more general operator for combining membership functions,

\[
D_\gamma(\mu_A, \mu_B) = \frac{\mu_A \mu_B}{\gamma + (1 - \gamma)(\mu_A + \mu_B - \mu_A \mu_B)}, \quad \gamma > 0,
\]

where \( \mu_A, \mu_B \) are two membership functions (\( D_\gamma \) can be extended to \( m \) membership functions, \( \gamma \) is a parameter through which the operator can be calibrated; for \( \gamma = 1 \) we have the special case \( D_1(\mu_A, \mu_B) = \mu_A \mu_B \)).

Compared with the min-operator the \( \gamma \)-operator is less conservative in the sense that the solution space is larger, and it thus offers more alternatives for consideration; it can, with some modifications, be made to fit into an LP model. There are, however, no provisions for interdependences between an objective and constraints, or between multiple objectives, neither are there any means for exploiting relative importances of objectives and constraints. It might be possible to develop the parameter into a set of functions through which these aspects of a decision problem can be captured and exploited.

Dubois and Prade [4] give some extensions to the original formulations of the fuzzy LP model. In their formulation imprecision is represented by imprecise parameters, which is achieved by defining the coefficients \( a_i \in A \) and \( b_i \in b \) as LR fuzzy numbers. With these elements we get

\[
(Ax)_i \leq \tilde{b}_i, \quad \text{or} \quad (Ax)_i \geq \tilde{b}_i \quad (i = 0, \ldots, m)
\]

where \( i \) again denotes the \( i \)-th constraint; the equality should be interpreted as an
equality of the membership functions; the inclusion is weaker than the equality, and \( \tilde{b}_i \) should be interpreted as the maximum tolerance – including the fuzziness – for \( (\tilde{A}x)_i \). With L–R fuzzy numbers the matrix \( A \) is written in the following form:

\[
A = (A, \underline{A}, \overline{A})_{LR}
\]

(2.8)

where \( A, \underline{A}, \overline{A} \) are vectors of mean values and left and right spreads. Then, as the vector \( x \) is positive, an equivalent model of (2.7) would be

\[
(Ax)_i = b_i, \\
(Ax)_i \geq b_i, \quad (i = 0, \ldots, m; x \geq 0), \\
(Ax)_i \leq \tilde{b}_i
\]

(2.9)

which is an ordinary linear system of equalities and inequalities, and solvable under normal conditions. A similar formulation – the 'robust programming model' – was introduced by Negoita [7] but was not reduced to a traditional LP formulation.

Another way to handle LP models in which the coefficients are fuzzy numbers is to formulate it as an approximate equality,

\[
\tilde{A}_j x = (\tilde{A}x)_j = \tilde{b}_j \quad (i = 0, \ldots, m, j = 1, \ldots, n)
\]

(2.10)

where \( \tilde{A}_j \) is a row vector of L–R fuzzy numbers, \( \tilde{b}_j \) is a L–R fuzzy number and = denotes approximate equality. Dubois and Prade [4] have shown that this system is equivalent to a nonfuzzy system

\[
b_i - (Ax)_i \leq \tilde{b}_i + (\tilde{A}x)_i \quad \text{when } 0 \leq b_i - (Ax)_i, \\
(Ax)_i - b_i \leq \tilde{b}_i + (Ax)_i \quad \text{when } 0 \leq (Ax)_i - b_i.
\]

(2.11)

Another way to interpret the approximate equality is to define the constraint domain by the following membership function \( \mu_i \):

\[
\mu_i(x) = \begin{cases} 
R \left[ \frac{b_i - (Ax)_i}{\tilde{b}_i + (\tilde{A}x)_i} \right] & \text{if } b_i - (Ax)_i \geq 0, \\
L \left[ \frac{(Ax)_i - b_i}{\tilde{b}_i + (\tilde{A}x)_i} \right] & \text{if } (Ax)_i - b_i \geq 0.
\end{cases}
\]

(2.12)

The optimal decision \( x \), with respect to the \( m \) fuzzy constraints,

\[
\max_{x \geq 0} \min_i \mu_i(x) \quad (i = 0, \ldots, m),
\]

(2.13)

is found by solving the nonlinear programming model formulated in (2.13). Dubois and Prade also state that this approach can be extended to fuzzy linear objective functions and to linear approximate inequality constraints, as well as generalized to fuzzy variables.

A similar, but algorithmically different way to handle imprecision has been introduced by Chanas [2], who by extending Zimmermann's [11] fuzzy LP model.
proposes the following formulation:

\[
\text{max } cx, \\
\text{s.t. } \sum_{j=1}^{n} a_{ij} x_j \leq h_i + \theta p_i, \quad (i = 1, \ldots, m) \\
x \geq 0
\]  

(2.14)

where \( \theta (0 \leq \theta \leq 1) \) is the grade of constraint violation and \( p_i (p_i > 0) \) the tolerance, or actual violation. Now it is possible to select a \( \theta^{(1)} \) such that for an active constraint \( i \) of an admissible solution \( \theta^{(1)} \), we get (assuming appropriate linear membership functions) \( \mu_i(\theta^{(1)}) = 1 - \theta^{(1)} \). Chanas states that when solving (2.14) with a parametric programming technique it is possible – through successive selections of \( \theta \) – to find a set of solutions, each of which represents a joint attainment of the constraints by the grade \( 1 - \theta \). For each \( \theta^{(k)} \) there is a vector \( x^{(k)} \) which maximizes the objective function, and it should thus be possible to trace an analytical relationship between \( z^* = \text{max } cx^{(k)} \) and \( \theta^{(k)} \); Chanas states that this relationship is a continuous, piece-wise linear and concave function in \( \theta \).

If this can be proved it should be a useful result, and would allow a continuous trade-off between grades of constraint violation (and with some extension – also among individual constraints) and grades of attainment of goal aspirations. Chanas, however, does not develop his model in this direction; he solves for the \( \theta \) for which the maximum tolerance can be applied in the constraints. Thus the trade-off possibilities are not exploited and the solution is probably not ‘trade-off optimal’. The parametric programming technique is also applied to a fuzzy MOLP problem, but in a similar fashion, and does not enhance existing methods for coming to terms with the fundamental problem – interdependence of the objectives. Chanas’ parametric programming approach is thus essentially no extension of the traditional fuzzy LP model.

A traditional way to deal with imprecision is to equalize it with uncertainty, and to describe and explain it with the help of stochastic variables – or in our case through a stochastic programming model. The debate on the differences between uncertainty and fuzziness is on-going and open-ended, and is relatively far a question of to what extent concepts are misunderstood. It is, therefore, interesting to find that ‘imprecision’ can be dealt with through a combination of stochastic and fuzzy linear programming. This is done by Luhandjula [6], who proposes to treat the coefficients of an LP model as discrete random variables. This corresponds to building an LP model for a possibilistic/probabilistic problem formulation, such as “find the maximum for objective \( i \) by meeting stochastic constraint fluctuations as well as possible”, and is based on the concept of the probability of a fuzzy event; this, in turn, is the expectation on the membership function defining the event. The general formulation of the fuzzy stochastic linear programming model is

\[
\text{min } \delta x, \\
\text{s.t. } Ax \delta b, \quad x \geq 0,
\]  

(2.15)
A parametric approach to fuzzy linear programming

where \( \tilde{\min} \) is a modified, flexible version of the normal minimisation; \( \tilde{c} \) and \( \tilde{b} \) are stochastic vectors with known distributions, and \( \tilde{\theta} \) denotes a fuzzy equality or inequality. Luhandjula shows how to reduce the fuzzy SLP problem into a classical one, which can be solved with one of the traditional algorithms. This can be done in several ways, by: (i) reducing a fuzzy stochastic objective function to a deterministic formulation; (ii) reformulating fuzzy stochastic constraints with the help of nonfuzzy probabilities of fuzzy events, and (iii) reformulating fuzzy stochastic constraints using fuzzy probabilities of fuzzy events.

Luhandjula's model is an extension and enhancement of both traditional LP modelling and fuzzy LP techniques. Luhandjula deals with a specific category of cases, in which imprecision is due to hybrid situations involving both uncertainty and fuzziness.

For an excellent review of key principles and main contributions to fuzzy mathematical programming, see Zimmermann [12, 13].

In this section we have been discussing a few approaches to fuzzy linear programming, and will now pursue one of them further: the parametric programming approach. We will take the direction indicated in the discussion of Chanas' article [2], and study the effects on the objective function of various grades of constraint violation. We will allow the membership function(s), which connect the objective function to various aspiration levels, to act as a function of the grades of constraint violation. In an application this means that a decision maker will be able to learn the effects on his aspiration levels of various combinations of constraint violations, and find an 'optimal and satisfying' compromise between aspirations and violations.

3. Fuzzy parametric programming

Let us consider the following LP model:

\[
\begin{align*}
\max & \quad z = c^T x, \\
\text{s.t.} & \quad x \in \mathcal{X} = \{x \mid -Ax \leq b, x \geq 0\},
\end{align*}
\]

where \( c \) is an \( n \)-vector, \( A \) is an \( (m \times n) \)-matrix and \( b \) is an \( m \)-vector. When a model of this type is applied to a practical problem we cannot expect the parameters to be known exactly - we have to accept that they are better or worse estimates from existing data or subjective knowledge. On the other hand, the solution we get in a model should preferably also be an optimal solution to the problem described in the model, which means that we should not only get an optimal solution, but one which is both optimal and implementable. There are grades to the implementability, and these grades are given by membership functions, which, in turn, can be derived from the grades of imprecision in the parameters.

Let us start by establishing a relationship between a solution in the model (3.1), and its parameters:

**Theorem 1.** The solution \( z^* = z^*(c, A, b) \) of the model (3.1) is an increasing function of the parameters \( c, A \) and \( b \).
Proof. Let us consider the model

\[
\begin{align*}
\max & \quad z - (c + \Delta c)x, \\
n & \quad x \in X' = \{x \mid -(A + \Delta A)x \leq b + \Delta b, x \geq 0\},
\end{align*}
\]

where \(\Delta c, \Delta A, \Delta b \geq 0\). Because \(-\Delta Ax \leq 0\) and \(\Delta b \geq 0 \rightarrow X \subseteq X'\) and \(\Delta cx \geq 0\) for all \(x \geq 0\), it follows that

\[
z^*(c + \Delta c, A + \Delta A, b + \Delta b) \geq z^*(c, A, b).
\]

Let us now assume that the user can specify the intervals \([c^0, c^1]\), \([A^0, A^1]\), and \([b^0, b^1]\) for the possible values of the parameters. The lower bounds represent 'risk-free' values in the sense that a solution most certainly should be implementable. The upper bounds, on the other hand, represent parameter values which are most certainly unrealistic, 'impossible' and the solution obtained by using these values is not implementable. When we move from 'risk-free' towards 'impossible' parameter values, we move from solutions with a high grade to solutions with a low grade on implementing – from 'secure' to 'optimistic' solutions. Our task is to find an optimal compromise 'in-between' as a function of the grades of imprecision in the parameters.

In this context it is reasonable to assume that membership functions are monotonically decreasing functions of the parameters. We will denote membership functions by \(\mu\) and use subscripts to specify to which parameters they are associated (\(\mu_c\), \(\mu_A\), or \(\mu_b\)); we will use the subscript \(p\) to refer to a parameter of the model (3.1). Let us scale the membership functions such that \(\mu_p = 1\), if \(p \leq p^0\), and \(\mu_p = 0\), if \(p \geq p^1\); cf. Figure 1.

There are many possible forms for a membership function: linear, exponential, hyperbolic, hyperbolic inverse, piece-wise linear, etc. (cf. Sakawa [8]). Here we will try the exponential form as it is not as restrictive as the linear form, but flexible enough to describe the grades of precision in the parameter values. A membership function of exponential form may be defined as follows [8]:

\[
\mu_p = \mu_p[1 - \exp(-b_p(p - p^1)/(p^0 - p^1))],
\]

(3.2)
where $b_p > 0$ or $b_p < 0$ and $a_p = 1/(1 - \exp(-b_p))$ and $p \in [p^0, p^1]$. $\mu_p = 1$ when $p \leq p^0$, $\mu_p = 0$ when $p \geq p^1$. Figure 2 illustrates the behaviour of the exponential membership function.

The parameter $b_p$ should be specified by the model user, who is to answer, for instance, the following question: "Which value of the parameter $p$ has a fifty-fifty possibility of being realistic?" Using this value $p \in (p^0, p^1)$ we solve the following equation:

$$0.5 = 1/(1 - \exp(-b_p))(1 - \exp(-b_p(p - p^1)/(p^0 - p^1))).$$

which has a solution if $p \neq \frac{1}{2}(p^0 + p^1)$. If the $p$-value given by the user is $p = \frac{1}{2}(p^0 + p^1)$, we use the linear membership function

$$\mu_p = [(p - p^1)/(p^0 - p^1)]$$

where $p \in [p^0, p^1]$ and $\mu_p = 1$ if $p < p^0$, $\mu_p = 0$ if $p > p^1$.

To evaluate the precision of the optimal solution we can apply some results from fuzzy algebra. The membership function $\mu_s$, which outlines the solution, is defined as follows:

$$\mu_s = \min\{\mu_{c_i}, \mu_{a_{ij}}, \mu_{b_i}\}, \quad i = 1, 2, \ldots, m; j = 1, 2, \ldots, n. \quad (3.4)$$

This means that the precision inherit in the optimal solution equals the precision of the most 'risky' of the parameters. In model (3.1) the best value for the objective function at the fixed level of $\mu_s$ is reached when

$$\mu_s = \mu_{c_i} = \mu_{a_{ij}} = \mu_{b_i}, \quad \text{for all } i = 1, 2, \ldots, m; j = 1, 2, \ldots, n, \quad (3.5)$$

because the model tends to use the 'risky' values of the parameters. This can be translated into: "The best value of the objective function, at a fixed level of precision, can be found by using parameter values of the same level of precision". For some component of the right-hand-side vector we could probably use a lower level of precision, because all the constraints are not active in an optimal solution.
This has, however, no effect on the optimal value of the objective function.

Let us then find out how to solve the fuzzy optimization problem. From the linear and exponential membership functions we get the parameter \( p \) as follows:

\[
p = \mu_p(p^0 - p^1) + p^1, \quad (3.6a)
\]
\[
p = p^1 - (p^0 - p^1)/b_p \log(1 - \mu_p/a_p). \quad (3.6b)
\]

Then we can rewrite our model (3.1) in parametric form as:

\[
\text{max } \quad z = (c^1 + A^1) x, \\
\text{s.t. } \quad -(A^1 + A^1 d(b^1 + A^1)) x \geq b^1 + A^1 b^1, \quad (3.7)
\]
\[x \geq 0,\]

since \( \Delta p = \mu_p(p^0 - p^1) \) in the linear case and \( \Delta p = -(p^0 - p^1) \log(1 - \mu_p/a_p)/b_p \) in the exponential case. Because \( \mu_c = \mu_A = \mu_b \) we can omit the indices.

If all the membership functions are linear, (3.1) will have the form

\[
\text{max } \quad z = (c^1 + A^1(x_0 - x^1)) x, \\
\text{s.t. } \quad -(A^1 + A^1 d(b^1 - b^1)) x \geq b^1 + A^1 b^1, \quad (3.8)
\]
\[x \geq 0,\]

and in the exponential case, with all the membership functions identical, we have

\[
\text{max } \quad z = (c^1 + A^1(x_0 - x^1)) x, \\
\text{s.t. } \quad -(A^1 + A^1 d(b^1 - b^1)) x \geq b^1 + A^1 b^1, \quad (3.9)
\]

where

\[\lambda = -\log(1 - \mu/a)/b.\]

In each case we have a linear parametric programming problem to solve. In the linear case the parameter \( \mu : 0 \rightarrow 1 \) and in the exponential case \( \lambda : 0 \rightarrow \log(1 - 1/a)/b. \)

If the membership functions are different in form, we cannot solve the optimization problem with linear parametric programming, but will have to find some alternative way. This is quite easy to do in practice, because we can experiment with different values of the membership function, \( \mu = 0, 0.1, 0.2, \ldots, 1.0 \), and solve the corresponding LP problems. Then, with a set of solutions to this series of LP problems, it is possible to represent the optimal solutions \( z^*_q; q = 1, \ldots, s \) as functions of the membership functions in graphical mode. The graphics give a user a clear holistic perception of how the objective function behaves for varying grades of precision, and enables him to arrive at appropriate conclusions.

4. An illustrative example

In this section we will present a small numerical example on the parametric case in order to illustrate our approach.
A parametric approach to fuzzy linear programming

In the following problem formulation the parameters of the model are defined on an interval. The first numbers represent 'risk-free' values and the second numbers 'impossible'.

\[
\begin{align*}
\text{max} & \quad [1, 1.5)x_1 + [1, 3)x_2 + [2, 2.2)x_3, \\
\text{subject to} & \quad [3, 2)x_1 + [2, 0)x_2 + [3, 1.5)x_3 \leq [18, 22], \\
& \quad [1, 0.5)x_1 + [2, 1)x_2 + [1, 0)x_3 \leq [10, 40], \\
& \quad [9, 6)x_1 + [20, 18)x_2 + [7, 3)x_3 \leq [96, 110], \\
& \quad [7, 6.5)x_1 + [20, 15)x_2 + [9, 8)x_3 \leq [96, 110].
\end{align*}
\]

Note that in order to simplify notations the elements of the matrix have been given in the form \([a^0, a^1]\), where \(a^0 > a^1\).

Let us assume that the following membership functions are associated with each type of parameters:

\[
\begin{align*}
\mu_{a_{ij}} &= (a_{ij} - a^0_{ij})/(a^1_{ij} - a^0_{ij}), & i = 1, 2, 3, 4; j = 1, 2, 3, \\
\mu_{b_i} &= 1/(1 - \exp(-0.8))[1 - \exp(-0.8(b_i - b^0_i)/b^1_i)], & i = 1, 2, 3, 4, \\
\mu_{c_j} &= 1/(1 - \exp(3.0))[1 - \exp(3.0(c_j - c^0_j)/c^1_j)], & j = 1, 2, 3.
\end{align*}
\]

With these membership functions we get the representations of the elements of the model shown in Figures 3a-c.

The number 0 corresponds to 'risk-free' values and the number 1 to 'impossible' values. From the figures (3a–c) we can read the following parameters for \(\mu = 0.5\):

\[
\begin{align*}
\mu_{a_{ij}} &= 0.5: & a_{ij} &= 0.5(a^0_{ij} + a^1_{ij}), & i = 1, 2, 3, 4; j = 1, 2, 3, \\
\mu_{b_i} &= 0.5: & b_i &= 0.6b^0_i + 0.4b^1_i, & i = 1, 2, 3, 4, \\
\mu_{c_j} &= 0.5: & c_j &= 0.22c^0_j + 0.78c^1_j, & j = 1, 2, 3.
\end{align*}
\]

Generally, the parameters can be presented as functions of the membership functions as follows:

\[
\begin{align*}
a_{ij} &= a^1_{ij} + \mu a_{ij}(a^0_{ij} - a^1_{ij}), & i = 1, 2, 3, 4; j = 1, 2, 3, \\
b_i &= b_i - \log(1 - \mu b_i[1 - \exp(-0.8)])/[0.8(h^0_i - h^1_i)], & i = 1, 2, 3, 4, \\
c_j &= c_j + \log(1 - \mu c_j[1 - \exp(3.0)])/[3.0(c^0_j - c^1_j)], & j = 1, 2, 3.
\end{align*}
\]

Now we have a parametric LP problem, where all the coefficients are parametrized, but we cannot use the linear parametric formulation, because the membership functions are of different form (cf. (3.8) and (3.9)). Thus we will have to carry out a series of experiments with various membership values: \(\mu_{a_{ij}} = \mu_{b_i} = \mu_{c_j} = \mu = 0.0, 0.1, \ldots, 1.0\). In this way we solved 11 different LP problems. The corresponding values of the objective function are presented in Figure 4, in which we used linear interpolation for missing values.

From Figure 4 we can see that the objective function has value 27.5 if 'impossible' parameter values are used, and value 12.0, if we use 'risk-free'
Fig. 3a. The membership function of the coefficients of the matrix.

Fig. 3b. The membership function of the right-hand-side vector.
A parametric approach to fuzzy linear programming

Fig. 3c. The membership function of the parameters of the objective function.

Fig. 4. The values of the objective function.
parameter values. From this figure the user can evaluate how the objective function changes as a function of $\mu$. For instance, he can see that the value that corresponds to the fifty-fifty possibility is about 19.5.

5. Concluding remarks

We have been studying modelling problems related to decision problems which are formulated with elements of imprecision. These modelling problems were specified to LP models in which the parameters are known with varying grades of precision.

We made an overview of some central contributions to fuzzy linear programming in order to outline principal arguments in modelling with incomplete information. This was followed by the construction of a novel approach, called the fuzzy parametric programming, which was developed and discussed in some detail; it was illustrated with a simple numerical example.

In conclusion, it seems as if imprecision can be handled fairly easily in a mathematical programming context – if we apply the techniques and results of the theory of fuzzy sets.

References