The Journal of
Mathematics and Computer Science

Available online at
http://www.TJMCS.com

The Journal of Mathematics and Computer Science Vol .5 No.3 (2012) 212-218

Cutting-Plane Algorithm for Solving Linear Semi-infinite Programming in Fuzzy Case

Alireza Fakharzadeh
Department of Mathematics, Shiraz University of Technology
a_fakharzadeh@sutech.ac.ir

Somayeh. Khosravi
Department of Mathematics, Shiraz University of Technology
s.khosravi@sutech.ac.ir

Hamidreza Maleki
Department of Mathematics, Shiraz University of Technology
maleki@sutech.ac.ir

Received: February 2012, Revised: November 2012
Online Publication: December 2012

Abstract

This paper introduces a cutting-plane algorithm for solving semi-infinite linear programming problems in fuzzy case; the problem contains a crisp objective linear function and the infinite number of fuzzy linear constraints. In the first step; the designed algorithm solves a LP problem, which was created by the ranking function method based on a fuzzy sub-problem of the original one. In each iteration of the proposed algorithm, a cutting is created by adding a fuzzy constraint of the original problem to the fuzzy sub-problem. The convergence of the algorithm is proved and some numerical examples are given.

Keywords: Semi-infinite linear programming, Cutting-plane, Fuzzy linear programming.

2010 Mathematics Subject Classification: Primary 54A40; Secondary 46S40.

1. Introduction.

Fuzzy set theory has extensively developed since Zadeh' s pioneer work in 1965 [13] till now. Linear programming (LP) is one of the field has been evolved with the help of fuzzy set theory very rapidly. But, in spite of more adaption of linear semi-infinite programming (LSIP) models with the real life is than LP, rare efficient effort has been done for modeling the natural phenomena as fuzzy linear semi-infinite programming (FLSIP). This many cause since these kind of models have not studied theoretically and more important, no solution method has been presented. The purpose at this paper is to introduce non-symmetric fuzzy linear semi-infinite programming (NFLSIP) model and provide a solution method for it. In this paper we introduce a NFLSIP in section2; then a cutting plane algorithm for solving these kind of problems is illustrated in section 3, which designed based on cutting-plane algorithm for solving LSIP and ranking method function. Also the convergence of the introduced algorithm is proved in this section. Next section is devoted to present several useful numerical test examples in order to demonstrate the application of the method and its efficiency.

2. Non-Symmetric Fuzzy Linear Semi-Infinite Programming

LSIP is a generalization of LP problem in which either the number of constraints or variables (but not both) is allowed to be infinite. The case in which the number of constraints is infinite is as follows [10], [7]:

$$\begin{aligned} (P)\ Min \quad & z = c'x \\ S.to: & a_t'x \geq b_t,\ t \in T \\ & x \geq 0. \end{aligned}$$

Here c, $a_t \in R^n$ and T is an arbitrary infinite index set. In real world Many problems can be modeled as LSIP, such as robot trajectory planning[2], optimal signal sets, production planning and digital filter design [5], air and water pollution control ([5],[4]), design of finite impulse response filters [1]. But in the real world, such problems have fuzzy nature and their goal function or constraints for the possible actions are not crisp. For example when we investigated the air pollution rate, depending to the politic, environmental and other limitations, we would find a rang for its changes. Therefore, the sign $\geq$ in its modeling, does not mean the usual strictly mathematical sense; so, the model (P) can be remodeled as:

$$\begin{aligned} (PF)\ Min: \quad & z = c'x \\ S.to: & a_t'x \succeq b_t,\ t \in T; \\ & x \geq 0, \end{aligned}$$

where the sign $\succeq$ denotes the fuzzy version of $\geq$ and has the linguistic interpretation "essentially greater than or equal to". We call this problem Non-Symmetric Fuzzy Linear Semi-Infinite programming (NFLSIP).

3. Fuzzy Cutting-Plane Algorithm

The most important methods for solving LSIP problems are cutting-plane algorithms, since their applicability under the mild conditions and their strong convergence. In this paper we synthesize the Alternating cutting-plane algorithm, which was proposed for LSIP continuous problems by Gustafson and Kortanks in [4] and method to solve non-symmetric fuzzy linear programming (NFLP) based on ranking functions[10]; then, we designed a solution method for NFLSIP.

In each iteration, the Alternating cutting plane algorithms creates a sub-problem, which is a LP and can be solved by simplex method. But in fuzzy case in each iteration a NFLP problem is created. which is modeled as follows:

$$\begin{aligned} Min: \quad & z = c'x \\ S.to: & a_t'x \succeq b_t,\ t = 1,2,\ldots,r; \quad (3) \\ & x \geq 0, \end{aligned}$$

In order to solve (3), first, for each fuzzy constraint a membership function should be defined; then, one could setup a fuzzy variable linear programming (LPPFV), which has the same optimal solution as (3) with regard to [10]. In order to define a membership function for the t-th constraint of (3), it is assumed that p_t is the maximum tolerance for this constraint; also, without losing the generality, it is supposed that $a_t' \geq 0$ and $b_t \geq 0$. Now, the membership function of the t-th constraint can be defined as:

$$\mu_{(a_t'x)}(x) = \begin{cases} 1, & a_t'x \geq b_t \\ 1 - \dfrac{a_t'x - b_t}{p_t}, & b_t - p_t \leq a_t'x \leq b_t \\ 0, & o.w. \end{cases}$$

Therefore, $S(\mu_{a_t'x}(x)) = \overline{\{x \mid \mu_{a_t'x}(x) > 0\}} = [b_t - p_t, \infty)$. But, we remind that to use Ranking function method, which is explained in [10], $S(\mu_{a_t'x}(x))$ have to be bounded. To overcome this difficulty, we consider corresponding crisp problem of (3) as:

$$\begin{aligned} & Min: z = c'x \\ & S.to: a_t'x \geq b_t,\ t = 1, 2, \ldots, r; \quad (4) \\ & \qquad x \geq 0. \end{aligned}$$

Let $\eta_t = a_t'x_*^r$, in which x_*^r is an optimal solution of the problem (P_r). Now the membership function of the t-th constraint of (4) can be redefined as:

$$\mu^*_{(a_t'x)}(x) = \begin{cases} 1, & b_t \leq a_t'x \leq \eta \\ 1 - \dfrac{a_t'x - b_t}{p_t}, & b_t - p_t \leq a_t'x \leq b_t \\ 0, & o.w. \end{cases}$$

Now, the following LPPFV problem and the problem (3) have the same optimal point provided that they have the same membership functions (see[10]):

$$\begin{aligned} & Min \quad z = c'\tilde{x} \\ & S.to: a_t'\tilde{x} \underset{R}{\geq} (b_t, \eta_t, p_t, 0),\ t = 1, 2, \ldots, r \qquad (5) \\ & \qquad \tilde{x}_j \underset{R}{\geq} 0, \end{aligned}$$

where $(b_t, \eta_t, p_t, 0)$ is a quasi-trapezoidal fuzzy number and $\tilde{c} \underset{R}{\geq} \tilde{d}$ if and only if $R(\tilde{c}) \geq R(\tilde{d})$, which R is a ranking function and in this section will be defined. Now problem (5) is a LPPFV problem and its corresponding auxiliary problem (ALLPFV), which is a fuzzy number linear programming, is defined as follows:

$$\begin{aligned} & Max \quad z = \sum_{t=1}^{r} \tilde{b}_t y_t \\ & S.to: \ Ay \leq c, \qquad (6) \\ & \qquad y \geq 0, \end{aligned}$$

where $\tilde{b}_t = (b_t, \eta_t, p_t, 0)$, $A = [a_{ti}]_{r \times n}$ and $y \in R^r$. Problem (6) can be transformed by applying one of the Ranking function based methods. It is preferred to apply the Roubence ranking function [12] since it is a linear ranking function which maps each real number to itself; the definition of this function is as follows:

$$R(\tilde{a}) = \frac{1}{2}\int_0^1 (\inf \tilde{a}_\alpha + \sup \tilde{a}_\alpha) d\alpha,$$

where $\tilde{a}_\alpha$ is an α-level set of the fuzzy number $\tilde{a}$; for example if $\tilde{a}$ is a trapezoidal fuzzy number $\tilde{a} = (a^L, a^U, \gamma, \beta)$, then:

$$R(\tilde{a}) = \frac{1}{2}\int_0^1 [\alpha(\gamma-\beta) + a^L - \gamma + a^U + \beta] d\alpha = \frac{1}{2}[a^U + a^L + \frac{1}{2}(\gamma-\beta)].$$

Now, we use Roubence ranking function and obtain:

$$\begin{aligned} &Max: \quad z = \sum_{t=1}^{r} R(\tilde{b}_t) y_t \\ &S.to: \ Ay \le c; \qquad\qquad (7) \\ &\qquad\quad y \ge 0, \end{aligned}$$

(7) is a classic LP ([10]). Dual problem Corresponding to (1-6) problem is defined as follows:

$$\begin{aligned} &Min \quad z = c'w \\ &S.to: a_t'w \ge R(\tilde{b}_t), \ t = 1,2,\dots,r \qquad (8) \\ &\qquad\quad w \ge 0. \end{aligned}$$

Proposition: *Optimal solution of problems (8) and (3) are the same.*

Proof: Let $Y_B = B^{-1}c$ be an optimal basic feasible solution for (7), then according to theorem 7. [10] $X = R(\tilde{b}_{t_B} B^{-1})$ is a fuzzy optimal solution for (3). Also as for relation between primal and dual problems in classic LP, $W = R(\tilde{b}_t)B^{-1}$ is optimal feasible solution of (8). Now we have:

$$X = R(\tilde{b}_{t_B} B^{-1}) = R(\tilde{b}_{t_B})B^{-1} = W.$$

Let $\varepsilon \ge 0$ and T_0 are given ($T_0 \subset T$, T_0 is finite and the obtained sub-problem of (PF) by replacing T with T_0 has non-empty bounded level sets ($L(\alpha) = \{x \in R^n \mid a_t'x \ge b_t, \forall t \in T_0; c'x \le \alpha\}$ is called an α-level set). By supposing r=0, the description of Fuzzy cutting-plane algorithm (FCP) is as follows; the convergence of this algorithm is proved.

Step1. Solve the following LP problem:

$$\begin{aligned} &(PF)_r \ Min \quad z = c'w \\ &S.to: a_t'w \ge R(\tilde{b}_t), \ t \in T_r \\ &\qquad\quad w \ge 0. \end{aligned}$$

If $(PF)_r$ is inconsistent, then stop, since the problem PF is inconsistent too. Otherwise, calculate the optimal solution of $(PF)_r$; say it x^r and then go to Step2.

Step2. Compute $s_r = \underset{t \in T}{Inf} \ (a_t'x - R(\tilde{b}_t))$.

If $s_r \ge -\varepsilon$ then x^r is optimal solution of (PF) and stop; otherwise, define $t_r = \arg\min_{t \in T} (a_t'x^r - R(\tilde{b}_t))$ and go to Step3.

Step3. Consider $T_{r+1} = T_r \bigcup \{t_r\}$, r=r+1 and then loop to Step1.

The following convergence theorem, which is similar to the theorem 11.2 in [4], hold:

Theorem 3.1. *Assumes that PF is consistent; that is, the set $\{a_t \mid t \in T\}$ is bounded and the slack function $(a_t'x - b_t)$ is bounded from below at each iteration of the introduced algorithm. Then, the algorithm terminates after a finite number of iterations or generates an infinite sequence, which its cluster points are optimal solution of (PF).*

Proof : Suppose the introduced algorithm generates an infinite sequence. Also let F_r is the feasible set in r-th iteration. Since, $T_r \subset T_{r+1}$ for $r \in N$, $F_{r+1} \subset F_r$ for r = 0,1, 2,...; then $\{x^r\} \subseteq \hat{F}_0 = \{x \in F_0 \mid c'x \geq c'z\}$, where z is a given feasible solution of the PF problem. Since $\hat{F}_0$ is a compact set, $\{x^r\}$ has at least a cluster point (e.g. a subsequence $\{x^{k_r}\}$ of $\{x^r\}$ is existed such that $\lim_{r\to\infty} x^{k_r} = \bar{x}$. Now we have:

$$c'\bar{x} = c' \underset{r\to\infty}{Lim}\, x^{k_r} = \underset{r\to\infty}{Lim}\, c'x^{k_r} = \underset{r\to\infty}{Lim}\, v(PF_{k_r}) \leq v(PF),$$

where v(PF) is the optimal value of the problem (PF) and $\hat{F}_0$ $v(PF_{k_r})$ is the optimal value of the PF_{k_r} problem. Therefore, if we prove that $\bar{x}$ is a feasible solution of the (PF) (i.e. $\underline{g}(\bar{x}) \succeq 0$), proof will be completed. In this regard, first, we consider $\underline{g}(x) = \underset{t\in T}{Inf}\, g(t,x)$, which was called the marginal function and is continuous [4].

We consider sequence $\{t_r\}$ such that for all r, $t_r = \arg\min_{t\in T}(a_t'x^{k_r} - b_t)$. Therefore

$$a_{t_r}' x^{k_p} \succeq b_{t_r} \quad for\ all \quad p \geq r.$$

So, taking $p \to \infty$ with r fixed, we obtain $a_{t_r}'\bar{x} \succeq b_{t_r}$ (i.e. for all r $g(t_r, \bar{x}) \succeq 0$)

Now, we have

$$\begin{aligned}\underline{g}(\bar{x}) &= \underline{g}(x^{k_r}) + [\underline{g}(\bar{x}) - \underline{g}(x^{k_r})] \\ &\geq g(t_r, x^{k_r}) + [\underline{g}(\bar{x}) - \underline{g}(x^{k_r})] \\ &\succeq g(t_r, x^{k_r}) - g(t_r, \bar{x}) + [\underline{g}(\bar{x}) - \underline{g}(x^{k_r})] \\ &= a_{t_r}'(x^{k_r} - \bar{x}) + [\underline{g}(\bar{x}) - \underline{g}(x^{k_r})].\end{aligned}$$

But $\underset{r\to\infty}{Lim}\,[a_{t_r}'(x^{k_r} - \bar{x}) + [\underline{g}(\bar{x}) - \underline{g}(x^{k_r})]] = 0$ (since $\lim_{r\to\infty} x^{k_r} = \bar{x}$ and $\underline{g}(x)$ is continuous function) , so $\underline{g}(\bar{x}) \succeq 0$.

4. Examples

In this section, the ability of the introduced algorithm for solving NFSILP problems is illustrated by solving several important test examples from LSIP literature, which here were modeled as NFLSIP problems.

Example 4.1. This test problem was remodeled based on the one presented in [7], page 137:

$$Min: \quad \sum_{i=1}^{8} i^{-1} x_i$$

$$S.to: \quad \sum_{i=1}^{8} t^{i-1} x_i \succeq \frac{1}{2-t}, \quad t \in [0,1];$$

$$x \geq 0$$

The algorithm was run under the following settings: $T_0 = \{\frac{i}{10} \mid i = 0,1,\ldots,10\}$ and ε = 0.00001.

The algorithm stopped at iteration r = 20 and Table 2 summarizes the achieved results and their accuracy.

Example 4.2. The following problem was reconstructed from the LSIP problem presented in [9]:

$$Min: \quad \sum_{i=1}^{9} i^{-1} x_i$$

$$S.to: \quad \sum_{i=1}^{9} t^{i-1} x_i \succeq \frac{1}{1+t^2}, \quad t \in [0,1],$$

$$x \geq 0.$$

Here, we supposed $T_0 = \{\frac{i-1}{10} \mid i = 1, 2, ..., 11\}$ and ε =0.0000001.

Example 4.3. The forth test example was made based on a famous LSIP problem (see [4] page 264):

$$Min: \quad \sum_{i=1}^{8} i^{-1} x_i$$

$$S.to: \quad \sum_{i=1}^{8} t^{i-1} x_i \succeq \tan(t), \quad t \in [0,1],$$

$$x \geq 0.$$

For solving this problem, we chose $T_0 = \{\frac{2i-1}{20} \mid i = 1, ..., 10\}$ and $\varepsilon = 10^{-10}$.

Example 4.4. We solve the problem of the designing finite impulse response (FIR) filters [10] in this example. It can be reformulated as a NFLSIP problem as follows:

$$Min: -\sum_{i=1}^{10} r_{2i-1} x_i$$

$$S.to: \; 2\sum_{i=1}^{10} \cos((2i-1)\pi t) x_i \succeq -1, \quad t \in [0, 0.5]$$

$$x \geq 0,$$

where r_i = 0.95i, T_0 were chosen as $T_0 = \{\frac{i-1}{18} \mid i = 1, ..., 10\}$ and the parameter $\varepsilon = 10^{-10}$.

The obtained results of solving example 4.1 - 4.4 by the introduced algorithm and the optimal value of crisp cases are summarized in the following table:

Table 1. Numerical results of examples 4.1 - 4.4

Problem	Estimate	Value (crisp)	CPU time (sec)
4.1	0.6473219234	0.6931481482	5.86
4.2	0.7234516332	0.7853995317	6
4.3	0.5546777321	0.6156532236	5
4.4	-0.4022345123	-0.4835484027	26.92

5. Conclusion

According to the demonstrated theorem 3.1 in section 3, the introduced algorithm terminated in the finite iteration for given $\varepsilon > 0$, which was confirmed by expressing the result in Table 1. In addition, the 4th column of Table 1 indicates that the convergence rate of the introduced algorithm is acceptable.

Moreover, the obtained results of examples 4.1 - 4.4 in comparison with the solutions of their corresponding crisp problem, which were expressed in column 3, were precisely better. Naturally, these results were anticipated since in fuzzy case one, can accept some solutions with small violation of the constraints.

References

[1] R.E. Bellman, and L.A. Zadeh, "Decision making in a fuzzy environment", Management Science, 17(1970), 141-164.

[2] B. Betro, "An accelerated centeral cutting plane", Math. Program, 101(2004), 479-495.
[3] B. Betro, "Numerical treatment of Bayesian robustness problems", International Journal of Approximate Reasoning, 50(2009) , 279-288.
[4] M. A. Goberna and M. A. Lopez, "Linear Semi Infinite Programming", Alicant University, 1998.
[5] A. Ismael, F. Vaz and C. Eugnio Ferreira, "Air pollution control with semi-infinite programming", Applied Mathematical Modelling, 33(2009), 1957-1969.
[6] A. Ismael and F. Vaz, Edite M.G.P. Fernandes and M. Paula S.F. Gomes , "Robot trajectory planning with semi-infinite programming", European Journal of Operational Research, 153(2004), 607-617.
[7] K. Glashoff and S.A. Gustafson, "Linear Optimization and Approximation", Springer-Verlag, Berlin, 1983.
[8] Li, He., Huang , H. Guo and Lu. Hongwei, "Bivariate interval semi-infinite programming with an application to environmental decision-making analysis", European Journal of Operational Research, 211(2011), 452-465.
[9] T. Leon and E. Vercher "A purification algorithm for semi-infinite programming", European Journal of Operational Research, 57(1992), 412-420.
[10] H. R. Maleki, "Ranking function and their applications to fuzzy linear programming", Far East J. Math. Sci (EFMS), 4(2002), NO. 3, 283-301.
[11] P. Moulin, M. Anitescu, K.O. Kortanek and F.A. Potra, "The role of linear semi-infinite programming in signal-adapted QMF bank design", IEEE Trans. Signal Processing, 45(1997), 2160-2174.
[12] M. Roubnes, "Inequality constraints between fuzzy number and their use in mathematical programming Stochastic Versus Fuzzy Approaches To Multi objective Mathematical Programming Under Uncertainly", Kluwer Academic Publishers, (1991), 321-330.
[13] L. A. Zadeh, “Fuzzy sets”, Information and Control, 8 (1965), 338-353.