**Mathematics**

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(6): 13706–13724.
DOI: 10.3934/math.2023696
Received: 27 October 2022
Revised: 17 March 2023
Accepted: 20 March 2023
Published: 10 April 2023

*Research article*

# A trust-region based an active-set interior-point algorithm for fuzzy continuous Static Games

**B. El-Sobky**[1] **and M. F. Zidan**[2,*]

[1] Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt
[2] Department of Physics and Engineering Mathematics, Faculty of Engineering-Tanta University, Egypt

* **Correspondence:** Email: Mai.fathi@f-eng.tanta.edu.eg; Tel: +201007059751.

**Abstract:** In this paper, a novel treatment for fuzzy continuous static games (FCSGs) is introduced. This treatment is based on the fact that, as well as having a fuzzy number, the fuzziness is applied to the control vectors to deal with high vagueness and imprecision in a continuous static game. The concept of the $\alpha$-level set used for converting the FCSGs to a deterministic problem $\alpha$-FCSGs. An active-set strategy is used with Newton's interior point method and a trust-region strategy to insure global convergence for deterministic $\alpha$-FCSGs problems from any starting point. A reduced Hessian technique is used to overcome the difficulty of having an infeasible trust-region subproblem. The active-set interior-point trust-region algorithm has new features; it is easy to implement and has rapid convergence. Preliminary numerical results are reported.

**Keywords:** fuzzy continuous static game; active-set; interior-point; trust-region; reduced Hessian technique
**Mathematics Subject Classification:** 49N10, 49N35, 65K05, 93D22, 93D52

## 1. Introduction

Game theory has been evolving since 1920 and continues to this day. In 1928, Van Neumann [33] proved the central theorem of matrix games for the first time. In 1981, Vincent and Grantham [34] introduced various game formulations. The more general case is achieved assuming that there are multiple decision-makers, each with their cost criterion. We have now entered the realm of game theory, with the game taking on a more general form when numerous decision-makers are present. This generalization introduces the competition concept among system controllers, referred to as "players", and the optimization problem under consideration is a "game". Each player in the game has control over a subset of the system parameters known as his control vector and attempts to minimize his cost function subject to specific constraints. Applications of game theory can be found in engineering, economics, biology, and various other fields. There are three types of games: matrix games, continuous static games, and differential games see [2,29]. In this paper, we will consider continuous static games, in which decision possibilities are not required to be discrete, and costs are also related in a continuous rather than discrete manner. Because there is no time history involved in the relationship between costs and decisions, the game is referred to as static and so, a fuzzy treatment of continuous static games is presented in this paper.

In many branches and fields of study, such as engineering, economics, and others, fuzzy set theory has evolved in various ways over the last 60 years. Zadeh [37] and Goguen [19] published the first papers on the theory of fuzzy sets. The fuzzy set theory was developed to solve problems with some vague and uncertain. Sakawa and Yano [30] proposed an interactive method for solving multi-objective non-linear programming problems with fuzzy parameters in both the objective function and constraints. Osman and El-Banna [28] studied the stability of multi-objective nonlinear programming problems involving fuzzy parameters. Nash equilibrium fuzzy continuous static games were introduced by El-Banna et al. [8]. A cooperative fuzzy game-theoretic approach to multiple objective designs was introduced by Dhingra and Rao [4]. Kassem and Ammar [24] also presented a study of multi-objective fuzzy nonlinear programming problems with fuzzy parameters. Ammar [1] also studies the stability of multi-objective non-linear programming problems with fuzzy parameters in both the objectives and constraints. Khalifa and Zeineldin [22] presented an interactive approach to solving cooperative continuous static games with fuzzy parameters in the objective functions. Khalifa and Zeineldin [23] also published a novel study of cooperative continuous static games in a fuzzy environment.

In this paper, We introduce a multi-player fuzzy continuous static game. The cost functions and constraints in this game both have fuzzy parameters. To deal with high vagueness and imprecision in a continuous static game, these fuzzy parameters are also applied to the control vectors. The concept of the $\alpha$-level set is used to transform the FCSGs into a deterministic problem $\alpha$-FCSGs. To obtain an $\alpha$-Pareto optimal solution to the deterministic problem $\alpha$-FCSGs, an active-set strategy is used with Newton's interior point method and a trust-region technique. This method converges quadratically to $\alpha$-Pareto optimal solutions from any starting point. For the detailed exposition, the reader review [10,16,25,26,35,36].

A projected Hessian method which is suggested by [3,27] and used by [12,13,16,18], utilizes in this paper to treat the difficulty of having an infeasible trust-region subproblem. In this method, the trial step is decomposed into two components and each component is computed by solving a trust-region unconstrained subproblem.

In this paper, we use the symbol $f_{j_k} = f_j(\tilde{x}_k)$, $j = 1, \cdots, p$, $h_k = h(\tilde{x}_k)$, $g_k = g(\tilde{x}_k)$, $\ell_{j_k} = \ell_j(\tilde{x}_k, \lambda_k)$, $\nabla_{\tilde{x}}\ell_{j_k} = \nabla_{\tilde{x}}\ell_j(\tilde{x}_k, \lambda_k)$, and so on. Finally, We use $\|.\|$ to denote the Euclidean norm $\|.\|_2$.

The paper is organized as follows. In Section 2, some basic fuzzy concepts and how the problem FCSGs are converted to a deterministic problem $\alpha$-FCSGs are discussed. A detailed description of the main steps of the active-set technique with Newton's interior point method, trust-region algorithm, and the main steps for the algorithm to solve the $\alpha$-FCSGs problem are presented in Section 3. In Section 4, numerical results are reported. Finally, Section 5 contains concluding remarks.

## 2. Theoretical fuzzy foundations

Fuzzy set theory has been developed for solving problems in which descriptions of activities and observations are imprecise, vague, and uncertain. The term "fuzzy" refers to a situation in which there are no well-defined boundaries of the set of activities or observations to which the descriptions apply.

A fuzzy set is a class of objects with membership grades. A membership function, which assigns to each object a grade of membership, is associated with each fuzzy set. Usually, the membership grades are in [0, 1]. When the grade of the membership for an object in a set is one, this object is absolutely in that set when the grade of the membership function is zero, the object is not in that set. Borderline cases are assigned numbers between zero and one. A fuzzy number is defined differently by many authors such as [31, 32].

Before presenting our approach to the problem FCSGs, we present some definitions which belong to a convex fuzzy type.

**Definition 2.1.** *(Fuzzy number) Let* $\Re$ *be the set of real numbers. A fuzzy number* $\tilde{\beta}$ *is a mapping* $\mu_{\tilde{\beta}} : \Re \to [0, 1]$ *with the following properties*

*(1)* $\mu_{\tilde{\beta}}$ *is upper semi continuous membership function.*
*(2)* $\tilde{\beta}$ *is convex fuzzy set.*
*(3)* $\tilde{\beta}$ *is normal. That is,* $\exists \beta_0 \in \Re$ *for which* $\mu_{\tilde{\beta}}(\beta_0) = 1$.
*(4)* $sup\, \tilde{\beta} = \beta \in \Re : \mu_{\tilde{\beta}}(\beta) > 0$ *is a support of the* $\tilde{\beta}$.

For more details see [31].

**Definition 2.2.** *(A triangle membership) A triangle fuzzy number* $\tilde{\beta}$ *is a continuous fuzzy subset from real line* $\Re$ *whose membership function* $\mu_{\tilde{\beta}}(\beta)$ *satisfies the following conditions*

*(1)* $\mu_{\tilde{\beta}}(\beta)$ *is continuous function from* $\Re$ *to closed interval* $[0, 1]$,
*(2)* $\mu_{\tilde{\beta}}(\beta) = 0$, *if* $\beta < \beta_1$,
*(3)* $\mu_{\tilde{\beta}}(\beta)$ *is strictly increasing with constant rate on* $\beta_1 \leq \beta \leq \beta_2$,
*(4)* $\mu_{\tilde{\beta}}(\beta)$ *is strictly decreasing on* $\beta_2 \leq \beta \leq \beta_3$,
*(5)* $\mu_{\tilde{\beta}}(\beta) = 0$ *if* $\beta_3 < \beta$.

Throughout this paper, a membership function in the following form will be elicited:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} 0 & \text{if } \beta < \beta_1, \\ \frac{\beta-\beta_1}{\beta_2-\beta_1} & \text{if } \beta_1 \leq \beta \leq \beta_2, \\ \frac{\beta_3-\beta}{\beta_3-\beta_2} & \text{if } \beta_2 \leq \beta \leq \beta_3, \\ 0 & \text{if } \beta_3 < \beta. \end{cases} \tag{2.1}$$

**Definition 2.3.** *($\alpha$-level set): The* $\alpha$*-level set of the fuzzy parameters* $\tilde{\beta}$*, is an ordinary set* $L_\alpha(\tilde{\beta})$ *for which the degree of its membership function exceeds the level set* $\alpha \in [0, 1]$ *where*

$$L_\alpha(\tilde{\beta}) = \{\beta \in \Re \mid \mu_{\tilde{\beta}} \geq \alpha\} = \{\beta \in [\tilde{\beta}_\alpha{}^l, \tilde{\beta}_\alpha{}^u] \mid \mu_{\tilde{\beta}} \geq \alpha\}. \tag{2.2}$$

For more details see [21].

In this paper, we will consider the following continuous static games with fuzzy cost functions and fuzzy conditions

$$\begin{array}{ll} \min & f_j(\tilde{t},\tilde{v}) \quad j=1,\cdots,p \\ s.t. & h(\tilde{t},\tilde{v})=0, \\ & g(\tilde{t},\tilde{v})\le 0, \\ & \tilde{t}\ge 0, \quad \tilde{v}\ge 0, \end{array} \tag{2.3}$$

where $f_j$, $j=1,\cdots,p$ represents cost functions for players. The vectors $\tilde{t}\in\Re^{n_t}$ and $\tilde{v}\in\Re^{n_v}$ represent fuzzy state and fuzzy controls respectively.

By using $\alpha$-level, $\alpha\in[0,1]$, the game problem 2.3 restructured as follows

$$\begin{array}{ll} \min & [f_j^l(\tilde{t}_\alpha,\tilde{v}_\alpha), f_j^u(\tilde{t}_\alpha,\tilde{v}_\alpha)] \quad j=1,\cdots,p \\ s.t. & [h^l(\tilde{t}_\alpha,\tilde{v}_\alpha), h^u(\tilde{t}_\alpha,\tilde{v}_\alpha)]=0, \\ & [g^l(\tilde{t}_\alpha,\tilde{v}_\alpha), g^u(\tilde{t}_\alpha,\tilde{v}_\alpha)]\le 0, \\ & \tilde{t}_\alpha\ge 0, \quad \tilde{v}_\alpha\ge 0. \end{array} \tag{2.4}$$

That is, the game problem 2.4 can be divided into the following two games which are defined as a lower game and an upper game respectively

$$\begin{array}{ll} \min & f_j^l(\tilde{t}_\alpha,\tilde{v}_\alpha) \quad j=1,\cdots,p \\ s.t. & h^l(\tilde{t}_\alpha,\tilde{v}_\alpha)=0 \\ & g^l(\tilde{t}_\alpha,\tilde{v}_\alpha)\le 0, \\ & \tilde{t}_\alpha\ge 0, \quad \tilde{v}_\alpha\ge 0, \end{array} \tag{2.5}$$

and

$$\begin{array}{ll} \min & f_j^u(\tilde{t}_\alpha,\tilde{v}_\alpha) \quad j=1,\cdots,p \\ s.t. & h^u(\tilde{t}_\alpha,\tilde{v}_\alpha)=0 \\ & g^u(\tilde{t}_\alpha,\tilde{v}_\alpha)\le 0, \\ & \tilde{t}_\alpha\ge 0, \quad \tilde{v}_\alpha\ge 0. \end{array} \tag{2.6}$$

Both lower game problem 2.5 and upper game problem 2.6 represent a general nonlinear programming problem concerning players $j=1,\cdots,p$. In general, problem 2.5 or 2.6 can be written as follows

$$\begin{array}{lrcl} \text{minimize} & f_j(\tilde{x}_\alpha), & & j=1,\cdots,p \\ \text{subject to} & h(\tilde{x}_\alpha) & = & 0, \\ & g(\tilde{x}_\alpha) & \le & 0, \\ & \multicolumn{3}{l}{\tilde{a}_\alpha\le\tilde{x}_\alpha\le\tilde{b}_\alpha,} \end{array} \tag{2.7}$$

where $\tilde{x}_\alpha=(\tilde{t}_\alpha,\tilde{v}_\alpha)^T\in\Re^n$, $n=n_t+n_v$, $\tilde{a}_\alpha\in\{\Re\bigcup\{-\infty\}\}^n$, $\tilde{b}_\alpha\in\{\Re\bigcup\{+\infty\}\}^n$, and $\tilde{a}_\alpha<\tilde{b}_\alpha$. The functions $f_j:\Re^n\to\Re$, $\forall j=1,\cdots,p$, $h:\Re^n\to\Re^{m_h}$, and $g:\Re^n\to\Re^{m_g}$ are twice continuously differentiable. We assume that $m_h<n$ and no restriction is assumed on $m_g$.

Various approaches have been proposed to solve nonlinear programming problem 2.7 for all $j=1,\cdots,p$, see [9–12,15]. In the following section, we introduce the main steps for the active set strategy with Newton's interior point method and trust-region strategy to solve problem 2.7 for all $j=1,\cdots,p$.

## 3. An active-set strategy with Newton's interior-point and trust-region technique

In this section, firstly, we will introduce a detailed description of the active-set strategy to reduce problem 2.7 to an equivalent equality-constrained optimization problem with bound on variable $\tilde{x}_\alpha$. Secondly, Newton's interior-point method is used to solve the equivalent equality-constrained optimization problem with bound on variable $\tilde{x}_\alpha$. But Newton's method may not converge to a stationary point if the starting point is far away from the solution. To guarantee convergence from any starting point, we will introduce in the third part of this section, a detailed description of the trust-region algorithm. Finally, we will introduce steps for the main algorithm to solve game problem 2.3.

### *3.1. An active-set strategy*

Motivated by the active-set strategy in [6] and used by [11, 12, 14, 17, 18], we define a diagonal matrix $Y(x) \in \Re^{m_g \times m_g}$, whose diagonal entries are

$$y_i(\tilde{x}_\alpha) = \begin{cases} 1 & \text{if } g_i(\tilde{x}_\alpha) \geq 0, \\ 0 & \text{if } g_i(\tilde{x}_\alpha) < 0. \end{cases} \tag{3.1}$$

Using the above diagonal matrix, problem (2.7) is reduced to the following equality-constrained optimization problem with bound on variable $\tilde{x}_\alpha$

$$\begin{aligned} \text{minimize} \quad & f_j(\tilde{x}_\alpha), \quad j = 1, \cdots, p \\ \text{subject to} \quad & h(\tilde{x}_\alpha) = 0, \\ & g(\tilde{x}_\alpha)^T Y(\tilde{x}_\alpha) g(\tilde{x}_\alpha) = 0, \\ & \tilde{a}_\alpha \leq \tilde{x}_\alpha \leq \tilde{b}_\alpha. \end{aligned}$$

The above problem can be reduced to the following problem

$$\begin{aligned} \text{minimize} \quad & f_j(\tilde{x}_\alpha) + \tfrac{\rho}{2}\|Y(\tilde{x}_\alpha) g(\tilde{x}_\alpha)\|^2 \quad j = 1, \cdots, p \\ \text{subject to} \quad & h(\tilde{x}_\alpha) = 0, \\ & \tilde{a}_\alpha \leq \tilde{x}_\alpha \leq \tilde{b}_\alpha, \end{aligned} \tag{3.2}$$

where $\rho$ represents a positive parameter. For the detailed exposition, the reader review [17, 18]. Let

$$\ell_j(\tilde{x}_\alpha, \lambda) = f_j(\tilde{x}_\alpha) + \lambda^T h(\tilde{x}_\alpha), \tag{3.3}$$

and

$$\ell_j(\tilde{x}_\alpha, \lambda; \rho) = \ell_j(\tilde{x}_\alpha, \lambda) + \frac{\rho}{2}\|Y(\tilde{x}_\alpha) g(\tilde{x}_\alpha)\|^2, \tag{3.4}$$

for all $j = 1, \cdots, p$, where $\lambda \in \Re^{m_h}$ represents a Lagrange multiplier vector associated with equality constraint $h(\tilde{x}_\alpha)$.

The Lagrangian function associated with problem (3.2) is defined as follows

$$L_j(\tilde{x}_\alpha, \lambda, \lambda_a, \lambda_b) = \ell_j(\tilde{x}_\alpha, \lambda; \rho) - \lambda_a^T(\tilde{x}_\alpha - \tilde{a}_\alpha) - \lambda_b^T(\tilde{b}_\alpha - \tilde{x}_\alpha), \tag{3.5}$$

where the vectors $\lambda_a$, and $\lambda_b$ are Lagrange multiplier vectors associated with inequality constraints $(\tilde{x}_\alpha - \tilde{a}_\alpha)$ and $(\tilde{b}_\alpha - \tilde{x}_\alpha)$ respectively.

The first-order necessary conditions for a point $\tilde{x}_\alpha^*$ to be a local minimizer of problem (3.2) are the existence of multipliers $\lambda^* \in \Re^{m_h}$, $\lambda_a{}^* \in \Re_+^n$, and $\lambda_b{}^* \in \Re_+^n$, such that $(\tilde{x}_\alpha^*, \lambda^*, \lambda_a{}^*, \lambda_b{}^*)$ satisfies

$$\begin{aligned} \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha^*, \lambda^*; \rho^*) - \lambda_a{}^* + \lambda_b{}^* &= 0, & (3.6)\\ h(\tilde{x}_\alpha^*) &= 0, & (3.7)\\ \tilde{a}_\alpha \le \tilde{x}_\alpha^* &\le \tilde{b}_\alpha, & (3.8) \end{aligned}$$

and for all $e$ corresponding to $\tilde{x}_\alpha^{(e)}$ with finite bound, we have

$$\begin{aligned} \lambda_a{}^{*(e)}(\tilde{x}_\alpha^{*(e)} - \tilde{a}_\alpha^{(e)}) &= 0, & (3.9)\\ \lambda_b{}^{*(e)}(\tilde{b}_\alpha^{(e)} - \tilde{x}_\alpha^{*(e)}) &= 0, & (3.10) \end{aligned}$$

where

$$\nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha^*, \lambda^*; \rho^*) = \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha^*, \lambda^*) + \rho^* \nabla g(\tilde{x}_\alpha^*) Y(\tilde{x}_\alpha^*) g(\tilde{x}_\alpha^*), \quad (3.11)$$

and

$$\nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha^*, \lambda^*) = \nabla f_j(\tilde{x}_\alpha^*) + \nabla h(\tilde{x}_\alpha^*)\lambda^*, \quad (3.12)$$

for all $j = 1, \cdots, p$.

To solve the equality-constrained optimization problem with bound on variable $\tilde{x}_\alpha$ 3.2 for all $j = 1, \cdots, p$, Newton's interior-point method is introduced in the following section.

### *3.2. An interior-point technique*

Motivated by the impressive computational performance of the interior-point technique in [5] and introduced in [11, 12, 16], we define a diagonal matrix $W(x)$ whose diagonal elements are

$$w^{(e)}(\tilde{x}_\alpha) = \begin{cases} \sqrt{(\tilde{x}_\alpha^{*(e)} - \tilde{a}_\alpha^{(e)})}, & \text{if } (\nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha, \lambda; \rho))^{(e)} \ge 0 \text{ and } \tilde{a}_\alpha^{(e)} > -\infty,\\ \sqrt{(\tilde{b}_\alpha^{(e)} - \tilde{x}_\alpha^{*(e)})}, & \text{if } (\nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha, \lambda; \rho))^{(e)} < 0 \text{ and } \tilde{b}_\alpha^{(e)} < +\infty\ ,\\ 1, & \text{otherwise.} \end{cases} \quad (3.13)$$

Using the scaling matrix $W(\tilde{x}_\alpha)$, the first-order necessary conditions (3.7)–(3.10) reduced to the following nonlinear system

$$\begin{aligned} W^2(\tilde{x}_\alpha) \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha, \lambda; \rho) &= 0, & (3.14)\\ h(\tilde{x}_\alpha) &= 0. & (3.15) \end{aligned}$$

Let $\boldsymbol{D} = \{\tilde{x}_\alpha : \tilde{a}_\alpha \le \tilde{x}_\alpha \le \tilde{b}_\alpha\}$ and $int(\boldsymbol{D}) = \{\tilde{x}_\alpha : \tilde{a}_\alpha < \tilde{x}_\alpha < \tilde{b}_\alpha\}$. Systems (3.14) and (3.15) is continuous but not everywhere differentiable. The non-differentiability occurs in two cases:

i) If $w^{(e)}(\tilde{x}_\alpha) = 0$, then these points are avoided by restricting $\tilde{x}_\alpha \in int\boldsymbol{D}$.

ii) If a variable $\tilde{x}_\alpha^{(e)}$ has a finite lower bound and an infinite upper bound (or vice-verse) and $(\nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_\alpha, \lambda; \rho))^{(e)} = 0$. So, we define a vector

$$\eta^{(e)}(\tilde{x}_\alpha) = \frac{\partial (w^{(e)}(\tilde{x}_\alpha))^2}{\partial \tilde{x}_\alpha^{(e)}}, \quad e = 1, \ldots, n+1,$$

such that $\eta^{(e)}(\tilde{x}_\alpha) = 0$ when $(\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))^{(e)} = 0$. This is equivalent to

$$\eta^{(e)}(\tilde{x}_\alpha) = \begin{cases} 1, & \text{if } \ (\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))^{(e)} \geq 0 \ \text{ and } \ \tilde{a}_\alpha^{(e)} > -\infty, \\ -1, & \text{if } \ (\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))^{(e)} < 0 \ \text{ and } \ \tilde{b}_\alpha^{(e)} < \infty, \\ 0, & \text{otherwise.} \end{cases} \tag{3.16}$$

Applying Newton's method on the nonlinear systems (3.14) and (3.15), then we have

$$\begin{aligned} [W^2(\tilde{x}_\alpha)\nabla^2_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho) \ &+ \ \text{diag}(\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))\text{diag}(\eta(\tilde{x}_\alpha))]\Delta\tilde{x}_\alpha + W^2(\tilde{x}_\alpha)\nabla h(\tilde{x}_\alpha)\Delta\lambda \\ &= \ -W^2(\tilde{x}_\alpha)\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho), \end{aligned} \tag{3.17}$$

$$\nabla h(\tilde{x}_\alpha)^T\Delta\tilde{x}_\alpha = -h(\tilde{x}_\alpha), \tag{3.18}$$

where

$$\nabla^2_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho) = H + \rho\nabla g(\tilde{x}_\alpha)Y(\tilde{x}_\alpha)\nabla g(\tilde{x}_\alpha)^T, \tag{3.19}$$

and $H$ is the Hessian of the Lagrangian function (3.3) or an approximation to it.

The diagonal matrix $W(x)$ must be nonsingular, so we restrict $\tilde{x}_\alpha \in int(\boldsymbol{D})$. Set $\Delta\tilde{x}_\alpha = W(\tilde{x}_\alpha)s$ in both Eqs (3.17) and (3.18), and multiply both sides of the Eq (3.17) by $W^{-1}(x)$, we have

$$\begin{aligned} [W(\tilde{x}_\alpha)\nabla^2_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho)W(\tilde{x}_\alpha) \ &+ \ \text{diag}(\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))\text{diag}(\eta(\tilde{x}_\alpha))]s + W(\tilde{x}_\alpha)\nabla h(\tilde{x}_\alpha)\Delta\lambda \\ &= \ -W(\tilde{x}_\alpha)\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho), \end{aligned} \tag{3.20}$$

$$(W(\tilde{x}_\alpha)\nabla h(\tilde{x}_\alpha))^T s = -h(\tilde{x}_\alpha). \tag{3.21}$$

Notice that, Eqs (3.20) and (3.21) are equivalent to the first-order necessary conditions of the following the sequential quadratic programming problem

$$\begin{aligned} &\text{minimize} \quad \ell_j(\tilde{x}_\alpha, \lambda; \rho) + (W(\tilde{x}_\alpha)\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))^T s + \tfrac{1}{2}s^T Bs \\ &\text{subject to} \quad h(\tilde{x}_\alpha) + (W(\tilde{x}_\alpha)\nabla h(\tilde{x}_\alpha))^T s = 0, \end{aligned} \tag{3.22}$$

where

$$B = G(\tilde{x}_\alpha) + \rho W(\tilde{x}_\alpha)\nabla g(\tilde{x}_\alpha)Y(\tilde{x}_\alpha)\nabla g(\tilde{x}_\alpha)^T W(\tilde{x}_\alpha), \tag{3.23}$$

and

$$G(\tilde{x}_\alpha) = W(\tilde{x}_\alpha)H(\tilde{x}_\alpha)W(\tilde{x}_\alpha) + \text{diag}(\nabla_{\tilde{x}_\alpha}\ell_j(\tilde{x}_\alpha, \lambda; \rho))\text{diag}(\eta(\tilde{x}_\alpha)). \tag{3.24}$$

Although Newton's method converges quadratically to a stationary point under reasonable assumptions, it may not converge at all if the starting point is far away from the solution. To guarantee convergence from any starting point, a trust-region globalization strategy is used. The trust-region globalization strategy can induce strong global convergence. It is more robust when it deals with rounding errors. It does not require the Hessian of the objective function must be positive definite or the objective function of the model must be convex. Also, some criteria are used to test whether the trial step is acceptable or not. If it is not acceptable, then the subproblem must be resolved with a reduced trust-region radius.

In the following section, we present the main steps of the trust-region technique for solving the problem (3.22).

### 3.3. Outline of the trust-region strategy

Trust-region strategy is a very successful approach to ensure global convergence to the stationary point from any starting point. The trust-region subproblem which is associated with problem 3.22 is the following

$$\begin{array}{ll} \text{minimize} & \ell_j(\tilde{x}_{\alpha_k}, \lambda_k; \rho_k) + (W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k; \rho_k))^T s + \frac{1}{2} s^T B_k s \\ \text{subject to} & h_k + (W_k \nabla h_k)^T s = 0, \\ & \|s\| \leq \delta_k, \end{array} \tag{3.25}$$

where $\delta_k$ is the radius of the trust-region.

Notice that, in subproblem 3.25 there may be no intersecting points between the linearized constraints $h_k + (W_k \nabla h_k)^T s = 0$ and the inequality constraint $\|s\| \leq \delta_k$, see [7]. Byrd-Omojokun [3, 27] has overcome this difficulty by decomposing the trial step $s_k$ into two orthogonal components. The first component is the normal component $s_k^n$ to improve the feasibility and the second component is the tangential component$s_k^t$ to improve optimality.

- **To obtain the normal component $s_k^n$**

The normal component $s_k^n$ is computed by solving the following trust-region subproblem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|h_k + (W_k \nabla h_k)^T s^n\|^2 \\ \text{subject to} & \|s^n\| \leq \zeta \delta_k, \end{array} \tag{3.26}$$

for some $\zeta \in (0, 1)$.

In the proposed method, a dogleg technique is used to approximate the solution curve of subproblem 3.26 by a piecewise linear function connecting the Cauchy point and Newton point. The main steps for the dogleg method are clarified in the following algorithm

**Algorithm 3.1.** *(To compute $s_k^n$)*

***Step 1.*** *Evaluate $t_k^{ncp}$ as follows*

$$t_k^{ncp} = \begin{cases} \frac{\|W_k \nabla h_k h_k\|^2}{\|(W_k \nabla h_k)^T W_k \nabla h_k h_k\|^2} & \text{if } \frac{\|W_k \nabla h_k h_k\|^3}{\|(W_k \nabla h_k)^T W_k \nabla h_k h_k)\|^2} \leq \delta_k \\ & \text{and } \|(W_k \nabla h_k)^T W_k \nabla h_k h_k)\| > 0, \\ \frac{\delta_k}{\|W_k \nabla h_k h_k\|} & \text{otherwise.} \end{cases} \tag{3.27}$$

***Step 2.*** *Compute the normal Cauchy step $s^{ncp} = -t_k^{ncp} W_k \nabla h_k h_k$.*
***Step 3.*** *If $\|s^{ncp}\| = \delta_k$, then set $s_k^n = s^{ncp}$.*
*Else*
*If $W_k \nabla h_k h_k + W_k \nabla h_k \nabla h_k^T W_k s^{ncp} = 0$, then set $s_k^n = s^{ncp}$.*
*Else, solving the following subproblem to compute Newton step $s^{nlf}$*

$$\textit{minimize} \quad \frac{1}{2}\|h_k + W_k \nabla h_k^T s^{nlf}\|^2.$$

*If $\|s^{nlf}\| \leq \delta_k$, then set $s_k^n = s^{nlf}$.*
*Else, compute $s_k^n$ by dogleg between $s^{ncp}$ and $s^{nlf}$.*
*End if; End if; End if.*

- **To obtain the tangential step $s_k^t$**

Let $q(W_k s)$ be the quadratic form of the function (3.4) and defined as follows

$$q(W_k s) = \ell_j(\tilde{x}_{\alpha_k}, \lambda_k; \rho_k) + (W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k; \rho_k))^T s + \frac{1}{2} s^T B_k s. \tag{3.28}$$

Then $\nabla q_k(W_k s_k^n) = W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k; \rho_k) + B_k s_k^n$.

Once obtaining the normal component $s_k^n$, the following subproblem is used to obtain the tangential step which is defined by $s_k^t = Z_k \bar{s}_k^t$ such that $Z_k$ is a matrix whose columns form a basis for the null space of $(W_k \nabla h_k)^T$.

$$\begin{array}{ll} \text{minimize} & [Z_k^T \nabla q_k(W_k s_k^n) + B_k s_k^n]^T \bar{s}^t + \frac{1}{2} \bar{s}^{t^T} Z_k^T B_k Z_k \bar{s}^t \\ \text{subject to} & \|Z_k \bar{s}^t\| \le \Delta_k, \end{array} \tag{3.29}$$

where $\Delta_k = \sqrt{\delta_k^2 - \|s_k^n\|^2}$.

Again, the dogleg technique is used to solve subproblem 3.29. The main steps for the dogleg method to obtain $s_k^t$ are clarified in the following algorithm

**Algorithm 3.2.** *(To compute $s_k^t$)*

***Step 1.*** *Compute $t_k^{tcp}$ as follows*

$$t_k^{tcp} = \begin{cases} \frac{\|Z_k^T \nabla q_k(W_k s_k^n)\|^2}{(Z_k^T \nabla q_k(W_k s_k^n))^T \bar{B}_k Z_k^T \nabla q_k(W_k s_k^n)} & \text{if } \frac{\|Z_k^T \nabla q_k(W_k s_k^n)\|^3}{(Z_k^T \nabla q_k(W_k s_k^n))^T \bar{B}_k Z_k^T \nabla q_k(W_k s_k^n)} \le \Delta_k \\ & \text{and } (Z_k^T \nabla q_k(W_k s_k^n))^T \bar{B}_k Z_k^T \nabla q_k(W_k s_k^n) > 0, \\ \frac{\Delta_k}{\|Z_k^T \nabla q_k(W_k s_k^n)\|} & \text{otherwise}, \end{cases} \tag{3.30}$$

*such that $\bar{B}_k = Z_k^T B_k Z_k$.*
***Step 2.*** *Compute the Cauchy step $\bar{s}^{tcp} = -t_k^{tcp} Z_k^T \nabla q_k(W_k s_k^n)$.*
***Step 3.*** *If $\|\bar{s}^{tcp}\| = \Delta_k$, then set $s_k^t = Z_k \bar{s}_k^{tcp}$.*
*Else,*
*If $Z_k^T \nabla q_k(W_k s_k^n) + \bar{B}_k \bar{s}^{tcp} = 0$, then set $s_k^t = Z_k s^{tcp}$.*
*Else, solve the following subproblem to obtain Newton step $\bar{s}^{tlf}$*

$$\textit{minimize} \quad [Z_k^T \nabla q_k(W_k s_k^n)]^T \bar{s}^{tlf} + \tfrac{1}{2} \bar{s}^{tlf^T} Z_k^T B_k Z_k \bar{s}^{tlf}.$$

*If $\|s^{tlf}\| \le \Delta_k$, then set $s_k^t = Z_k \bar{s}^{tlf}$.*
*Else, compute $s_k^t$ by dogleg between $s^{tcp}$ and $\bar{s}^{tlf}$.*
*End if; End if; End if.*

Once obtaining $x_{k+1} = x_k + W_k s_k$, we need to restrict it in $\boldsymbol{D}$ to ensure that the diagonal matrix $W(x)$ nonsingular. So, we need the damping parameter $\psi_k$.

- **To obtain the damping parameter $\psi_k$**

The damping parameter $\psi_k$ which is needed to ensure $x_{k+1} \in \boldsymbol{intD}$ is evaluated as follows.

$$\psi = \min\{\min_i\{c_k^{(i)}, \sigma_k^{(i)}\}, 1\}, \tag{3.31}$$

where

$$c_k^{(i)} = \begin{cases} \frac{\tilde{a}_\alpha^{(i)} - \tilde{x}_{\alpha_k}^{(i)}}{W_k^{(i)} s_k^{(i)}}, & \text{if } \tilde{a}_\alpha^{(i)} > -\infty \text{ and } W_k^{(i)} s_k^{(i)} < 0 \\ 1, & \text{otherwise,} \end{cases}$$

and

$$\sigma_k^{(i)} = \begin{cases} \frac{\tilde{b}_\alpha^{(i)} - \tilde{x}_{\alpha_k}^{(i)}}{W_k^{(i)} s_k^{(i)}}, & \text{if } \tilde{b}_\alpha^{(i)} < \infty \text{ and } W_k^{(i)} s_k^{(i)} > 0 \\ 1, & \text{otherwise.} \end{cases}$$

To check whether the scaled step $\psi_k W_k s_k$ will be accepted or not, we need a merit function that ties the objective function and the constraints in such a way that progress in the merit function means progress in solving the problem. The following augmented Lagrangian function is used as a merit function, see [20].

$$\Phi(\tilde{x}_\alpha, \lambda; \rho; r) = f(\tilde{x}_\alpha) + \lambda^T h(\tilde{x}_\alpha) + \frac{\rho}{2}\|Y(\tilde{x}_\alpha) g(\tilde{x}_\alpha)\|^2 + r\|h(\tilde{x}_\alpha)\|^2, \tag{3.32}$$

where $r$ represents the penalty parameter.

To test the scaled step, we need to solve the following subproblem to estimate the Lagrange multiplier vector $\lambda_{k+1}$

$$\text{minimize } \|\nabla f_{k+1} + \nabla h_{k+1}\lambda + \rho_k \nabla g_{k+1} Y_{k+1} g_{k+1}\|^2. \tag{3.33}$$

To check whether the point $(\tilde{x}_{\alpha_{k+1}}, \lambda_{k+1})$, will be accepted in the next iterate or not we need to define the following actual reduction and the predicted reduction.

The actual reduction $Ared_k$ in the merit function 3.32 in moving from $(\tilde{x}_{\alpha_k}, \lambda_k)$ to $(\tilde{x}_{\alpha_k} + s_k, \lambda_{k+1})$ is defined as

$$Ared_k = \Phi(\tilde{x}_{\alpha_k}, \lambda_k, ; \rho_k; r_k) - \Phi(\tilde{x}_{\alpha_k} + \psi_k W_k s_k, \lambda_{k+1}; \rho_k; r_k).$$

$Ared_k$ can be written as,

The predicted reduction $Pred_k$ in the merit function 3.32 is defined as follows

$$\begin{aligned} Pred_k \;=\; & -(W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k))^T \psi_k s_k - \frac{1}{2}\psi_k^2 s_k^T G_k s_k + \frac{\rho_k}{2}[\|Y_k g_k\|^2 - \|Y_k(g_k + (W_k \nabla g_k)^T \psi_k s_k)\|^2] \\ & + r_k[\|h_k\|^2 - \|h_k + (W_k \nabla h_k)^T \psi_k s_k\|^2]. \end{aligned} \tag{3.34}$$

The predicted reduction can be written as

$$Pred_k \;=\; q_k(0) - q_k(W_k \psi_k s_k) + r_k[\|h_k\|^2 - \|h_k + (W_k \nabla h_k)^T \psi_k s_k\|^2], \tag{3.35}$$

where

$$q_k(W_k \psi_k s_k) = \ell_j(\tilde{x}_{\alpha_k}, \lambda_k) + (W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k))^T \psi_k s_k + \frac{1}{2}\psi_k^2 s_k^T G_k s_k + \frac{\rho_k}{2}\|Y_k(g_k + (W_k \nabla g_k)^T \psi_k s_k)\|^2. \tag{3.36}$$

After computing the scaled step and updating the Lagrange multiplier, the penalty parameter is updated to ensure that $Pred_k \geq 0$.

- **To update the penalty parameter** $r_k$

To ensure that $Pred_k \geq 0$, we need to update the penalty parameter $r_k$ by using the following algorithm

**Algorithm 3.3.** *(To update $r_k$)*
*If*

$$Pred_k \leq \frac{r_k}{2}[||h_k||^2 - ||h_k + (W_k \nabla h_k)^T \psi_k s_k||^2], \tag{3.37}$$

*set*

$$r_k = \frac{2[q_k(W_k \psi_k s_k) - q_k(0) + \Delta\lambda_k^T(h_k + (W_k \nabla h_k)^T \psi_k s_k)]}{||h_k||^2 - ||h_k + (W_k \nabla h_k)^T \psi_k s_k||^2} + b_0, \tag{3.38}$$

*where $b_0 > 0$ is a small fixed constant.*
*Else, set $r_{k+1} = r_k$.*
*End if.*

For more details, see [12].

The scaled step $W_k \psi_k s_k$ is tested by comparing $Pred_k$ against $Ared_k$ to know whether it is accepted. Also, the radius of the trust region $\delta_k$ must be updated.

- **To test the step** $s_k$ **and update** $\delta_k$

The framework to test the step $s_k$ and update $\delta_k$ is clarified in the following algorithm.

**Algorithm 3.4.** *(To test the step $s_k$ and update $\delta_k$)*
*Choose* $0 < \tau_1 < \tau_2 < 1$, $0 < \hat{\beta}_1 < 1 < \hat{\beta}_2$, *and* $\delta_{min} \leq \delta_0 \leq \delta_{max}$.
*While* $Ared_k < \tau_1 Pred_k$ *or* $Pred_k \leq 0$.
*Set* $\delta_k = \hat{\beta}_1 ||s_k||$.
*Go to algorithms 3.1 and 3.2 to compute a new trial step $s_k$.*
*End while.*
*If* $\frac{Ared_k}{Pred_k} \in [\tau_1, \tau_2)$.
*Then accept the step $s_k$.*
*Set* $\delta_{k+1} = \max(\delta_k, \delta_{min})$.
*End if.*
*If* $\tau_2 \leq \frac{Ared_k}{Pred_k} \leq 1$.
*Set* $\delta_{k+1} = \min\{\delta_{max}, \max\{\delta_{min}, \hat{\beta}_2 \delta_k\}\}$ .
*End if.*

Let $Tpred_k$ be a tangential predicted decrease which is obtained by the tangential component $s_k^t$ and defined as follows

$$Tpred_k = q_k(W_k s_k^n) - q_k(W_k(s_k^n + Z_k \bar{s}_k^t)).$$

In our method, the positive parameter $\rho_k$ must be updated at every iteration.

- **To update the positive parameter** $\rho_k$

To update $\rho_k$, we use the following algorithm

**Algorithm 3.5.** *(To update $\rho_k$)*
*If*

$$\frac{1}{2} Tpred_k \geq \|W_k \nabla g_k Y_k g_k\| \min\{\|W_k \nabla g_k Y_k g_k\|, \delta_k\}, \tag{3.39}$$

*Set $\rho_{k+1} = \rho_k$.*
*Else, set $\rho_{k+1} = 2\rho_k$.*
*End if.*

For more details see, [11, 12].

Finally, the algorithm is terminated when either $\|Z_k^T W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k)\| + \|W_k \nabla g_k Y_k g_k\| + \|h_k\| \leq \varepsilon_1$ or $\|s_k\| \leq \varepsilon_2$, for some $\varepsilon_1, \varepsilon_2 > 0$.

- **A trust-region algorithm**

A formal description of the trust-region algorithm to solve subproblem 3.25 is clarified as follows

**Algorithm 3.6.** *(Trust-region algorithm)*

***Step 0***. *Starting with the point $\tilde{x}_{\alpha_0} \in int\boldsymbol{D}$. Compute the following initial value $Y_0$, $W_0$, $\eta_0$, and $\lambda_0$. Set $\rho_0 = 1$, $r_0 = 1$, and $b_0 = 0.1$.*
*Choose $\theta > 0$, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$. Choose $\delta_{min}$, $\delta_{max}$, and $\delta_0$ such that $\delta_{min} \leq \delta_0 \leq \delta_{max}$.*
*Choose $\hat{\beta}_1$, $\hat{\beta}_2$, $\tau_1$, and $\tau_2$ such that $0 < \hat{\beta}_1 < 1 < \hat{\beta}_2$, and $0 < \tau_1 < \tau_2 < 1$. Set $k = 0$.*
***Step 1***. *If $\|Z_k^T W_k \nabla_{\tilde{x}_\alpha} \ell_j(\tilde{x}_{\alpha_k}, \lambda_k)\| + \|W_k \nabla g_k Y_k g_k\| + \|h_k\| \leq \varepsilon_1$ , then stop the algorithm.*
***Step 2***. *(To evaluate the trial step $s_k$)*
*a) Using algorithm 3.1 to obtain the normal component $s_k^n$.*
*b) Using algorithm 3.2 to obtain the tangential step $\bar{s}_k^t$.*
*c) Set $s_k = s_k^n + Z_k \bar{s}_k^t$.*
***Step 3***. *If $\|s_k\| \leq \varepsilon_2$, then stop.*
***Step 4***. *a) Compute the damping parameter $\psi_k$ using (3.31).*
*b) Set $\tilde{x}_{\alpha_{k+1}} = \tilde{x}_{\alpha_k} + W_k \psi_k s_k$.*
***Step 5***. *Compute $Y_{k+1}$ which is defined by (3.1).*
***Step 6***. *Solve the following subproblem to obtain the lagrange multiplier vector $\lambda_{k+1}$*

$$minimize \ \ \|\nabla f_{k+1} + \nabla h_{k+1} \lambda + \rho_k \nabla g_{k+1} Y_{k+1} g_{k+1}\|^2.$$

***Step 7***. *Using algorithm 3.3 to update the penalty parameter $r_k$.*
***Step 8***. *Using algorithm 3.4 to test the trial step $s_k$ and update $\delta_k$.*
***Step 9***. *Using algorithm 3.5 to update $\rho_k$.*
***Step 10***. *Using (3.13) to obtain $W_{k+1}$ and using (3.16) to obtain $\eta_{k+1}$.*
***Step 11***. *Set $k = k + 1$ and go to Step 1.*

A trust-region algorithm 3.6 is proved theoretically in [11].

The main steps for solving the continuous static games with fuzzy cost functions and fuzzy conditions 2.3 are clarified in the following algorithm.

*3.4. An active-set interior-point trust-region algorithm*

The framework to solve the continuous static games with fuzzy cost functions and fuzzy conditions 2.3 is summarized in the following algorithm.

**Algorithm 3.7.** *(An active-set interior-point trust-region algorithm):*
***Step 1)*** *Use $\alpha$-level, $\alpha \in [0, 1]$ to restructure problem 2.3 to form 2.4.*
***Step 2)*** *Dividing problem 2.4 into two problems, the lower problem 2.5 and the upper problem 2.6.*
***Step 3)*** *Using the active set strategy with Newton's interior point method and trust region algorithm 3.6 to solve the lower problem 2.5 for all $j = 1, \cdots, p$.*
***Step 4)*** *Using the active set strategy with Newton's interior point method and trust region algorithm 3.6 to solve the upper problem 2.6 for all $j = 1, \cdots, p$.*

## 4. Numerical results

In this section, we will consider the following continuous static game with four players and fuzzy rough

$$\begin{array}{ll}
\min & \tilde{f}_1 = -\tilde{5}\tilde{t}_1 + \tilde{1}\tilde{v}_1 \\
\min & \tilde{f}_2 = -\tilde{3}\tilde{t}_2 + \tilde{1}\tilde{v}_2 \\
\min & \tilde{f}_3 = -\tilde{8}\tilde{t}_3 + \tilde{1}\tilde{v}_3 \\
\min & \tilde{f}_4 = -\tilde{5}\tilde{t}_4 + \tilde{1}\tilde{v}_4 \\
S.T. & -\tilde{3}\tilde{t}_1 + \tilde{1}\tilde{v}_1 - \tilde{1}\tilde{v}_1^2 - \tilde{1}\tilde{t}_1\tilde{v}_2 = \tilde{0} \\
& -\tilde{2}\tilde{t}_2 + \tilde{1}\tilde{v}_2 - \tilde{1}\tilde{v}_2^2 - \tilde{2}\tilde{t}_2\tilde{v}_1 = \tilde{0} \\
& -\tilde{5}\tilde{t}_3 + \tilde{1}\tilde{v}_3 - \tilde{1}\tilde{v}_3^2 - \tilde{2}\tilde{t}_3\tilde{v}_4 = \tilde{0} \\
& -\tilde{4}\tilde{t}_4 + \tilde{1}\tilde{v}_4 - \tilde{1}\tilde{v}_4^2 - \tilde{4}\tilde{t}_4\tilde{v}_3 = \tilde{0} \\
& \tilde{t}_1 + \tilde{t}_2 \leq \tilde{1} \\
& \tilde{v}_1 \geq \tilde{0}, \quad \tilde{v}_2 \geq \tilde{0}, \quad \tilde{v}_3 \geq \tilde{0}, \quad \tilde{v}_4 \geq \tilde{0} \\
& \tilde{t}_1 \geq \tilde{0}, \quad \tilde{t}_2 \geq \tilde{0}, \quad \tilde{t}_3 \geq \tilde{0}, \quad \tilde{t}_4 \geq \tilde{0},
\end{array}$$

where $\tilde{0} = (\alpha, 2-\alpha)$,$\tilde{1} = (\alpha+1, 3-\alpha)$, $\tilde{2} = [(\alpha+2, 4-\alpha)$, $\tilde{3} = (\alpha+3, 5-\alpha)$, $\tilde{4} = (\alpha+4, 6-\alpha)$, $\tilde{5} = (\alpha+5, 7-\alpha)$, $\tilde{8} = (\alpha+8, 10-\alpha)$.
The above test problem can be divided into the lower game and upper game for every player as follows
**For every player: the lower game**

$$\begin{array}{ll}
\min & \tilde{f}_1^l = -(\alpha+5)\tilde{t}_1^l + (\alpha+1)\tilde{v}_1^l \\
\min & \tilde{f}_2^l = -(\alpha+3)\tilde{t}_2^l + (\alpha+1)\tilde{v}_2^l \\
\min & \tilde{f}_3^l = -(\alpha+8)\tilde{t}_3^l + (\alpha+1)\tilde{v}_3^l \\
\min & \tilde{f}_4^l = -(\alpha+5)\tilde{t}_4^l + (\alpha+1)\tilde{v}_4^l \\
S.T. & -(\alpha+3)\tilde{t}_1^l + (\alpha+1)\tilde{v}_1^l - (\alpha+1)\tilde{v}_1^{2^l} - (\alpha+1)\tilde{t}_1^l\tilde{v}_2^l = \alpha \\
& -(\alpha+2)\tilde{t}_2^l + (\alpha+1)\tilde{v}_2^l - (\alpha+1)\tilde{v}_2^{2^l} - (\alpha+2)\tilde{t}_2^l\tilde{v}_1^l = \alpha \\
& -(\alpha+5)\tilde{t}_3^l + (\alpha+1)\tilde{v}_3^l - (\alpha+1)\tilde{v}_3^{2^l} - (\alpha+2)\tilde{t}_3^l\tilde{v}_4^l = \alpha \\
& -(\alpha+4)\tilde{t}_4^l + (\alpha+1)\tilde{v}_4^l - (\alpha+1)\tilde{v}_4^{2^l} - (\alpha+4)\tilde{t}_4^l\tilde{v}_3^l = \alpha \\
& \tilde{t}_1^l + \tilde{t}_2^l \leq (\alpha+1) \\
& \tilde{v}_1^l \geq \alpha, \quad \tilde{v}_2^l \geq \alpha, \quad \alpha \leq \tilde{v}_3^l \leq \alpha+1, \quad \alpha \leq \tilde{v}_4^l \leq \alpha+1 \\
& \tilde{t}_1^l \geq \alpha, \quad \tilde{t}_2^l \geq \alpha, \quad \tilde{t}_3^l \geq \alpha, \quad \tilde{t}_4^l \geq \alpha.
\end{array}$$

**For every player: the upper game**

$$\begin{aligned}
\min\quad & \tilde{f}_1^u = -(7-\alpha)\tilde{t}_1^u + (3-\alpha)\tilde{v}_1^u \\
\min\quad & \tilde{f}_2^u = -(\alpha+3)\tilde{t}_2^u + (\alpha+1)\tilde{v}_2^u \\
\min\quad & \tilde{f}_3^u = -(\alpha+8)\tilde{t}_3^u + (\alpha+1)\tilde{v}_3^u \\
\min\quad & \tilde{f}_4^u = -(\alpha+5)\tilde{t}_4^u + (\alpha+1)\tilde{v}_4^u \\
S.T.\quad & -(5-\alpha)\tilde{t}_1^u + (3-\alpha)\tilde{v}_1^u - (3-\alpha)\tilde{v}_1^{2^u} - (3-\alpha)\tilde{t}_1^u\tilde{v}_2^u = 2-\alpha \\
& -(4-\alpha)\tilde{t}_2^u + (3-\alpha)\tilde{v}_2^u - (3-\alpha)\tilde{v}_2^{2^u} - (4-\alpha)\tilde{t}_2^u\tilde{v}_1^u = 2-\alpha \\
& -(7-\alpha)\tilde{t}_3^u + (3-\alpha)\tilde{v}_3^u - (3-\alpha)\tilde{v}_3^{2^u} - (4-\alpha)\tilde{t}_3^u\tilde{v}_4^u = 2-\alpha \\
& -(6-\alpha)\tilde{t}_4^u + (3-\alpha)\tilde{v}_4^u - (3-\alpha)\tilde{v}_4^{2^u} - (6-\alpha)\tilde{t}_4^u\tilde{v}_3^u = 2-\alpha \\
& \tilde{t}_1^u + \tilde{t}_2^u \le (3-\alpha) \\
& \tilde{v}_1^u \ge 2-\alpha,\quad \tilde{v}_2^u \ge 2-\alpha,\quad 2-\alpha \le \tilde{v}_3^u \le 3-\alpha,\quad 2-\alpha \le \tilde{v}_4^u \le 3-\alpha \\
& \tilde{t}_1^u \ge 2-\alpha,\quad \tilde{t}_2^u \ge 2-\alpha,\quad \tilde{t}_3^u \ge 2-\alpha,\quad \tilde{t}_4^u \ge 2-\alpha.
\end{aligned}$$

Using algorithm 3.7 which was implemented as a MATLAB code and run under MATLAB version 7.10.0.499 (R2010a) to obtain an approximate solution for the lower-game problem and upper game problem for every player at different values of $\alpha$. The approximate solutions at $\alpha = 0; 0.1; 0.3; 0.5; 0.7; 1$ are clarified in Tables 1–6 respectively.

**Table 1.** Results at $\alpha = 0$.

| Player | $\tilde{t}_1$ | $\tilde{t}_2$ | $\tilde{t}_3$ | $\tilde{t}_4$ | $\tilde{v}_1$ | $\tilde{v}_2$ | $\tilde{v}_3$ | $\tilde{v}_4$ | NI |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{f}_1^l = 0.0667$ | .0533 | 0 | .0405 | .0443 | 0.2 | 0 | .3939 | .447 | 5 |
| $\tilde{f}_1^u = 8.6265$ | 2.1164 | 2.2399 | 2 | 2 | 2.0628 | 2.0437 | 2.3197 | 2.3502 | 4 |
| $\tilde{f}_2^l = 0.0417$ | 0 | .0694 | .0405 | .04433 | 0 | 0.16667 | 0.39396 | .44698 | 6 |
| $\tilde{f}_2^u = 4.925$ | 2.0825 | 2.2151 | 2.017 | 2 | 2 | 2.0502 | 2.3145 | 2.347 | 9 |
| $\tilde{f}_3^l = .05625$ | 0.0256 | .11361 | .0305 | 0 | .0990 | .48346 | .1875 | 0 | 5 |
| $\tilde{f}_3^u = 14.174$ | 2.0875 | 2.1765 | 2.110 | 2.175 | 2 | 2.2296 | 2.3093 | 2.2224 | 3 |
| $\tilde{f}_4^l = .0125$ | .06171 | 0.09510 | 0 | .0225 | .31375 | .48937 | 0 | .1 | 5 |
| $\tilde{f}_4^u = 8.2695$ | 2.0857 | 2.1741 | 2.1277 | 2.1769 | 2 | 2.2305 | 2.2102 | 2.3230 | 3 |

**Table 2.** Results at $\alpha = 0.1$.

| Player | $\tilde{t}_1$ | $\tilde{t}_2$ | $\tilde{t}_3$ | $\tilde{t}_4$ | $\tilde{v}_1$ | $\tilde{v}_2$ | $\tilde{v}_3$ | $\tilde{v}_4$ | NI |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{f}_1^l = .59519$ | .16266 | .19521 | .1 | .1 | .21308 | .10005 | .2916 | .10004 | 9 |
| $\tilde{f}_1^u = 8.8823$ | 2.1297 | 2.2943 | 1.976 | 1.9 | 2.004 | 1.9 | 2.144 | 2.211 | 5 |
| $\tilde{f}_2^l = .69613$ | .20653 | .28109 | .1 | .1 | .12612 | .15932 | .35415 | .5128 | 6 |
| $\tilde{f}_2^u = 3.9637$ | 2.0105 | 2.0791 | 1.9 | 2.013 | 1.9 | 2.1462 | 2.1993 | 2.258 | 9 |
| $\tilde{f}_3^l = .19087$ | 0.13537 | .1 | .1 | 0.1 | .6139 | .1005 | .56285 | 0.5499 | 11 |
| $\tilde{f}_3^u = 15.039$ | 2.0485 | 2.1210 | 2.124 | 2.156 | 1.9 | 2.1632 | 2.0664 | 1.9 | 4 |
| $\tilde{f}_4^l = .032766$ | .11581 | 0.1 | 0.1 | .1 | .57804 | .2546 | .4006 | .43385 | 11 |
| $\tilde{f}_4^u = 8.6081$ | 2.0453 | 2.1173 | 2.1406 | 2.1454 | 1.9 | 2.164 | 1.9 | 2.1362 | 4 |

**Table 3.** Results at $\alpha = 0.3$.

| Player | $\tilde{t}_1$ | $\tilde{t}_2$ | $\tilde{t}_3$ | $\tilde{t}_4$ | $\tilde{v}_1$ | $\tilde{v}_2$ | $\tilde{v}_3$ | $\tilde{v}_4$ | NI |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{f}_1^l = 2.3039$ | .5468 | .591 | .3 | .3668 | .457 | .3 | .5242 | .5975 | 4 |
| $\tilde{f}_1^u = 8.6480$ | 2.0468 | 2.1911 | 1.8474 | 1.7 | 1.8762 | 1.7411 | 1.7 | 1.8318 | 7 |
| $\tilde{f}_2^l = 1.4608$ | .51523 | .5975 | .3 | .313 | .3 | .3931 | .4982 | .3 | 5 |
| $\tilde{f}_2^u = 4.796$ | 2.0125 | 2.0920 | 1.8347 | 1.7 | 1.7 | 1.8654 | 1.7 | 1.7419 | 8 |
| $\tilde{f}_3^l = 1.9187$ | .55016 | .50547 | .3 | .3087 | .5572 | .571 | .4394 | .5277 | 3 |
| $\tilde{f}_3^u = 14.725$ | 1.8296 | 1.8613 | 1.9912 | 1.7 | 1.8284 | 1.8318 | 1.7 | 1.7 | 6 |
| $\tilde{f}_4^l = 1.0227$ | .48496 | .5599 | .3205 | .3 | .5576 | .57141 | .48169 | .43636 | 3 |
| $\tilde{f}_1^u = 6.6543$ | 1.8329 | 1.8670 | 1.963 | 1.7 | 1.8006 | 1.7967 | 1.7 | 1.754 | 6 |

**Table 4.** Results at $\alpha = 0.5$.

| Player | $\tilde{t}_1$ | $\tilde{t}_2$ | $\tilde{t}_3$ | $\tilde{t}_4$ | $\tilde{v}_1$ | $\tilde{v}_2$ | $\tilde{v}_3$ | $\tilde{v}_4$ | NI |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{f}_1^l = 3.6487$ | .82681 | .93138 | .5 | .5 | .5992 | .5 | .7633 | .84622 | 6 |
| $\tilde{f}_1^u = 5.962$ | 1.5557 | 1.7086 | 1.5 | 1.5 | 1.6601 | 1.5401 | 1.7324 | 1.9338 | 6 |
| $\tilde{f}_2^l = 2.115$ | .81646 | .8529 | .5 | .5 | .5 | .58013 | .6061 | .8405 | 6 |
| $\tilde{f}_2^u = 3.0869$ | 1.5634 | 1.6605 | 1.5 | 1.5 | 1.5265 | 1.7541 | 1.7580 | 1.9714 | 8 |
| $\tilde{f}_3^l = 3.7693$ | .58543 | .6679 | .5575 | .5 | .86199 | .873 | .6464 | .5288 | 3 |
| $\tilde{f}_3^u = 10.275$ | 1.5 | 1.5741 | 1.5 | 1.5606 | 1.6202 | 1.7860 | 1.5899 | 1.5 | 4 |
| $\tilde{f}_4^l = 1.7199$ | .5016 | .504 | .5082 | .5 | .865 | .8751 | .50005 | .6867 | 4 |
| $\tilde{f}_4^u = 5.627$ | 1.5 | 1.5729 | 1.5 | 1.5646 | 1.6415 | 1.8204 | 1.5 | 1.8172 | 4 |

**Table 5.** Results at $\alpha = 0.7$.

| Player | $\tilde{t}_1$ | $\tilde{t}_2$ | $\tilde{t}_3$ | $\tilde{t}_4$ | $\tilde{v}_1$ | $\tilde{v}_2$ | $\tilde{v}_3$ | $\tilde{v}_4$ | NI |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{f}_1^l = 2.6853$ | .71848 | .76744 | 0.7 | 0.7 | .82947 | 0.7 | .74389 | 1.102 | 6 |
| $\tilde{f}_1^u = 4.936$ | 1.3 | 1.3218 | 1.3119 | 1.3 | 1.4148 | 1.8539 | 1.5761 | 1.816 | 4 |
| $\tilde{f}_2^l = 1.5381$ | .7559 | .78602 | .7 | .7 | .7 | .80597 | 0.7 | 1.0422 | 8 |
| $\tilde{f}_2^u = 3.2477$ | 1.4444 | 1.5664 | 1.3 | 1.3 | 1.3008 | 1.5165 | 1.574 | 1.8183 | 9 |
| $\tilde{f}_3^l = 5.0726$ | .85777 | .88314 | .75304 | .7 | .88897 | .81370 | .86994 | .7 | 4 |
| $\tilde{f}_3^u = 8.9775$ | 1.3592 | 1.4232 | 1.3 | 1.3364 | 1.3676 | 1.5349 | 1.3533 | 1.3 | 4 |
| $\tilde{f}_4^l = 2.4321$ | .8546 | .8849 | .7229 | .7 | .8613 | .77382 | .7 | .9164 | 4 |
| $\tilde{f}_4^u = 5.1238$ | 1.3585 | 1.4259 | 1.3 | 1.3735 | 1.4079 | 1.5645 | 1.3 | 1.5345 | 4 |

**Table 6.** Results at $\alpha = 1$.

| Player | $\tilde{t}_1$ | $\tilde{t}_2$ | $\tilde{t}_3$ | $\tilde{t}_4$ | $\tilde{v}_1$ | $\tilde{v}_2$ | $\tilde{v}\tilde{v}_3$ | $\tilde{v}_4$ | NI |
|---|---|---|---|---|---|---|---|---|---|
| $\tilde{f}_1^l = 3.673$ | 1 | 1.0487 | 1.0113 | 1.201 | 1.1635 | 1.6777 | 1.3718 | 1.6390 | 4 |
| $\tilde{f}_1^u = 3.673$ | 1 | 1.0487 | 1.0113 | 1.201 | 1.1635 | 1.6777 | 1.3718 | 1.6390 | 4 |
| $\tilde{f}_2^l = 2.4998$ | 1.1004 | 1.2685 | 1 | 1 | 1.0297 | 1.2871 | 1.2998 | 1.5464 | 5 |
| $\tilde{f}_2^u = 2.5277$ | 1.1034 | 1.2729 | 1 | 1 | 1.0256 | 1.2819 | 1.2987 | 1.5435 | 5 |
| $\tilde{f}_3^l = 6.7625$ | 1.0723 | 1.1429 | 1 | 1.0526 | 1.1431 | 1.4151 | 1.1187 | 1 | 4 |
| $\tilde{f}_3^u = 6.7625$ | 1.0723 | 1.1429 | 1 | 1.0526 | 1.1431 | 1.4151 | 1.1187 | 1 | 4 |
| $\tilde{f}_4^l = 4.0403$ | 1.0708 | 1.1422 | 1 | 1.1030 | 1.1669 | 1.4325 | 1 | 1.2888 | 4 |
| $\tilde{f}_4^u = 4.0403$ | 1.0708 | 1.1422 | 1 | 1.1030 | 1.1669 | 1.4325 | 1 | 1.2888 | 4 |

In algorithm 3.7, we choose the initial trust-region radius $\delta_0 = max(\|s_0^{ncp}\|, \delta_{min})$, where $\delta_{min} = 10^{-3}$. Also, We choose the maximum trust-region radius $\delta_{max} = 10^3\delta_0$ and the values of the constants $\tau_1 = .25$, $\tau_2 = 0.75$, $\hat{\beta}1 = 0.5$, $\hat{\beta}_2 = 2$, $\varepsilon_1 = 10^{-8}$, $\varepsilon_2 = 10^{-10}$.

Successful termination concerning algorithm 3.7 means that the termination condition of the algorithm is met with $\varepsilon_1 = 10^{-8}$. On the other hand, unsuccessful termination means that the number of iterations is greater than 500, the number of function evaluations is greater than 1000, or the length of the trial step is less than $\varepsilon_2$.

## 5. Conclusions

This paper presented a new technique to solve continuous static games. This technique introduced a novel treatment for multi-player fuzzy continuous static games. This treatment is based on the fact that as well as having a fuzzy number, the fuzziness is applied to the control vectors to deal with high vagueness and imprecision in a continuous static game. The $\alpha$-level set is used to convert the FCSGs to deterministic upper and lower $\alpha$-FCSGs problems. The $\alpha$-Pareto optimal solutions for the deterministic upper and lower $\alpha$-FCSGs problems are obtained by using active-set interior-point trust-region algorithm 3.7. This method converges quadratically to $\alpha$-Pareto optimal solutions from any starting point. A projected Hessian method is used to treat the difficulty of having an infeasible trust-region subproblem where the trial step is decomposed into normal and tangent components and each component is computed by solving a trust-region unconstrained subproblem.

An application to mathematical continuous static games with four players and fuzzy rough with equilibrium constraints is given to clarify the effectiveness of the proposed approach. Numerical results reflect the good behavior of algorithm 3.7 for the lower and upper $\alpha$-FCSGs problems for every player at different values of $\alpha$.

In applying this methodology we cope with some known difficulties in handling such problems, as

- Using the active-set strategy reduces upper and lower $\alpha$-FCSGs problems to equivalent equality constrained optimization problems which allow using the methods that are used to solve equality constrained optimization problems.
- Using Newton's interior-point technique warrants the converges quadratically to a stationary point.

- A trust-region globalization strategy can induce strong global convergence and it is more robust when they deal with rounding errors. It is a very important technique for solving unconstrained and constrained optimization problems.

## Conflict of interest

All authors declare that they have no conflicts of interest.

## References


1. E. E. Ammar, Stability of multiobjective NlP problems with fuzzy parameters in the objectives and constraints functions, *Fuzzy Set. Syst.*, **90** (1997), 225–234. https://doi.org/10.1016/s0165-0114(96)00134-0
2. R. J. Aumann, *Game theory*, The new Palgrave, 1989, 1–53. https://doi.org/10.1007/978-1-349-20181-5_1
3. R. Byrd, Robust trust-region methods for nonlinearly constrained optimization, In: *Third SIAM conference on optimization*, Houston, 1987.
4. A. Dhingra, S. Rao, A cooperative fuzzy game theoretic approach to multiple objective design optimization, *Eur. J. Oper. Res.*, **83** (1995), 547–567. https://doi.org/10.1016/0377-2217(93)E0324-Q
5. I. Das, An interior point algorithm for the general nonlinear programming problem with trust region globalization, Institute for Computer Applications in Science and Engineering Hampton, VA, 1996.
6. J. Dennis, M. El-Alem, K. Williamson, A trust-region approach to nonlinear systems of equalities and inequalities, *SIAM J. Optimiz.*, **9** (1999), 291-315. https://doi.org/10.1137/S1052623494276208
7. J. Dennis, M. Heinkenschloss, L. Vicente, Trust-region interior-point SQP algorithms for a class of nonlinear programming problems, *SIAM J. Control Optim.*, **36** (1998), 1750–1794. https://doi.org/10.1137/S036012995279031
8. Y. A. Abonaga, M Shokry, M. F. Zidan, Nash-equilibrium solutions for fuzzy rough continuous static games, *IJETST*, **7** (2020), 6950–6965. https://doi.org/10.18535//ijetst/v7i10.01
9. B. El-Sobky, A global convergence theory for an active trust region algorithm for solving the general nonlinear programming problem, *Appl. Math. comput.*, **144** (2003), 127–157. https://doi.org/10.1016/S0096-3003(02)00397-1
10. B. El-Sobky, A Multiplier active-set trust-region algorithm for solving constrained optimization problem, *Appl. Math. Comput.*, **219** (2012), 928–946. https://doi.org/10.1016/j.amc.2012.06.072
11. B. El-Sobky, An interior-point penalty active-set trust-region algorithm, *J. Egypt. Math. Soc.*, **24** (2016), 672–680. https://doi.org/10.1016/j.joems.2016.04.003
12. B. El-Sobky, An active-set interior-point trust-region algorithm, *Pac. J. Optim.*, **14** (2018), 125–159.

13. B. El-Sobky, A. Abotahoun, An active-set algorithm and a trust-region approach in constrained minimax problem, *Comp. Appl. Math.*, **37** (2018), 2605–2631. https://doi.org/10.1007/s40314-017-0468-3

14. B. El-Sobky, A. Abotahoun, A trust-region algorithm for solving mini-max problem, *J. Comput. Math.*, **36** (2018), 776–791. https://doi.org/10.4208/jcm.1705-m2016-0735

15. B. El-Sobky, Y. Abo-Elnaga, A. Mousa, A. El-Shorbagy, Trust-region based penalty barrier algorithm for constrained nonlinear programming problems: An application of design of minimum cost canal sections, *Mathematics*, **9** (2021), 1551. https://doi.org/10.3390/math9131551

16. B. El-Sobky, G. Ashry, An interior-point trust-region algorithm to solve a nonlinear bilevel programming problem, *AIMS Math.*, **7** (2022), 5534–5562. https://doi.org/10.3934/math.2022307

17. B. El-Sobky, G. Ashry, An active-set Fischer-Burmeister trust-region algorithm to solve a nonlinear bilevel optimization problem, *Fractal Fract.*, **6** (2022), 412. https://doi.org/10.3390/fractalfract6080412

18. B. El-Sobky, G. Ashry, Y. Abo-Elnaga, An active-set with barrier method and trust-region mechanism to solve a nonlinear Bilevel programming problem, *AIMS Math.*, **7** (2022), 16112–16146. https://doi.org/10.3934/math.2022882

19. J. A. Goguen, L-fuzzy sets, *J. Math. Anal. Appl.*, **18** (1967), 145–174. https://doi.org/10.1016/0022-247X(67)90189-8

20. M. R. Hestenes, Multiplier and gradient methods, *J. Optim. Theory Appl.*, **4** (1969), 303–320. https://doi.org/10.1007/BF00927673

21. M. G. Iskander, Using different dominance criteria in stochastic fuzzy linear multiobjective programming: A case of fuzzy weighted objective function, *Math. Comput. Model.*, **37** (2003), 167–176. https://doi.org/10.1016/S0895-7177(03)80012-2

22. H. A. Khalifa, R. A. Zeineldin, An interactive approach for solving fuzzy cooperative continuous static games, *Int. J. Comput. Appl.*, **113** (2015), 16–20.

23. H. Khalifa, Study on cooperative continuous static games under fuzzy environment, *Int. J. Comput. Appl. Found. Comput. Sci.*, **13** (2019), 20–29.

24. M. Kassem, E. Ammar, Stability of multiobjective nonlinear programming problems with fuzzy parameters in the constraints, *Fuzzy Set. Syst.*, **74** (1995), 343–351. https://doi.org/10.1016/0165-0114(94)00344-7

25. N. Li, D. Xue, W. Sun, J. Wang, A stochastic trust-region method for unconstrained optimization problems, *Math. Probl. Eng.*, **2019** (2019), 8095054. https://doi.org/10.1155/2019/8095054

26. L.Niu, Y. Yuan, A new trust region algorithm for nonlinear constrained optimization, *J. Comput. Math.*, **28** (2010), 72–86. https://doi.org/10.4208/jcm.2009.09-m2924

27. E. Omojokun, Trust-region strategies for optimization with nonlinear equality and inequality constraints, Department of Computer Science, University of Colorado, 1989.

28. M. Osman, A. H. El-Banna, Stability of multiobjective nonlinear programming problems with fuzzy parameters, *Math. Comput. Simulat.*, **35** (1993), 321–326. https://doi.org/10.1016/0378-4754(93)90062-Y

29. E. Rasmusen, *Games and information*, Basil Blackwell, 2000.

30. M. Sakawa, H. Yano, Interactive decision-making for multiobjective programming problems with fuzzy parameters, In: *Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty*, Dordrecht: Springer, 1990. https://doi.org/10.1007/978-94-009-2111-5_10

31. M. Sakawa, H. Yano, An interactive fuzzy satisfying method for multiobjective non-linear programming problems with fuzzy parameters, *IFAC Proceedings Volumes*, **20** (1987), 437–442. https://doi.org/10.1016/s1474-6670(17)55745-6

32. H. Tanaka, H. Ichihashi, K. Asai, Formulation of fuzzy linear programming problem by fuzzy objective function, *J. Oper. Res. Soc. JPN*, **27** (1984), 178–190. https://doi.org/10.15807/jorsj.27.178

33. J. Von Neumann, On the theory of parlor games, *Math. Ann.*, **100** (1928), 295–320.

34. T. Vincent, W. Grantham, Optimality in parametric systems, *J. Appl. Mech.*, **50** (1983), 476. https://doi.org/10.1115/1.3167074

35. X. Wang, Y. Yuan, A trust region method based on a new affine scaling technique for simple bounded optimization, *Optim. Method. Softw.*, **28** (2013), 871–888. https://doi.org/10.1080/10556788.2011.622378

36. X. Wang, Y. Yuan, An augmented Lagrangian trust region method for equality constrained optimization, *Optim. Method. Softw.*, **30** (2015), 559–582. https://doi.org/10.1080/10556788.2014.940947

37. L. A. Zadeh, Fuzzy sets, *Inf. Control*, **8** (1965), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)