Optimization under Fuzzy Constraints: From a Heuristic Algorithm to an Algorithm That Always Converges*

Vladik Kreinovich[1] [0000-0002-1244-1650] and
Juan Carlos Figueroa-García[2] [0000-0001-5544-5937]

[1] University of Texas at El Paso, El Paso TX 79968, USA, vladik@utep.edu
http://www.cs.utep.edu/vladik
[2] Universidad Distrital, Bogotá D.C, Colombia, jcfigueroag@udistrital.edu.co

Abstract. An efficient iterative heuristic algorithm has been used to implement Bellman-Zadeh solution to the problem of optimization under fuzzy constraints. In this paper, we analyze this algorithm, explain why it works, show that there are cases when this algorithm does not converge, and propose a modification that always converges.

Keywords: Optimization · Fuzzy constraints · Bellmna-Zadeh approach · Convergence

1 Formulation of the Problem

Need to select the best alternative. In many practical situations, we want to select the best of the possible alternatives x.

To use mathematical and computational techniques in solving such problems, we need describe this problem in precise terms. For this, we need to describe what is meant by "the best", and what is meant by "possible alternatives".

What does "the best" means. "The best" can usually be described in numerical form: we have an objective function $f(x)$ such that the larger the value of this function, the better the alternative. For example, in economics problems, we want to maximize profit.

In some cases, the better alternative corresponds to the smallest value of the corresponding function $g(x)$. For example, in economics, we may want to minimize the cost $g(x)$. In transportation problems, we may want to minimize travel time $g(x)$. Such problems can be easily reformulated in the maximization terms if we take $f(x) \stackrel{\text{def}}{=} -g(x)$.

The objective function is usually continuous, even smooth. Tiny changes in the selected alternative usually do not change the output much, so we expect that the values of the objective function $f(x)$ should not change much either.

* This work was supported in part by US National Science Foundation grant HRD-1242122 (Cyber-ShARE Center).

Thus, we expect the objective function to be continuous – or, if we interpret "not much" as bounded by a certain constant times Δx, as Lipschitz continuous, and thus, as differentiable almost everywhere. In practice, the objective function is usually smooth.

Which alternatives are possible? An alternative is possible if it satisfies certain constraints. Usually, these constraints are equalities $g_i(x) = 0$ or inequalities $h_j(x) \geq 0$. For example, in chemical manufacturing, constraints are that the number of potential pollutants $p(x)$ released into the environment does not exceed some threshold t: $p(x) \leq t$. This constraint can be described as $h_1(x) \geq 0$, where $h_1(x) \stackrel{\text{def}}{=} t - p(x)$.

In principle, we can have more general constraints. Let us denote the set of all possible alternatives – i.e., of all the alternatives that satisfy all the constraints – by $S \subset R^N$ for an appropriate N.

It is reasonable to assume that constraints describe a closed set. Selecting an alternative means selecting the parameters that describe this alternative. For example, in control applications, we select the values of the control parameters. For a car, we can select the acceleration and the torque, etc.

In practice, we can set up the desired values only with some accuracy – and we can only measure how well we have set them with some accuracy. As a result, if we have a sequence of possible alternative $x_1, x_2, \ldots$ that converges to a limit alternative x, then, for any desired implementation and/or measuring accuracy $\varepsilon > 0$, there is a possible state x_n which is ε-close to x and is, thus, practically indistinguishable from the alternative x. Since we cannot distinguish the limit alternative x from possible alternatives, no matter how much we increase our accuracy, it makes sense to assume that the limit alternative x is also possible.

Under this assumption, the set S of all possible alternatives has the property that is $x_i \in S$ for all i and $x_i \to x$, then $x \in S$. In mathematical terms, this means that the set S is *closed.*

Comment. This closeness assumption is the main reason why in traditional optimization problems, we consider constraints of the non-strict inequality type $h_j(x) \geq 0$ but not constraints of the strict inequality type $h_j(x) > 0$. Indeed, non-strict inequalities are preserved in the limit, while strict inequalities are not necessarily preserved: e.g., $2^{-i} > 0$, $2^{-i} \to 0$, but $0 \not> 0$.

In practice, the set of possible alternatives is always bounded. In practice, the values of all the quantities are bounded. For example, the speeds are limited by the speed of light, the distances for Earth travel are bounded by the Earth's size, accelerations are bounded by our technical abilities, etc. Thus, in practice, the set S of possible alternatives is always bounded.

Mathematical conclusion: the set of possible alternatives is a compact set. Since the set S is closed and bounded, it is a compact set. This means, in particular, that for every continuous function $F(x)$ on this set, there exists an alternative x_{opt} at which this function attains its maximum, i.e., at which $F(x_{\text{opt}}) = \max\limits_{x \in S} F(x)$.

Resulting formulation: optimization under constraints. Thus, the above practical problem takes the following form: maximize the objective function $f(x)$ under the constraint that $g_1(x) = 0, \ldots, g_m(x) = 0$, $h_1(x) \geq 0, \ldots, h_m(x) \geq 0$ (or, more generally, that $x \in S$ for some compact set S).

Since the set S of possible alternatives is compact, for continuous objective functions $f(x)$, there is always an alternative x_{opt} that solves this problem.

Algorithms for optimization under constraints. There exist many efficient algorithms for optimization under constraints; see, e.g., [9].

The most well-known methods are based on Lagrange multiplier techniques, according to which maximizing a function $f(x)$ under the constraints $g_1(x) = 0$, $\ldots$, $g_m(x) = 0$ can be reduced to the unconstrained problem of maximizing the auxiliary function

$$f(x) + \lambda_1 \cdot g_1(x) + \ldots + \lambda_m \cdot g_m(x),$$

where the auxiliary constants λ_i (known as *Lagrange multipliers*) can be determined by the condition that the resulting solution x satisfies all m equality constraints $g_i(x) = 0$, $1 \leq i \leq m$.

Thus, equality-type constraint optimization problem can be reduced to an unconstraint optimization problem, and for such problems, many efficient optimization algorithms are known; see, e.g., [9].

If some of the constraints are inequalities, then the constrained maximum is attained when some of them are equalities, and some are not. In this case, we need to consider all 2^n possible subsets $I \subseteq \{1, \ldots, n\}$, and for each of these subsets, look for local maxima of the auxiliary function

$$f(x) + \sum_{i=1}^{m} \lambda_i(x) \cdot g_i(x) + \sum_{j \in I} \lambda'_j \cdot h_j(x).$$

For each I, we select the coefficients λ_i and λ'_j from the condition that $g_i(x) = 0$ for all i and $h_j(x) = 0$ for all $j \in I$; then, we check that $h_j(x) > 0$ for all $j \notin I$.

This procedure leads to several different possible maxima x; out of them, we select the one for which the value of the objective function $f(x)$ is the largest.

Need for imprecise ("fuzzy") constraints. In many practical situations, constraints are imprecise. For example, when we select a hotel, we want it to be "comfortable" and/or "not very expensive". These are not precise terms: in many cases, we are not 100% sure what "not very expensive" means.

Fuzzy logic: a way to describe imprecise ("fuzzy") constraints. Situations where we have information described in such imprecise ("fuzzy") natural-language terms are ubiquitous. To take this imprecise information into account when using computers, it is necessary to describe this information in precise terms. Such a description was proposed by Lotfi Zadeh and is now known as *fuzzy logic*; see, e.g., [2, 4, 7, 8, 10, 11].

The main idea behind fuzzy logic is that to describe an imprecise property like "not very expensive", we ask an expert, for each possible value of the corresponding quantity q (i.e., of the hotel daily rate) describe, on a scale from

0 to 1, the degree to which this amount is, in the opinion of this expert, not very expensive. For example, for a hotel in El Paso, Texas, a daily rate of \$140 would definitely not satisfy this property, so we assign degree 0. The daily rate of \$35 would definitely satisfy this property, so we assign degree 1. For some intermediate values like \$80, we will assign intermediate degrees.

Instead of using the scale from 0 to 1, we can alternatively use a scale, e.g., from 0 to 10, and then divide the result by 10. For example, if an expert estimates his/her degree as 7 on a scale from 0 to 10, we get the degree $7/10 = 0.7$.

As a result, as a description of the desired imprecise property, we get a function that assigns, to each possible value q of this quantity, the degree $\mu(q)$ to which this value satisfies this property. This function is known as a *membership function*, or, alternatively, as a *fuzzy set*.

Tiny changes in x usually only slightly change the degree to which x is possible. So, similarly to our conclusion that the objective function be continuous and even smooth, we conclude that the membership function should also be continuous – and, if possible, smooth.

"And"-operations. In most practical situations, we have several different constraints that describe different quantities. For example, when selecting a hotel, we want a hotel which is not very expensive (which is a limitation on the daily rate), not very noisy (which is a restriction on noise level), not too far from the city center (which is a restriction on the distance), etc.

By using the above procedure, we can find, for each of the related quantities, the degree to which the given value of this quantity satisfies the corresponding constraint. But what we are interested in is the degree to which the hotel as a whole satisfies all these properties, i.e., the degree to which the hotel is not very expensive *and* not very noisy *and* not too far from the city center, etc. How can we find this degree?

Theoretically, we could use the same procedure as above and ask the expert's opinion about all possible combinations of the corresponding quantities. However, the number of such combinations grows exponentially with the number of quantities, and even for reasonable number of quantities, becomes astronomically large. It is therefore not practically possible to ask for expert's opinion about all these combinations.

Thus, we need to be able to estimate the degree to which an "and"-combination $A \,\&\, B$ is satisfied if we know to what extent A and B are satisfied. In other words, we need to be able, given the degree a to which A is satisfied and the degree b to which b is satisfied, to come up with an estimate for the degree to which the combination $A \,\&\, B$ is satisfied. This estimate is usually denoted by $f_\&(a,b)$, and the corresponding function $f_\&$ is known as an *"and"-operation*, or, for historical reasons, a *t-norm.*

This operation must satisfy several reasonable conditions. For example, since $A \,\&\, B$ means the same as $B \,\&\, A$, it is reasonable to require that the estimates for these two combinations are the same, i.e., that $f_\&(a,b) = f_\&(b,a)$. In mathematical terms, this means that the "and"-operations be commutative.

Similarly, since $A \,\&\, (B \,\&\, C)$ means the same as $(A \,\&\, B) \,\&\, C$, we expect that the corresponding estimates are equal, i.e., that $f_\&(a, f_\&(b, c)) = f_\&(f_\&(a, b), c)$. In other words, the "and"-operations must be associative.

The degree to which we believe that $A \,\&\, B$ holds cannot exceed the degree to which A or B holds, so we must have $f_\&(a, b) \le a$ and $f_\&(a, b) \le b$.

If our degree of belief in A and/or in B increases, then the degree of belief in $A \,\&\, B$ will either increase or remain the same. So, the "and"-operation should be monotonic: if $a \le a'$ and $b \le b'$, then $f_\&(a, b) \le f_\&(a', b')$.

It is also reasonable, since $A \,\&\, A$ means the same as A, to require that $f_\&(a, a) = a$ for all a. It turns out that the only "and"-operation that satisfies all these properties is the minimum $f_\&(a, b) = \min(a, b)$.

Proposition 1. *Let $f_\&(a, b)$ be a function from $[0, 1] \times [0, 1]$ to $[0, 1]$ that satisfies the following properties:*

- *it is monotonic, i.e., $a \le a'$ and $b \le b'$ imply $f_\&(a, b) \le f_\&(a', b')$;*
- *it is idempotent, i.e., $f_\&(a, a) = a$ for all a, and*
- *it satisfies the inequalities $f_\&(a, b) \le a$ and $f_\&(a, b) \le b$ for all a and b.*

Then, $f_\&(a, b) = \min(a, b)$ for all a and b.

Comments.

- All the proofs are places in a special (last) proofs section.
- Please note that this result does not require that $f_\&$ be commutative and/or associative.
- Minimum is indeed one of the most widely use "and"-operations.

Fuzzy constraints. After applying the appropriate "and"-operation to constraints describing individual quantities, we get a membership function (fuzzy set) $\mu(x)$ that describes, for each alternative x, to what extend this alternative satisfies all the given constraints, i.e., to what extent this alternative x is possible.

Alpha-cuts: an alternative way of describing fuzzy sets. Instead of a membership function $\mu(x)$, we can describe the same imprecise information if we describe, for each $\alpha \in (0, 1]$, the set $S_\alpha \stackrel{\text{def}}{=} \{x : \mu(x) \ge \alpha\}$ of all the alternatives for which the expert's degree of confidence that x is possible is at least α. Such sets are known as *alpha-cuts.*

Once we know all the α-cuts, we can uniquely reconstruct the membership function: namely, one can easily prove that for every x, $\mu(x)$ is the largest α for which $x \in S_\alpha$.

Alpha-cuts are usually closed and compact. Since the membership functions are continuous, alpha-cuts are closed sets; since the set of possible alternatives is bounded, each α-cut is a compact set.

Additional natural property. Similar to our arguments that $f(x)$ and $\mu(x)$ be continuous, it is also reasonable to assume that the set S_α to also continuously depend on α – e.g., in terms of the usual Hausdorff metric

$$d_H(S,S')=\max\left(\max_{s\in S}d(s,S'),\max_{s'\in S'}d(s',S)\right),$$

where the distance $d(s',S)$ between an element s' and a set S is defined in the usual way $d(s',S)\stackrel{\text{def}}{=}\min\limits_{s\in S}d(s',s)$. This is true for many known membership functions.

Optimization under fuzzy constraints: Bellman-Zadeh formulation of the problem. Intuitively, it makes sense to say that the desired alternative should be optimal *and* satisfy all the constraints. For example, when we look for a hotel which is the cheapest among all the hotel which are not too far away from the city center, what we are really meaning is that we are looking for a hotel which is cheap and not too far away from the city center.

To describe this idea in precise terms, we need to be able to describe, for each alternative x, the degree $\mu_{\text{opt}}(x)$ to what extent this alternative is optimal. The corresponding degree depends on the value of the objective function $f(x)$, i.e., we must have $\mu_{\text{opt}}(x)=F(f(x))$ for some function $F(x)$.

When the value $f(x)$ is the smallest possible, i.e., when $f(x)=m\stackrel{\text{def}}{=}\min\limits_{x\in X}f(x)$, where X is the set of all alternatives, then this degree is 0: $\mu_{\text{opt}}(x)=0$. In other words, we must have $F(m)=0$.

Similarly, when the value $f(x)$ is the largest possible, i.e., when $f(x)=M\stackrel{\text{def}}{=}\max\limits_{x\in X}f(x)$, then this degree is 1: $\mu_{\text{opt}}(x)=1$. In other words, we must have $F(M)=1$.

It is reasonable to use the simplest linear interpolation to find the values $F(f(x))$ for $f(x)\in(m,M)$. Thus, we get $\mu_{\text{opt}}(x)=\dfrac{f(x)-m}{M-m}$. The degree to which the alternative x is optimal *and* satisfies the constraints can be obtained by applying the corresponding "and"-operation (which we agreed to be min) and is, thus, equal to

$$J(x)\stackrel{\text{def}}{=}\min\left(\frac{f(x)-m}{M-m},\mu(x)\right). \tag{1}$$

We should select the alternative for which this degree of satisfaction is the largest, i.e., we should select the alternative for which $J(x)$ attains the largest possible value.

This formulation was proposed by L. Zadeh and R. Bellman in 1970 and is thus known as Bellman-Zadeh approach to optimization under fuzzy constraints [1]; see also [5, 6].

Comment. Since both the original objective function $f(x)$ and the membership function $\mu(x)$ are continuous, the function $J(x)$ is also continuous. Due to the fact that the set of all possible alternatives S is a compact set, there always exists an alternative at which the new objective function $J(x)$ attains its maximum. Thus, the Bellman-Zadeh formulation always leads to a solution.

Bellman-Zadeh approach: need for new algorithms. At first glance, the situation is good: we have reduced the original practical problem to the problem of unconstrained optimization, and for this problem, as we have mentioned, there are many efficient algorithms.

However, from the computational viewpoint, the situation is not so good: most efficient optimization algorithms require that the objective function be smooth (everywhere differentiable), and the expression $J(x)$ is not differentiable – even when the original objective function $f(x)$ and the membership function $\mu(x)$ describing constraints is differentiable. This is because the function $\min(a, b)$ is not differentiable when $a = b$.

For non-smooth functions, optimization algorithms are not that efficient; thus, it is desirable to design new efficient algorithm for solving the corresponding problem.

A known heuristic algorithm. A heuristic iterative algorithm for optimization under fuzzy constraints is known. This algorithm assumes that we can efficiently solve the corresponding crisp optimization problems. Specifically, we assume that for every α, we can efficiently maximize $f(x)$ under the constraint $x \in S_\alpha$, where S_α is the α-cut of the fuzzy constraint set $\mu(x)$.

In this algorithm, we start with an arbitrary value $\alpha_0 \in (0, 1)$, and then compute the values $\alpha_1, \alpha_2, \ldots$ as follows. Once we have the value α_k, we:

- solve the corresponding constraint optimization problem, i.e., find the maximum M_k of the original objective function $f(x)$ under the constraint $x \in S_{\alpha_k}$ (and find the value x_k at which this maximum is attained); and
- then compute $\alpha_{k+1} \stackrel{\text{def}}{=} \dfrac{M_k - m}{M - m}$.

We stop when the difference between the two consecutive values of α_k becomes sufficiently small, i.e., when $|\alpha_{k+1} - \alpha_k| \le \varepsilon$ for some pre-determined stopping threshold $\varepsilon > 0$. In this case, we return the corresponding alternative x_k as the optimal one.

The main advantage of this heuristic algorithm comes from the fact that for each α, the constraint $x \in S_\alpha$ has a traditional non-fuzzy form. For example, for the inequality constraint of the type $H_j(x) \ge A$ for some fuzzy number A, the α-cut usually has a form $H_j(x) \ge a$ for some crisp a, i.e., the form $h_j(x) \ge 0$, where $h_j(x) \stackrel{\text{def}}{=} H_j(x) - a$. Thus, to find x_k, we can use known and well-developed efficient algorithms for traditional (non-fuzzy) constraint optimization.

Results of testing this algorithm. In [3], one of the authors applied this heuristic algorithm to several different instances of optimization under fuzzy constraints. In all these instances, no matter what the initial value α_0 we selected, the above iterative process converged and let to the solution of the Bellman-Zadeh problem.

Theoretical challenges. These empirical results led to the following theoretical challenges:

1. If the above process converges, do we always get the solution to the Bellman-Zadeh problem (and if yes, why)?
2. Why for each case when the process converged, the resulting limit $\lim_k \alpha_k$ did not depend on the initial value α_0?
3. Does the above process always converge? and
4. If the above iterative process does not always converge, then how can we modify this algorithm to guarantee convergence?

In this paper, we provide answers to all these questions.

2 Analysis of the Problem

Definitions. Let us first describe the above ideas in precise terms.

Definition 1. *Let X be a compact set. By a* reasonable membership function *on X, we mean a continuous function $\mu : X \to [0,1]$ for which the family $S_\alpha \stackrel{\rm def}{=} \{x : \mu(x) \ge \alpha\}$ continuously depends on α for $\alpha > 0$.*

Definition 2.

- *By a problem of* optimization under fuzzy constraints*, we mean the triple $\langle X, f, \mu\rangle$, where $X \subseteq R^N$ is a bounded closed (compact) set, $f : X \to \mathbb{R}$, and μ is a reasonable membership function on the set X.*
- *We say that an element $x_{\rm opt}$ is a* solution *to the problem of optimization under fuzzy constraints if $J(x_{\rm opt}) = \max\limits_{x\in X} J(x)$, where $J(x)$ is defined by the formula (1).*

Simplifying notations. To analyze our problem, it is useful to reformulate it by using simpler notations.

For every α, let us denote $M(\alpha) \stackrel{\rm def}{=} \max\{f(x) : x \in S_\alpha\}$, and $G(\alpha) \stackrel{\rm def}{=} \dfrac{M(\alpha) - m}{M - m}$.

In terms of the function $G(\alpha)$, the existing algorithm takes a very simple form: $\alpha_{k+1} = G(\alpha_k)$.

Properties of the newly defined functions. Since $\alpha < \alpha'$ implies $S_\alpha \supseteq S_{\alpha'}$, we have $M(\alpha) \ge M(\alpha')$ and thus, $G(\alpha) \ge G(\alpha')$. So, the functions $M(\alpha)$ and $G(\alpha)$ are (non-strictly) decreasing functions.

Since S_α continuously depends on α and the function $f(x)$ is continuous, one can show that the function $M(\alpha)$ also continuously depends on α. Thus, the function $G(\alpha) = \dfrac{M(\alpha) - m}{M - m}$ is also continuous.

What happens when the process converges. When the process converges, i.e., when $\alpha_k \to \alpha$, then, due to the continuity of the function $G(\alpha)$, in the limit $k \to \infty$, we get

$$G(\alpha) = \alpha. \tag{2}$$

3 Answer to the First Challenge

The first challenge: reminder. The first challenge was to explain why finding the value α for which the above iterative process converges, i.e., for which $G(\alpha) = \alpha$, is helpful in solving the original problem of optimization under fuzzy constraint.

Proposition 2.

- *If $G(\alpha) = \alpha$, then there exists an optimal solution x_{opt} for which $\mu(x_{\text{opt}}) = J(x_{\text{opt}}) = \alpha$.*
- *Vice versa, for every problem of optimization under fuzzy constraints, there exists an optimal solution x_{opt} for which, for $\alpha \overset{\text{def}}{=} \mu(x_{\text{opt}})$, we have $G(\alpha) = \alpha = J(x_{\text{opt}})$.*

Comments.

- Thus, finding α for which $G(\alpha) = \alpha$ indeed leads to optimal solution – and vice versa.
- When the function $G(\alpha)$ is strictly decreasing, the second part of Proposition 2 becomes even more straightforward.

Proposition 3. *When the function $G(\alpha)$ is strictly decreasing, then for every optimal solution x_{opt}, we have $G(\alpha) = \alpha = J(x_{\text{opt}})$, where $\alpha \overset{\text{def}}{=} \mu(x_{\text{opt}})$.*

4 Answer to the Second Challenge

The second challenge: reminder. The second challenge was to explain why for each case when the heuristic algorithm converged, the resulting limit $\lim\limits_{k} \alpha_k$ did not depend on the initial value α_0.

Our explanation. We know that when the process converges, the limit α satisfies the property $G(\alpha) = \alpha$. According to Proposition 2, this implies that $\alpha = \max\limits_{x \in X} J(x)$. Thus, whenever the heuristic algorithm converges, it converges to the same value $\alpha = \max\limits_{x \in X} J(x)$.

5 Answer to the Third Challenge

The third challenge: reminder. The third challenge was to check whether the above process always converges.

Our answer. We have a simple counter-example. Let $X = [0, 1]$, $f(x) = x$ and let $\mu(x) = 1 - x$. In this case, $m = 0$ and $M = 1$, so $J(x) = \min(f(x), \mu(x)) = \min(x, 1 - x)$. This function increases for $x \le 0.5$ and decreases for $x \ge 0.5$, so its largest possible value is attained for $x = 0.5$ and is equal to 0.5.

Here, for any α, we have $S_\alpha = \{x : 1 - x \geq \alpha\} = [0, 1 - \alpha]$. The largest possible value $M(\alpha)$ of $f(x) = x$ on this α-cut interval is equal to $1 - \alpha$. Since $m = 0$ and $M = 1$, we have $G(\alpha) = M(\alpha) = 1 - \alpha$.

Thus, whatever value $\alpha_0 \leq 0.5$ we start with, we get $\alpha_1 = G(\alpha_0) = 1 - \alpha_0$ and then $\alpha_2 = 1 - \alpha_1 = 1 - (1 - \alpha_0) = \alpha_0$ again. The iterative process oscillates between α_0 and $1 - \alpha_0$ and does not converge.

6 Answer to the Forth Challenge

The fourth challenge: reminder. The fourth challenge was to modify the current heuristic algorithm to guarantee convergence.

Analysis of the problem. According to Proposition 1, we need to find the value α for which $G(\alpha) = \alpha$. This is equivalent to finding the root of the equation $H(\alpha) < 0$, where $H(\alpha) \stackrel{\text{def}}{=} G(\alpha) - \alpha$.

If $G(\alpha_0) = \alpha_0$, we are done. What if $\alpha_1 \neq \alpha_0$?

- If $\alpha_1 = G(\alpha_0) < \alpha_0$, i.e., if $H(\alpha_0) < 0$, then, due to the fact that the function $G(\alpha)$ is non-strictly decreasing, we get $G(\alpha_1) \geq \alpha_1$, i.e., $H(\alpha_1) \geq 0$.
- Similarly, if $\alpha_1 = G(\alpha_0) > \alpha_0$, i.e., if $H(\alpha_0) > 0$, then, due to the fact that the function $G(\alpha)$ is non-strictly decreasing, we get $G(\alpha_1) \leq \alpha_1$, i.e., $H(\alpha_1) \leq 0$.

In both cases, we have two values $\underline{\alpha} < \overline{\alpha}$ for which $H(\underline{\alpha}) \geq 0 \geq H(\overline{\alpha})$. To always find the root α for which $H(\alpha) = 0$, we can use, e.g., bisection; see, e.g., [9]. Thus, we arrive at the following algorithm.

A modified algorithm that always converges. Start with an arbitrary value α_0 and compute $\alpha_1 = G(\alpha_0)$.

If $\alpha_1 = \alpha_0$, we are done, otherwise we form an interval $[\underline{\alpha}, \overline{\alpha}]$ for which $H(\underline{\alpha}) \geq 0 \geq H(\overline{\alpha})$: $\underline{\alpha} = \min(\alpha_0, \alpha_1)$ and $\overline{\alpha} = \max(\alpha_0, \alpha_1)$.

On each iteration, we take $m \stackrel{\text{def}}{=} \dfrac{1}{2} \cdot (\underline{\alpha} + \overline{\alpha})$ and compute $H(m) = G(m) - m$. Then:

- if $H(m) \geq 0$, we replace $\underline{\alpha}$ with m;
- otherwise, we replace $\overline{\alpha}$ with m.

In both cases, the size of the intervals halves. We stop when the difference $\overline{\alpha} - \underline{\alpha}$ becomes smaller than or equal to a pre-defined threshold ε.

7 Proofs

Proof of Proposition 1. Let us consider two possible cases: $a \leq b$ and $b \leq a$.

In the first case $a \leq b$, due to monotonicity, we have $f_\&(a, a) \leq f_\&(a, b)$. Since $f_\&(a, a) = a$, we conclude that $a \leq f_\&(a, b)$. On the other hand, we know that $f_\&(a, b) \leq a$. Thus, we conclude that $f_\&(a, b) = a$, i.e., that in this case, indeed $f_\&(a, b) = \min(a, b)$.

In the second case $b \le a$, due to monotonicity, we have $f_\&(b,b) \le f_\&(a,b)$. Since $f_\&(b,b) = b$, we conclude that $b \le f_\&(a,b)$. On the other hand, we know that $f_\&(a,b) \le b$. Thus, we conclude that $f_\&(a,b) = b$, i.e., that in this case, indeed $f_\&(a,b) = \min(a,b)$.

The proposition is proven.

Proof of Proposition 2.

1°. Let us first prove that if $G(\alpha) = \alpha$, then there exists an alternative $x_{\rm opt}$ for which $J(x_{\rm opt}) = \max\limits_{x \in X} J(x) = \alpha$.

Indeed, let us take x' for which $f(x') = \max\limits_{x \in S_\alpha} f(x)$. Then us show that this is indeed the optimizing value. Indeed, for this x', we have $f(x') = M(\alpha)$ and thus, $\dfrac{f(x') - m}{M - m} = \dfrac{M(\alpha) - m}{M - m} = G(\alpha) = \alpha$. Since $x' \in S_\alpha$, we have $\mu(x') \ge \alpha$ thus $\mu(x') \ge G(\alpha)$. Thus,

$$J(x') = \min\left(\frac{f(x') - m}{M - m}, \mu(x')\right) = \min(G(\alpha), \mu(x')) = G(\alpha) = \alpha.$$

Let us prove that this is indeed the optimal solution, i.e., that $J(x) \le J(x')$ for all $x \in X$. For this, let us consider two possible cases: $x \in S_\alpha$ and $x \notin S_\alpha$.

1.1°. If $x \in S_\alpha$, then, since the maximum of $f(x)$ on S_α is attained at the alternative x', we have $f(x) \le f(x')$. Hence, $\dfrac{f(x) - m}{M - m} \le \dfrac{f(x') - m}{M - m} = \alpha$. Thus,

$$J(x) = \min\left(\frac{f(x) - m}{M - m}, \mu(x)\right) \le \frac{f(x) - m}{M - m} \le \alpha = J(x').$$

So, in this case, indeed $J(x) \le J(x')$.

1.2°. If $x \notin S_\alpha$, then $\mu(x) < \alpha$. Thus,

$$J(x) = \min\left(\frac{f(x) - m}{M - m}, \mu(x)\right) \le \mu(x) < \alpha.$$

Since $J(x') = \alpha$, this also implies that $J(x) < J(x')$.

The first statement is thus proven.

2°. Let us now prove that there always exists an optimal solution $x_{\rm opt}$ for which, for $\alpha = \mu(x_{\rm opt})$, we have $G(\alpha) = \alpha = J(x_{\rm opt})$.

Since the new objective function $J(x)$ is continuous, and the set X is compact, there exists an alternative x' at which this objective function attains its maximum: $J(x') = \max\limits_{x \in X} J(x)$. Let us denote $\alpha' \stackrel{\rm def}{=} \mu(x')$.

2.1°. Let us first prove that $G(\alpha') \le \alpha'$. To prove this, let us assume that $G(\alpha') > \alpha'$ and let us derive a contradiction that will show that this inequality is impossible.

Let x'' be the value $x'' \in S_{\alpha'}$ at which the function $f(x)$ attains its maximum $M(\alpha')$ on the set $S_{\alpha'}$. Then, since $x' \in S(\alpha')$, we have $f(x'') \geq f(x')$ and thus,

$$\frac{f(x'')-m}{M-m} = \frac{M(\alpha')-m}{M-m} \geq \frac{f(x')-m}{M-m}.$$

Since $x'' \in S_{\alpha'}$, by definition of the α-cut, we have $\mu(x'') \geq \alpha'$. Thus,

$$\min\left(\frac{f(x'')-m}{M-m}, \mu(x'')\right) = \min(G(\alpha'), \mu(x')) \geq \min(G(\alpha'), \alpha') \geq$$

$$\min\left(\frac{f(x')-m}{M-m}, \alpha'\right) = J(x').$$

Since the value $J(x')$ is the largest possible value of the auxiliary , we cannot have strict inequality, so $\min(G(\alpha'), \alpha') = J(x')$. Since $G(\alpha') > \alpha'$, this means that $J(x') = \alpha'$, i.e., α' is the largest possible value of the function $J(x)$.

From $G(\alpha') > \alpha'$, it follows that $G(\alpha') > \alpha' + \delta$, where we denoted $\delta \stackrel{\text{def}}{=} \frac{1}{2} \cdot (G(\alpha') - \alpha')$. Since we assumed that the function $G(\alpha)$ is continuous, there exists a $\varepsilon > 0$ for which $|\alpha - \alpha'| \leq \varepsilon$ also implies $G(\alpha) > \alpha' + \delta$. Thus, for $\alpha'' \stackrel{\text{def}}{=} \alpha' + \min(\delta, \varepsilon)$, we have $\alpha'' > \alpha'$ and $G(\alpha'') > \alpha'$ hence $\min(G(\alpha''), \alpha'') > \alpha' = J(x')$. Let us now take the value x''' for which the maximum of $f(x)$ is attained on the set $S_{\alpha''}$, i.e., for which $f(x''') = M(\alpha'')$. Then, $\frac{f(x''')-m}{M-m} = \frac{M(\alpha'')-m}{M-m} = G(\alpha'')$ and – due to $x''' \in S(\alpha'')$ – we have $\mu(x''') \geq \alpha''$. Then, we have

$$\min\left(\frac{f(x''')-m}{M-m}, \mu(x''')\right) \geq \min(G(\alpha''), \alpha'') > \alpha' = J(x'),$$

which contradicts to our assumption that $J(x')$ is the largest possible value of the function $J(x)$.

Thus, the case $G(\alpha') > \alpha'$ is indeed impossible, and so $G(\alpha') \leq \alpha'$.

2.2°. Since $G(\alpha') \leq \alpha'$, we have two possible cases: $G(\alpha') < \alpha'$ and $G(\alpha') = \alpha'$. Let us show that in the first case, we can find some value α'' for which $G(\alpha'') = \alpha''$.

Indeed, let us assume that $G(\alpha') < \alpha'$. In this case, let us take x'' for which $f(x'') = \max\limits_{x \in S_{\alpha'}} f(x)$. Then, $f(x'') = M(\alpha') \geq f(x')$ and $\mu(x'') \geq \alpha'$, hence

$$J(x'') = \min\left(\frac{f(x'')-m}{M-m}, \mu(x'')\right) \geq \min(G(\alpha'), \mu(x'')) \geq$$

$$\min(G(\alpha'), \alpha') \geq \min\left(\frac{f(x')-m}{M-m}, \alpha'\right) = J(x').$$

Since $J(x')$ is the largest possible value of x', we thus get $J(x'') = J(x') = \min(G(\alpha'), \alpha')$. Since $G(\alpha') < \alpha'$, this means that $J(x') = G(\alpha')$.

Let us now take $\alpha'' \stackrel{\text{def}}{=} G(\alpha')$. Then, $\alpha'' < \alpha'$. Since the function $G(\alpha)$ is non-strictly decreasing, we have $G(\alpha'') \geq \alpha''$. Thus, $\min(G(\alpha''), \alpha'') \geq \alpha'' = G(\alpha') = J(x')$. By taking the alternative $x'' \in S_{\alpha''}$ that maximizes $f(x)$, we get $J(x'') \geq J(x')$, and since $J(x')$ is the largest value, we get $J(x'') = J(x')$.

Similarly to Part 2.1 of this proof, we can prove that we cannot have $G(\alpha'') > \alpha''$, so we have $G(\alpha'') = \alpha''$.

2.3°. Once we have a value α for which $G(\alpha) = \alpha$, we can use Part 1 of this proof to show that $\max\limits_{x \in X} J(x) = \alpha$. The proposition is proven.

Proof of Proposition 3. Let us assume that the function $G(\alpha)$ is strictly decreasing, that $J(x_{\text{opt}}) = \max\limits_{x \in X} J(x)$. Let us prove that for $\alpha \stackrel{\text{def}}{=} \mu(x_{\text{opt}})$, we have $G(\alpha) = \alpha = J(x_{\text{opt}})$.

Indeed, let x' be the alternative for which the original objective function $f(x)$ attains its maximum on the set S_α. Since $x' \in S_\alpha$, we thus get $f(x') \geq f(x_{\text{opt}})$, hence $\dfrac{f(x') - m}{M - m} = \dfrac{M(\alpha) - m}{M - m} = G(\alpha) \geq \dfrac{f(x_{\text{opt}}) - m}{M - m}$; also, $\mu(x') \geq \alpha_{\text{opt}} = \mu(x')$. Thus,

$$J(x') = \min\left(\frac{f(x') - m}{M - m}, \mu(x')\right) \geq \min(G(\alpha), \alpha) \geq J(x_{\text{opt}}).$$

Since x_{opt} is the optimal alternative, we have $J(x_{\text{opt}}) = \min(G(\alpha), \alpha)$.

Similar to the proof of Part 2.1 of Proposition 2, we can conclude that $G(\alpha) \leq \alpha$. So, to complete our proof, we need to prove that $G(\alpha) = \alpha$, i.e., that the case $G(\alpha) < \alpha$ is not possible.

Indeed, in this case, we can take $\alpha' = \dfrac{1}{2} \cdot (G(\alpha) + \alpha)$. Here, $\alpha' > G(\alpha)$. Due to strict monotonicity, we have $G(\alpha') > G(\alpha)$, thus, $\min(G(\alpha'), \alpha') > G(\alpha) = J(x')$. By taking an alternative x'' that maximizes $f(x)$ on $S_{\alpha'}$, we will get $J(x'') \geq \min(G(\alpha'), \alpha') > G(\alpha) = J(x')$ and thus, $J(x'') > J(x')$, which contradicts to our assumption that $J(x')$ is the largest possible value of $J(x)$.

The proposition is proven.

References

1. Bellman, R.E., Zadeh, L.A.: Decision making in a fuzzy environment. Management Science, **17**(4), B141-B164 (1970)
2. Belohlavek, R., Dauben, J.W., Klir, G.J.: Fuzzy Logic and Mathematics: A Historical Perspective. Oxford University Press, New York (2017)
3. Figueroa-García, J.C., Tenjo-García, J.S., Bustos-Tellez, C.A.: Solving transshipment problems with fuzzy delivery costs and fuzzy constraints, In: Proceedings of the 2018 Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2018, Fortaleza, Brazil, July 4–6, 2018.
4. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice Hall, Upper Saddle River, New Jersey (1995)

5. Kosheleva, O., Kreinovich, V.: Why Bellman-Zadeh Approach to Fuzzy Optimization, Applied Mathematical Sciences, **12**(11), 517–522 (2018)
6. Kreinovich, V., Kosheleva, O., Shahbazova, S.: Which t-Norm Is Most Appropriate for Bellman-Zadeh Optimization, In: Proceedings of the World Conference on Soft Computing, Baku, Azerbaijan, May 29–31, 2018.
7. Mendel, J.M.: Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions. Springer, Cham, Switzerland (2017)
8. Nguyen, H.T., Walker, E.A.: A First Course in Fuzzy Logic. Chapman and Hall/CRC, Boca Raton, Florida (2006)
9. Nocedal, J., Wright, S.: Numerical Optimization, 2nd Edition, Springer, New York (2006)
10. Novák, V., Perfilieva, I., Močkoř, J.: Mathematical Principles of Fuzzy Logic. Kluwer, Boston, Dordrecht (1999)
11. Zadeh, L.A.: Fuzzy sets, Information and Control, **8**, 338–353 (1965)