A Discrete Search and Rescue Problem Under Uncertain Interval Parameters*

Joël Bougron[ab], Julien Alexandre dit Sandretto[ac], Bruno Ricaud[de], Stéphane Cardon[fg], and Aline Hufschmitt[fh]

Abstract

The Search and Rescue Problem (SARP) can be formulated in an environment subject to both objective uncertainty (randomness inherent to nature) and subjective uncertainty (lack of knowledge about the state of the world). In this paper, we present an interval arithmetic interpretation of the uncertainty problem. A crisis scenario is modeled as an assignment and optimization problem with interval-valued parameters and constraints. These intervals capture uncertainty over the problem data. A branch and bound algorithm is used to explore the solution space. Interval arithmetic is employed to compute bounds and obtain feasible assignments. From the resulting assignments, residual injury intervals are derived to assess the impact of uncertainty on each wounded person. Parallel computing techniques are also investigated to reduce execution times in the solution process.

Keywords: intervals, Search And Rescue, WTA, uncertainty

1 Introduction

In crisis-response and humanitarian assistance scenarios, efficiently allocating limited medical resources is critical. Coordinating caregivers with diverse skills to meet heterogeneous needs under urgent conditions requires structured decision-making models. We are interested in this type of problem because it appears in various

*This work was supported by KNDS France, a Franco-German company of the defense industry. This project also received funding from the French National Association for Research and Technology (ANRT).

[a]ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France
[b]E-mail: joel.bougron@ensta.fr, ORCID: 0009-0004-2062-2832
[c]E-mail: julien.alexandre-dit-sandretto@ensta.fr, ORCID: 0000-0002-6185-2480
[d]Innovation and International Directorate (D2I), KNDS France, Versailles, France
[e]E-mail: bruno.ricaud@knds.fr, ORCID: 0009-0009-6360-0808
[f]Research Center (CReC), Académie Militaire de Saint-Cyr Coëtquidan, Guer, France
[g]E-mail: stephane.cardon@st-cyr.terre-net.defense.gouv.fr, ORCID: 0000-0002-2475-1365
[h]ORCID: 0000-0003-1649-3180

DOI: 10.14232/actacyb.323748

applications such as allocating technicians to jobs, vehicles to deliveries, servers to requests, weapons to targets, etc.

1.1 Assignment under uncertainties

In this paper, we consider the **Search and Rescue Problem** (SARP), which includes several subjects of study such as information perception and fusion, path planning, and more particularly, what interests us here, the assignment of caregivers to wounded individuals.

The problem of **assigning resources to tasks** has been widely studied in the literature and concerns many domains where resources must be optimized, such as industry, logistics, computing, military domain, etc. However, in all these domains, resources are assigned to tasks under various **constraints** (e.g., a resource cannot be assigned to two tasks simultaneously).

Under these constraints, it is necessary to calculate ideally the optimal assignments according to several parameters. However, in emergency situations, some parameters are not precisely known and are therefore uncertain. To model this uncertainty, we propose an interval-based approach [22], rather than a probabilistic approach. The goal is then to solve this problem of assigning caregivers to wounded individuals using a constraint optimization algorithm with interval parameters.

In Section 1.2, we formally specify the problem. We show in Section 2 why it remains an open problem in the existing literature. To address it, we have designed a Branch and Bound type algorithm that directly manipulates intervals, which we explain in Section 3. Then, we apply and analyze it in Section 4. The conclusion of this paper is given in Section 5.

1.2 A caregivers-to-wounded assignment problem with interval parameters

We consider a crisis scenario involving m **caregivers**, organized into t **teams**, whose objective is to minimize the injuries of n **wounded individuals**. An example is given in Figure 1. Each wounded person has a specific injury level, which influences the likelihood that a caregiver will engage with them. Caregivers possess **different skills**, while the wounded present **diverse needs**. The degree of adequacy between a caregiver's skills/equipment and a wounded person's injury type also affects the likelihood of engagement, and an assignment is only possible if at least one caregiver has characteristics that match those types. The crisis scenario can be modeled as an assignment problem which has specific characteristics. First, each caregiver can be assigned to at most one wounded person. In other words, the assignment function is **injective** from the wounded to the caregivers. Second, each wounded person can receive assistance from zero, one, or multiple caregivers — the assignment is thus **many-to-one**. It is relevant for combining the effects of the caregivers. Third, the numbers of caregivers and wounded individuals are arbitrary, making the problem **unbalanced**. Moreover, caregivers are required to remain within predefined teams in order to stay organized in this critical context.

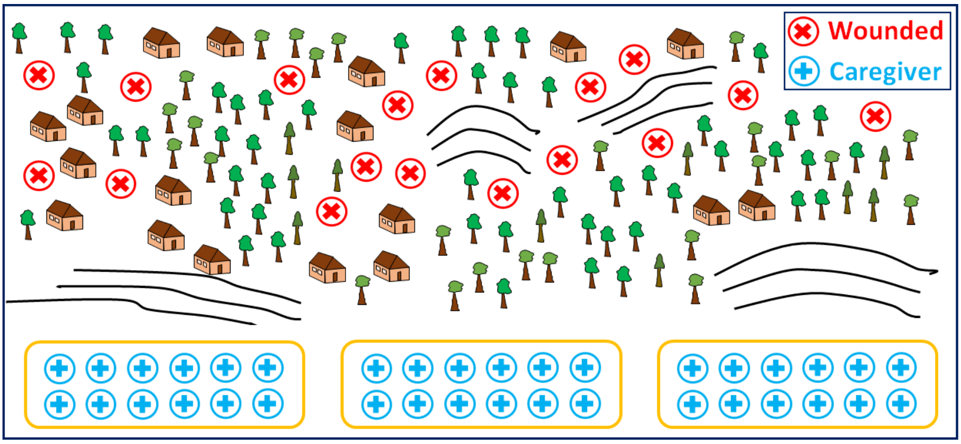

Figure 1: A crisis scenario

Each assignment — or non-assignment — is represented by a **binary decision variable** $x_{ij} \in \{0, 1\}$. Caregiver i is assigned to wounded person j if $x_{ij} = 1$; otherwise, $x_{ij} = 0$. The solution space is therefore the Cartesian product of $m \times n$ binary sets $\{0, 1\}$.

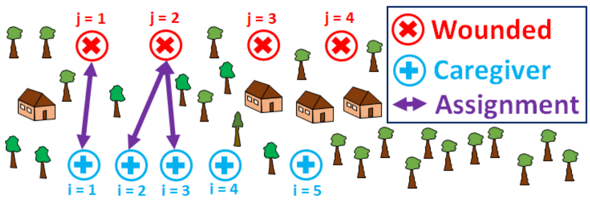

Figure 2: Illustration of the Assignment Problem

For illustration, Figure 2 shows an example with three assignments corresponding to $x_{11} = 1$, $x_{22} = 1$, and $x_{32} = 1$. The non-assignments correspond to $x_{ij} = 0$.

Two parameters model the problem, as input data, depending on the circumstances, derived from professional knowledge and habits, provided by experts or within the command and control (C2) framework. On the one hand, for a wounded person j, v_j denotes their initial injury level, i.e. the **severity level** of their health condition (e.g., severity of a snake bite, severity of bleeding). This value, intrinsic to the wounded person, is what we aim to minimize. On the other hand, for each caregiver-wounded pair (i, j), p_{ij} represents the probability of successful treatment, i.e. the adequacy between the skills and equipment of caregiver i and the **type of injury** of wounded person j (e.g., snake bite, bleeding wound), independently of their severity level. It indicates the reduction ratio of v_j if caregiver i treats wounded person j. Consequently, $1 - p_{ij}$ is the **probability of failed treatment**, which we aim to minimize.

These variables are considered uncertain, then we represent them using intervals [22]. The initial injury level is modeled as an interval $[v_j] = [\underline{v_j}, \overline{v_j}] \in \mathbb{IR}$, and the probability of failed treatment is also represented as an interval $1-[p_{ij}] \in \mathbb{IR}$, which we seek to keep as low as possible, by minimizing its upper bound.

We observe that if $x_{ij} = 0$, then $(1 - p_{ij}) \cdot x_{ij} = 0$, so the injury is not reduced. But if $x_{ij} = 1$, then $(1 - p_{ij}) \cdot x_{ij} = 1 - p_{ij}$, and the injury is reduced by $(1 - p_{ij})$ ratio. In the entire scenario, if these ratios are assumed to be independent, then they are **cumulative**. The product over the m caregivers assigned to wounded person j gives the uncertain cumulative probability interval that all caregivers fail to treat wounded j, which we aim to minimize. The product of this cumulative probability of failed treatment with the initial injury v_j yields the residual injury of wounded j. More precisely, the cumulative probability of failed treatment for wounded j is given by

$$\prod_{i=1}^{m}(1 - [p_{ij}])^{x_{ij}} \in \mathbb{IR} : \text{cumulative probability of failed treatment} \tag{1}$$

and, consequently, the residual injury of wounded j is expressed as

$$[v_j]\prod_{i=1}^{m}(1 - [p_{ij}])^{x_{ij}} \in \mathbb{IR} : \textbf{residual injury} \text{ of wounded } j. \tag{2}$$

The objective is to minimize the overall injuries of the wounded, represented as an interval quantity equal to the sum of their residual injuries. SARP is modeled as the minimization of the total injury interval $[F] = [\underline{F}, \overline{F}] \in \mathbb{IR}$. The optimization problem seeks the **binary assignment matrix** (x^*_{ij}) minimizing:

$$(x^*_{ij})_{1\le i\le m,\ 1\le j\le n} \in \arg \min_{x_{ij}\in\{0,1\}} \sum_{j=1}^{n}[v_j] \cdot \prod_{i=1}^{m}(1 - [p_{ij}])^{x_{ij}} \tag{3}$$

This nonlinear multiplicative formulation captures the accumulation of treatment probabilities and is equivalent to the classical Weapon Target Assignment Problem (WTAP [21]) which is found in military applications. Indeed, the two approaches share analogies as resource allocation problems to tasks [23]. Thus, the caregiver is considered as a weapon, a wounded person as a target, the initial injury level of a wounded person v_j corresponds to the initial threat level of a target, the probability of successful treatment of a wounded person corresponds to the probability of neutralizing a target, and finally, minimizing the total injury of all wounded individuals corresponds to minimizing the total threat of the targets. For this reason, SARP is viewed as an instance of WTAP in the remainder of the paper.

In the WTA problem, the two parameters v_j and p_{ij} are independent since the former is intrinsic to j, while the latter represents the adequacy between i and j. Indeed, in an assignment phase, even if several caregivers are assigned to the same wounded person, the effects of these assignments are modeled as simultaneous (we do not consider the movements). However, it can be noted that the problem is iterative, each iteration being independent of the preceding one. But, for each new assignment phase, the parameters v_j and p_{ij} are updated according to the effects of the previous iteration.

Intervals are used to encode input uncertainties, with v_j and p_{ij}. Interval arithmetic enables consistent propagation of these uncertainties to the output objective $[F]$. Interval arithmetic is also employed to check constraints. To complete our optimization problem, we need to consider five constraints.

First, the **binary constraint** (C.1), which reflects the domain of the decision variables. Second, the **at-most-one-wounded-per-caregiver constraint** (C.2), which follows from the definition of the assignment problem (this constraint is enforced by design in the algorithm). For example, in Figure 2, caregiver 1 cannot engage with another wounded at the same time as wounded 1. Third, the **minimum care constraint** (C.3): if caregiver i is assigned to wounded person j, then the expected care $[v_j] \cdot p_{ij}$ must be greater than or equal to a minimum threshold of minimum care $w_j^{\min}$ (gathered in the vector $W^{\min}$). This is an active filtering constraint to avoid negligible treatment. For example, if $\overline{v_3} = 10.0$, $\overline{p_{23}} = 0.4$, and $w_2^{\min} = 5$, then caregiver 2 will not engage with wounded 3. Fourth, the **team exclusivity constraint** (C.4): if caregivers from team k are assigned, then no caregiver from any other team $\ell \neq k$ may be assigned (this is also an active filtering constraint). The caregivers' teams (of equal size) are stored in the vector T, where $T[i] = t_k$ means that caregiver i belongs to team t_k. Thus, three teams are shown in Figure 1. For example, if wounded person 09 is treated by team 1 (e.g., caregivers 01 and 02), it can't be treated by any other team (this example corresponds to the results of Section 4). Fifth, the **maximum residual injury constraint** (C.5): the total sum must be less than or equal to the maximum threshold of residual injuries $w_j^{\max}$ (gathered in the vector $W^{\max}$).

$$\forall (i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}, \quad x_{ij} \in \{0,1\} \tag{C.1}$$

$$\forall i \in \{1,\dots,m\}, \quad \sum_{j=1}^{n} x_{ij} \leq 1 \tag{C.2}$$

$$\forall (i,j), \quad x_{ij} = 1 \Rightarrow [v_j] \cdot [p_{ij}] \subset [w_j^{\min}, +\infty] \tag{C.3}$$

$$\forall j, \forall k, \quad \left(\sum_{i \in t_k} x_{ij} \right) \cdot \left(\sum_{i \in t_l} x_{ij} \right) = 0 \tag{C.4}$$

$$\forall j, \quad [v_j] \prod_{i=1}^{m} (1 - [p_{ij}])^{x_{ij}} \subset [0.0, w_j^{\max}] \tag{C.5}$$

Since, as described above, the objective of this minimization problem is interval-valued, we choose to follow the Hurwicz optimism–pessimism criterion [19] to scalarize it using the upper and lower bounds of each interval, denoted respectively by $\mathrm{ub}(\cdot)$ and $\mathrm{lb}(\cdot)$, as follows. The function $F_\alpha(X)$ is defined by:

$$\alpha \cdot \mathrm{ub}\left(\sum_{j=1}^{n}[v_j] \cdot \prod_{i=1}^{m}(1-[p_{ij}])^{x_{ij}}\right) \;+\; (1-\alpha)\cdot \mathrm{lb}\left(\sum_{j=1}^{n}[v_j] \cdot \prod_{i=1}^{m}(1-[p_{ij}])^{x_{ij}}\right) \quad (4)$$

where the parameter $\alpha \in [0,1]$ expresses the relative weight given to the optimistic (lower bound) versus pessimistic (upper bound) evaluation of the interval objective of the minimization problem. When $\alpha = 1$, the decision is fully robust and based only on the worst case; when $\alpha = 0$, it is fully optimistic and relies only on the best case; intermediate values yield a compromise between these two extremes. In this work, we adopt the robust case $\alpha = 1$ (pessimistic approach), so that:

$$F(X) \;=\; \mathrm{ub}\left(\sum_{j=1}^{n}[v_j] \cdot \prod_{i=1}^{m}(1-[p_{ij}])^{x_{ij}}\right) \quad (5)$$

This implies that, as illustrated in Figure 3, our approach selects the green intervals, which are considered greater than the red ones in all three configurations, regardless of their widths.

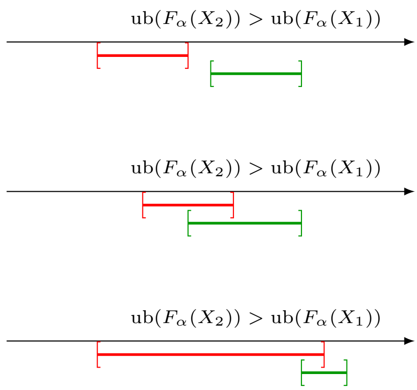

Figure 3: Interval comparisons for red $F_\alpha(X_1)$ and green $F_\alpha(X_2)$

Instead of minimizing the worst-case, it is possible to minimize the best-case (i.e., selecting the red intervals). This is the choice of optimism, which can be seen as taking a risk. If this optimism is indeed aligned with reality, it would allow us to use fewer caregivers for the same set of assignments, with these resources being redeployed elsewhere.

2 Survey and contribution

The context of epistemic uncertainty makes classical optimization methods on real numbers inoperative and requires approaches adapted to the manipulation of intervals, their propagation, and their consideration in the evaluation of solution quality. In this section, we present several approaches identified in the literature that could fit our problem.

2.1 Prior publications

Our problem is a generalization of an unbalanced assignment problem, with several caregivers assignable to one wounded (many-to-one), structured in teams, with a nonlinear objective function and parameter coefficients represented by intervals.

We designate our problem as *Interval Parameters Unbalanced Many-to-one by Teams NonLinear Assignment Problem* or **Interval Weapon Target Assignment Problem** (IWTAP) because of its proximity with WTA problem designation.

No publication considering intervals in the WTA problem has been identified. The state of the art is therefore oriented on the most closely related problems in terms of characteristics, such as the Assignment Problem (AP), which is one-to-one and balanced.

It is then a matter of characterizing and analyzing existing contributions according to several axes: the structure of the objective function, the nature of the decision variables, the nature of the parameters, the resolution methods, and above all the way intervals are manipulated in the resolution. We examine below the approaches that, on the one hand, use interval parameters, and on the other hand have points in common with IWTAP according to the definition. Thus, we first present continuous or binary-continuous problems. Then, we discuss discrete problems reduced to continuous ones. Next, we focus on discrete problems where intervals are not directly considered but via representative real numbers. Then we discuss some links with fuzzy logic and with probabilities. Finally, we present application contexts with the same structure.

2.2 Continuous problem with interval parameters

Our problem is discrete. In this section we detail several types of problems involving continuous decision variables and interval parameters, to identify approaches from continuous settings that can be reused or adapted.

2.2.1 Expression of a single-objective interval problem as a bi-objective problem on a bound and the center

As initiated and formalized in [15], a continuous linear single-objective problem with interval parameters can be transformed into a bi-objective problem on a bound and the center of the intervals. This is quite far from our problem, which is nonlinear and discrete. However, the idea is reused for linear and discrete problems as discussed in Sections 2.4.2 and 2.4.3.

In this framework, order relations are defined in [15] that allow comparing objectives. This is applied in [16] for a center-radius order relation, with a weighting factor between the center and the radius. But the solving method (genetic algorithm) cannot provide an optimal solution.

2.2.2 Interval analysis to solve continuous optimization problems

In general, interval analysis makes it possible to generate lower and upper bounds for a function. Thus, for the continuous domain, in [13] are described algorithms that use interval analysis to solve nonlinear optimization problems. In particular, interval analysis allows reducing the search space while keeping the box wide enough to contain the solution. For example, the study in [6] finds value intervals of photovoltaic cell models (continuous decision variables) in an optimization problem, solved using a *Depth-First Search with Backtracking* and an interval library (here, `Ibex` [14]).

Such solving methods require continuous decision variables and cannot handle discrete ones, as in our case.

2.2.3 Interval analysis to solve binary-continuous optimization problems

Other sources such as [1] and [29] which address cost minimization for solvent mixing, consider **binary** decision variables y_i (whether the solvent is selected or not) and **continuous** decision variables x_i (the solvent rate in the composition as real decision variables x_i). In this case, interval analysis is used to reduce the search space of the continuous variables x_i, the interval extension function taking as input the x_i and not the y_i.

So, such solving methods still require continuous decision variables and cannot handle discrete ones.

2.3 Assignment problems with interval parameters reduced to continuous problems

In this section, we describe problems with interval parameters reformulated as continuous optimization problems in order to identify approaches suitable for our discrete one.

2.3.1 Binary variables considered as continuous variables

Assignment problems with interval parameters can sometimes be reformulated as continuous optimization problems. For example, [12] addresses a problem of maximizing a linear multi-objective function with interval parameters, transformed into a single-objective one by moving the other objectives into constraints. The decision variables are binary for an unbalanced many-to-one location assignment problem. Interval parameters are reconsidered as exact via a penalty function. An **equivalent probability matrix** allows, instead of binary decision variables, to consider the probability in $[0, 1]$ that the binary decision variable equals 1. Decision variables thus become continuous (the rest being solved by a particle swarm optimization metaheuristic).

But these approaches are not adapted to the binary nature of the decision, nor to the nonlinear form of the Equation (3).

2.3.2 Substitution of binary variables into continuous parameter variables

In our problem, interval parameters make the resolution more complex because of their presence in the body of the objective function. One can be led to make a **substitution of variables** by moving the binary variables of the objective function to the constraints. At the same time, the problem is reformulated so that the new decision variable is the upper bound of the interval parameter (and not the lower bound which is in the constraints) and therefore is real. [28] presents such an approach for a linear, balanced, one-to-one multi-objective problem (reformulated and solved as a min-max problem).

However, this approach requires the objective function to be linear, unlike in our case.

2.4 Assignment problems with interval parameters reduced to assignment problems with real parameters

Problems with interval parameters are often simplified by replacing intervals with real values. In this section, we present some examples. No known solvers directly handle assignment problems combining binary variables and interval uncertainty. That is why interval uncertainty is often treated indirectly: the problem is solved by considering real parameters. In a first very direct approach, only the centers of the intervals can be considered. In a second, less direct approach, the single-objective problem is transformed into a multi-objective one: a disjunction can then be made on bounds and/or centers (which can also be combined with several objectives if the function is already multi-objective).

2.4.1 Assignment problem with Hungarian method on interval centers

The simplest of these approaches consists in replacing interval parameters with real values to obtain an equivalent problem.

Thus, a balanced, single-objective, linear assignment problem with interval parameters is treated in [20] via midpoints by applying the Hungarian algorithm on the matrix of interval centers. A numerical application is provided in [26]. This approach is extended to unbalanced cases in [27] via dummy rows/columns. This method, for other order relations, is also reused in [7].

However, these approaches are not compatible with the multiplicative form of the objective function in Equation (3).

2.4.2 Case disjunction methods with a bound and the center

As seen in [15], approaches disjoin the cases according to several interval characteristics (lower bound, upper bound, or center), and according to the components of the objective function. Thus, in [18], for a minimization problem of a linear bi-objective function (unbalanced one-to-one assignment), one ends up with four single-objective problems by disjunction of cases, according to the two objectives, and according to the upper bound and the center. Similarly, these approaches are not compatible with Equation (3).

2.4.3 Case disjunction methods on the bounds

In [2, 3], the same principle is applied, but with disjunction according to each of the lower and upper bounds, then solved by genetic algorithms or solvers.

In [24], under the close name *Grey Assignment Problem* (parameters being intervals), with *Grey Arithmetic* operations, operations on parameter matrices lead on the same principle to two problems (one on lower bounds, the other on upper bounds), each being solved by the Hungarian algorithm.

Similarly, these approaches are not compatible with the form of the objective function.

2.5 Links with fuzzy logic and probabilities

We chose to represent uncertainty with intervals, but other forms of uncertainty modeling also exist. For instance, a Fuzzy Assignment Problem can be turned into an interval one using the $\alpha - cut$ method [7].

More generally, the notion of *probability bounds analysis* (PBA) is introduced in [9], combining probability theory and interval arithmetic. A formalism is proposed in [17] where some parameters are described by intervals, while others are described by probability laws.

2.6 Application Contexts with Similar Structure

Minimizing a cost with such a multiplicative form is not specific to our crisis scenario. Several problem domains exhibit similar structural features, where entities are assigned to minimize uncertain risk or loss, or to maximize a benefit: assigning stimuli to brain regions to maximize human satisfaction [25], allocating budgets to maximize advertising reach [5], assigning treatments to minimize the probability of cancer damage [4], dispatching ambulances to emergency locations [11], and, as discussed, assigning weapons to targets to minimize residual threat.

2.7 Contribution

In this survey, we have shown that existing works focus on continuous problems or on simplified assignment problems, using indirect methods to compute the intervals.

Our conclusion is that Unbalanced Many-to-one by Teams NonLinear Assignment problem assimilable to WTA problem has not been solved using interval parameters.

As defined, WTA is a problem of constrained optimization. For discrete-continuous domains with interval parameters, interval-based Constraint Solving Problems (CSP) are generally solved using Constraint Programming (CP) techniques. For large and complex constraint instances, there are now effective techniques, such as SAT modulo theory (SMT) [10]. This would be relevant for problems with more complex constraints and could be useful for the search for counterexamples. For optimization problems, STM-Opt prototypes are known and are competitive, especially for combinatorial optimization problems. However, these prototypes do not seem to be publicly available yet. Moreover, note that the problem is potentially symmetric because parameters related to caregivers and wounded can coincide, thus defining equivalence classes. Symmetry could then reduce the search space, for example by integrating symmetry-breaking constraints [8]. This could be an interesting and original option for our problem (not developed here).

We therefore contribute with Algorithm 1, which directly computes the intervals (through the products of p_{ij}, the products of v_j by p_{ij}, the sums of s_j, and the comparison of upper bounds) for unbalanced and many-to-one assignments, with a multiplicative form of the objective function (a feature of WTA problem).

We have then implemented this algorithm in `C++` with the `Ibex` library [14]. In the next section, we analyse our approach, and study the variations of parameters (residual injuries and the upper bound of a wounded's initial injury level).

3 A Branch and Bound algorithm for Assignment problem with intervals parameters

To detail this contribution, this section presents the exact optimization Algorithm 1 used to explore all feasible caregiver-to-wounded assignments. The recursive `OPTIMIZE` function performs a depth-first Branch and Bound search. Equivalent to the assignment matrix introduced in Section 1, each component $A[i]$ is the wounded assigned to the caregiver i.

3.1 Branch and Bound Optimization function

The main `SOLVE` function takes as input the parameters of our problem: the number of caregivers m and wounded n, the intervals of the initial injury levels $[v_j] \in \mathbb{IR}$ gathered in the vector $\mathbf{V}$, the intervals of the treatment success probabilities $[p_{ij}] \in \mathbb{IR}$ gathered in the matrix $\mathbf{P}$, the parameters ($W^{\min}$, T, $W^{\max}$) associated with the constraints (C.3)–(C.4)-(C.5), a greedy heuristic first solution A_{gr} and its interval objective $[F_{\mathrm{gr}}]$.

It calls once the recursive `OPTIMIZE` function with the current partial assignment A initialized as $A = [-1, -1, \ldots, -1]$ (meaning that no caregiver is yet assigned). At each recursion, if caregiver i has no valid wounded candidate, $A[i] = -2$. The objective is to minimize the total residual injury interval, as defined in Equation (3) in Section 1.

Before calling main `SOLVE` function, a **greedy heuristic** is first used to build an initial solution A_{gr}. Caregivers $i \in \{1, \dots, m\}$ are sorted according to the maximum amount of care they can potentially give $S_i = \sum_{j \in \mathcal{C}_i} \overline{[v_j]\,[p_{ij}]}$ among the admissible wounded $\mathcal{C}_i = \{\, j \mid x_{ij}$ valid according to (C.3) and (C.4) $\}$. Those with lower S_i (i.e. limited overall care potential) are considered first. For each wounded, the candidate $j^\star \in \mathcal{C}_i$ is then selected as the one minimizing

$$j^\star = \arg\min_{j \in \mathcal{C}_i} \overline{F(A \text{ s.t. } A[i] = j)} \tag{6}$$

and the resulting solution is considered only if it satisfies constraint (C.5). The initial greedy solution will serve as a first best assignment as a minimum comparison baseline (line 4) during the first call to the `OPTIMIZE` function.

At **each recursion**, the `OPTIMIZE` function takes as input the parameters of our problem, the current partial assignment A describing which caregivers have already been assigned, the best solution found so far and its interval objective $[F^*] = [\underline{F^*}, \overline{F^*}]$.

When an assignment is complete, the maximum residual injury constraint (C.5) is satisfied, and the upper bound $\overline{F}$ of the new solution is smaller than that of the current best solution (line 4), and then the best solution is updated (lines 2–3). The function then returns A^* and $[F^*]$, which ends the current branch of the search tree (line 28).

Otherwise, the **evaluation function** f computes interval **bounds** of the total residual injury that can still be reached from a partial assignment A. For each wounded j, the evaluation first gathers the contribution of the caregivers already assigned to j by multiplying their failed-treatment probabilities. It then extends this partial product by including the possible influence of all unassigned caregivers, which gives

$$[P_j] = [P_j^{\min},\, P_j^{\max}] \times \prod_{i=-1} (1 - [p_{ij}]). \tag{7}$$

This interval represents all failed-treatment probabilities still compatible with the current partial assignment. This provides the residual injury intervals $[v_j] \times [P_j]$. If the test $\underline{[f(A)]} < \overline{[F^*]}$ succeeds, the partial assignment A can yet improve the best solution found so far. Then, the current branch is explored (lines 5-26). If it fails, the function then returns A^* and $[F^*]$, which ends the current branch of the search tree (line 28).

During this search (lines 5–26), at each recursion, the `OPTIMIZE` function selects the next caregiver following this heuristic (line 5). The caregiver, $i^\star$, having the lowest total potential treatment,

$$i^\star = \arg\min_{i=-1} \sum_j \overline{[v_j], [p_{ij}]}, \tag{8}$$

is selected first, and it ensures that caregivers with lower overall effectiveness are

Algorithm 1 An interval B&B algorithm for IWTAP.

Input: $\Sigma = (m, n, V, P, W^{\min}, T, W^{\max})$, A_{gr}, $[F_{\text{gr}}]$
Output: best assignment A^* and best objective value $[F^*]$

Function SOLVE($\Sigma, A_{\text{gr}}, [F_{\text{gr}}]$)

if $A_{\text{gr}} \neq \perp$ **then**
    return OPTIMIZE($\Sigma, [-1, \ldots, -1], A_{\text{gr}}, [F_{\text{gr}}]$)
else
    return OPTIMIZE($\Sigma, [-1, \ldots, -1], [-1, \ldots, -1], [+\infty, +\infty]$)
end if

Function OPTIMIZE($\Sigma, A, A^*, [F^*]$)

if (A is complete **and** (C.5) satisfied **and** $\overline{[F(A)]} < \overline{[F^*]}$ **then**
    $A^* \leftarrow A$
    $[F^*] \leftarrow [F(A)]$
else if $\underline{[f(A)]} < \overline{[F^*]}$ **then**
    Choose a caregiver following Heuristic (8)
    if no such caregiver exists **then**
        return $(A^*, [F^*])$
    end if
    Wounded-Candidates $\leftarrow \emptyset$
    for all wounded j **do**
        if (C.3) and (C.4) are satisfied **then**
            Wounded-Candidates $\leftarrow$ *Wounded-Candidates* $\cup \{j\}$
        end if
    end for
    if *Wounded-Candidates* $= \emptyset$ **then**
        $A[i^\star] = -2$ {caregiver non-assignment}
        $(A^*, [F^*]) \leftarrow$ OPTIMIZE($\Sigma, A, A^*, [F^*]$)
        $A[i^\star] = -1$ {backtracking: caregiver cancel}
    end if
    Sort *Wounded-Candidates* according to Heuristic (6)
    for all $j \in$ *Wounded-Candidates* **do**
        $A[i^\star] = j$ {caregiver assignment}
        Update team assigned to wounded j
        $(A^*, [F^*]) \leftarrow$ OPTIMIZE($\Sigma, A, A^*, [F^*]$)
        $A[i^\star] = -1$ {backtracking: caregiver cancel}
    end for
end if
return $(A^*, [F^*])$

processed before the most capable ones (this selection order was empirically found to give the best performance).

Candidate wounded are then generated for that caregiver (lines 9-14), restricted to admissible pairs and sorted according to the Heuristic (6) and its same order (line 20).

Each candidate is then tried **recursively** (lines 21-26). The algorithm assigns wounded j to caregiver i, explores the resulting subproblem through a recursive call (line 24), and then backtracks by cancelling this caregiver assignment (line 25). If a caregiver has no valid wounded candidate (i.e. $C = \emptyset$), it is explicitly marked as $A[i] = -2$ (line 16) and, similarly, the recursion continues (line 17).

Overall, this exhaustive function **evaluates all feasible solutions**, and keeps the one with the best objective value.

3.2 Order relation and pruning strategy

In our Branch and Bound algorithm, all operations and comparisons are performed with interval arithmetic. Each assignment therefore generates an objective interval. Complete solutions are compared to each other using their upper bounds (or, as a secondary criterion, their interval widths), as explained in 3.1. In particular, when $\alpha = 1$, optimization is driven by the worst-case scenario.

However, the evaluation of a partial branch cannot be treated in the same way. A branch represents a set of possible solutions, not a single computed solution. To ensure that no potentially better solution is lost, the pruning test must guarantee that a branch is never discarded if it could contain an improvement. For this reason, we compare the upper bound of the best solution already found with the lower bound attainable by the new branch (corresponding to the best case still possible within that branch), as performed in line 4.

Bounding and pruning are intended to accelerate the search. Yet, in this implementation, exactness is deliberately prioritized over execution speed. This pruning strategy is therefore less aggressive, but it guarantees correctness.

4 Application and impact of uncertainty

4.1 Use case with $m = 36$ and $n = 18$

We consider a use case with $m = 36$ caregivers and $n = 18$ wounded individuals. Both the initial injury levels $[v_j] = [\underline{v_j}, \overline{v_j}] \in \mathbb{IR}$ and the treatment success probabilities $[p_{ij}] = [\underline{p_{ij}}, \overline{p_{ij}}] \in \mathbb{IR}$ are uncertain and therefore modeled as intervals. The underlying estimated initial injury levels v_j are:

$$[7, 6, 9, 6, 9, 7, 6, 9, 7, 9, 7, 6, 6, 9, 7, 6, 6, 7, 9, 7, 6, 6, 9, 7].$$

Each v_j is considered as an interval with an asymmetric uncertainty margin around its value: $[v_j] = [v_j - 0.10,\ v_j + 0.25]$.

Similarly, each p_{ij} is represented as an interval of fixed width 0.1: $[p_{ij}] = [p_{ij} - 0.05,\ p_{ij} + 0.05]$. The set of all p_{ij} values is represented in Table 1, where we report the midpoints of the intervals for p_{ij}. For example, $[p_{11}] = [0.75, 0.85]$ is represented by its midpoint, 0.8.

Table 1: Estimated treatment success probabilities p_{ij} for $i = 1, \ldots, 36$

	j=1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
i=1	0.8	0.3	0.2	0.8	0.1	0.2	0.2	0.5	0.8	0.8	0.3	0.5	0.1	0.4	0.5	0.4	0.8	0.5
i=2	0.2	0.2	0.1	0.4	0.1	0.8	0.1	0.4	0.8	0.5	0.4	0.1	0.4	0.4	0.1	0.3	0.5	0.1
i=3	0.3	0.4	0.4	0.3	0.2	0.8	0.3	0.3	0.1	0.5	0.3	0.1	0.1	0.1	0.5	0.2	0.1	0.5
i=4	0.2	0.8	0.3	0.3	0.2	0.8	0.1	0.4	0.3	0.1	0.1	0.4	0.1	0.4	0.8	0.1	0.3	0.1
i=5	0.2	0.8	0.3	0.1	0.4	0.8	0.8	0.5	0.1	0.8	0.2	0.4	0.3	0.8	0.5	0.2	0.3	0.8
i=6	0.3	0.4	0.8	0.5	0.1	0.4	0.8	0.1	0.8	0.8	0.1	0.2	0.5	0.4	0.8	0.5	0.5	0.2
i=7	0.5	0.3	0.2	0.8	0.8	0.8	0.4	0.5	0.8	0.2	0.1	0.1	0.5	0.2	0.2	0.5	0.4	0.2
i=8	0.4	0.4	0.8	0.5	0.1	0.8	0.3	0.1	0.1	0.3	0.1	0.8	0.5	0.1	0.4	0.8	0.4	0.4
i=9	0.1	0.8	0.8	0.1	0.5	0.1	0.8	0.3	0.1	0.4	0.8	0.1	0.4	0.1	0.1	0.2	0.4	0.2
i=10	0.2	0.4	0.1	0.2	0.4	0.1	0.5	0.4	0.2	0.3	0.8	0.8	0.3	0.5	0.8	0.3	0.1	0.1
i=11	0.3	0.1	0.8	0.4	0.5	0.1	0.5	0.4	0.4	0.5	0.4	0.4	0.1	0.4	0.8	0.2	0.2	0.1
i=12	0.5	0.4	0.5	0.5	0.1	0.3	0.4	0.1	0.3	0.8	0.2	0.4	0.4	0.4	0.3	0.8	0.3	0.3
i=13	0.2	0.4	0.4	0.8	0.5	0.2	0.1	0.8	0.5	0.4	0.1	0.4	0.8	0.3	0.3	0.1	0.5	0.2
i=14	0.8	0.4	0.8	0.1	0.2	0.1	0.5	0.1	0.8	0.3	0.1	0.1	0.2	0.2	0.1	0.8	0.3	0.2
i=15	0.4	0.3	0.1	0.4	0.5	0.3	0.2	0.3	0.2	0.3	0.3	0.2	0.8	0.5	0.4	0.3	0.4	0.2
i=16	0.1	0.3	0.8	0.3	0.3	0.5	0.2	0.8	0.4	0.3	0.1	0.5	0.2	0.8	0.4	0.2	0.4	0.2
i=17	0.1	0.2	0.2	0.4	0.3	0.5	0.3	0.2	0.5	0.1	0.1	0.4	0.5	0.8	0.1	0.1	0.5	0.8
i=18	0.5	0.3	0.3	0.4	0.4	0.2	0.2	0.3	0.1	0.4	0.8	0.1	0.3	0.1	0.2	0.8	0.4	0.3
i=19	0.8	0.5	0.1	0.1	0.3	0.8	0.5	0.5	0.4	0.2	0.8	0.8	0.8	0.4	0.2	0.5	0.2	0.5
i=20	0.1	0.4	0.5	0.3	0.5	0.8	0.2	0.4	0.8	0.1	0.1	0.3	0.8	0.5	0.1	0.1	0.4	0.8
i=21	0.4	0.4	0.4	0.4	0.3	0.4	0.8	0.5	0.4	0.1	0.1	0.8	0.8	0.5	0.3	0.1	0.1	0.3
i=22	0.8	0.3	0.2	0.8	0.3	0.1	0.2	0.3	0.3	0.5	0.5	0.4	0.8	0.5	0.4	0.2	0.8	0.8
i=23	0.1	0.5	0.2	0.5	0.1	0.4	0.2	0.8	0.5	0.8	0.4	0.8	0.5	0.4	0.5	0.8	0.8	0.2
i=24	0.2	0.2	0.5	0.3	0.4	0.2	0.1	0.5	0.2	0.4	0.3	0.5	0.2	0.2	0.1	0.2	0.2	0.2
i=25	0.8	0.2	0.2	0.8	0.4	0.5	0.1	0.5	0.5	0.3	0.3	0.4	0.1	0.2	0.2	0.1	0.8	0.2
i=26	0.1	0.4	0.5	0.4	0.4	0.8	0.4	0.1	0.4	0.2	0.5	0.2	0.8	0.1	0.8	0.1	0.3	0.5
i=27	0.3	0.8	0.4	0.1	0.3	0.2	0.5	0.1	0.8	0.3	0.8	0.1	0.8	0.5	0.1	0.2	0.8	0.4
i=28	0.5	0.8	0.8	0.2	0.2	0.1	0.3	0.8	0.4	0.3	0.8	0.8	0.5	0.3	0.1	0.1	0.2	0.1
i=29	0.2	0.2	0.4	0.1	0.3	0.8	0.1	0.2	0.5	0.4	0.3	0.1	0.5	0.3	0.5	0.5	0.4	0.4
i=30	0.5	0.8	0.8	0.5	0.1	0.2	0.5	0.1	0.3	0.1	0.1	0.4	0.5	0.1	0.1	0.3	0.1	0.5
i=31	0.4	0.3	0.1	0.8	0.3	0.4	0.2	0.5	0.5	0.2	0.3	0.4	0.1	0.3	0.1	0.2	0.2	0.5
i=32	0.2	0.2	0.5	0.3	0.3	0.2	0.1	0.5	0.1	0.8	0.3	0.1	0.4	0.5	0.8	0.2	0.8	0.5
i=33	0.2	0.1	0.1	0.8	0.1	0.1	0.8	0.8	0.4	0.5	0.8	0.3	0.4	0.5	0.8	0.1	0.2	0.4
i=34	0.1	0.1	0.3	0.5	0.1	0.8	0.2	0.8	0.2	0.5	0.1	0.2	0.5	0.5	0.2	0.5	0.8	0.8
i=35	0.4	0.2	0.3	0.3	0.2	0.8	0.4	0.2	0.5	0.2	0.5	0.5	0.3	0.1	0.8	0.4	0.2	0.5
i=36	0.4	0.5	0.8	0.2	0.1	0.2	0.8	0.1	0.4	0.2	0.3	0.4	0.2	0.1	0.4	0.2	0.8	0.3

The objective of our optimization problem is to minimize the overall residual injury of the wounded,

$$\arg \min_{x_{ij} \in \{0,1\}} \sum_{j=1}^{18} [v_j] \cdot \prod_{i=1}^{36} \left(1 - [p_{ij}]\right)^{x_{ij}} , \tag{9}$$

subject to constraints (C.1)–(C.5), with $w_j^{\min} = 4.0$ and $w_j^{\max} = 5.0$.

Our problem was implemented in `C++`, using the `Ibex` library to handle interval arithmetic, compute rigorous bounds on the objective function, and return the solutions as interval-valued residual injuries.

4.2 Output Representations

The implementation produces four equivalent forms of output, each offering a different perspective on the same solution: a human-readable textual representation of the best assignment by team in Table 2, a matrix representation of the decision variables x_{ij} in Table 3, an m-tuple where the i-th entry indicates the index of the wounded person assigned to caregiver i or -1 if caregiver i is unassigned in Table 4, and possibly an assignment graph (a bipartite graph with caregivers on one side and the wounded on the other). For example, caregiver 03 is assigned to wounded 08: there is a 1 in the corresponding entry of Table 3, and for $i = 4$ there is an 8 in Table 4.

Table 2: Best assignment by team

Caregiver	Wounded
00	13
01	09
02	09
03	08

Team t_1.

Caregiver	Wounded
04	∅
05	∅
06	12
07	12

Team t_2.

Caregiver	Wounded
08	∅
09	01
10	∅
11	01

Team t_3.

Caregiver	Wounded
12	11
13	14
14	11
15	11

Team t_4.

Caregiver	Wounded
16	05
17	16
18	05
19	17

Team t_5.

Caregiver	Wounded
20	∅
21	∅
22	15
23	06

Team t_6.

Caregiver	Wounded
24	03
25	03
26	10
27	10

Team t_7.

Caregiver	Wounded
28	∅
29	07
30	07
31	00

Team t_8.

Caregiver	Wounded
32	∅
33	04
34	04
35	∅

Team t_9.

4.3 Residual Injury Intervals

An example output of the algorithm is illustrated in Figure 4. The uncertainties in resulting injuries (interval widths) are computed using interval arithmetic based on input uncertainties. The initial injury intervals are shown as green segments.

As expected, the residual injury intervals (blue segments in the figure) are all upper-bounded by the maximum injury constraint (C.5) (red dots), meaning that the wounded are sufficiently treated. The overall residual injury defined in Equation (9) has been globally minimized.

Table 3: Assignment matrix (first 10 caregivers × first 10 wounded). CXX = caregiver, WYY = wounded.

	W00	W01	W02	W03	W04	W05	W06	W07	W08	W09
C00	0	0	0	0	0	0	0	0	0	0
C01	0	0	0	0	0	0	0	0	0	**1**
C02	0	0	0	0	0	0	0	0	0	**1**
C03	0	0	0	0	0	0	0	0	**1**	0
C04	0	0	0	0	0	0	0	0	0	0
C05	0	0	0	0	0	0	0	0	0	0
C06	0	0	0	0	0	0	0	0	0	0
C07	0	0	0	0	0	0	0	0	0	0
C08	0	0	0	0	0	0	0	0	0	0
C09	0	**1**	0	0	0	0	0	0	0	0

Table 4: Assignment as an m-tuple.

13, 9, 9, 8, ∅, ∅, 12, 12, ∅, 1, ∅, 1, 11, 14, 11, 11, 5, 16, 5, 17, ...

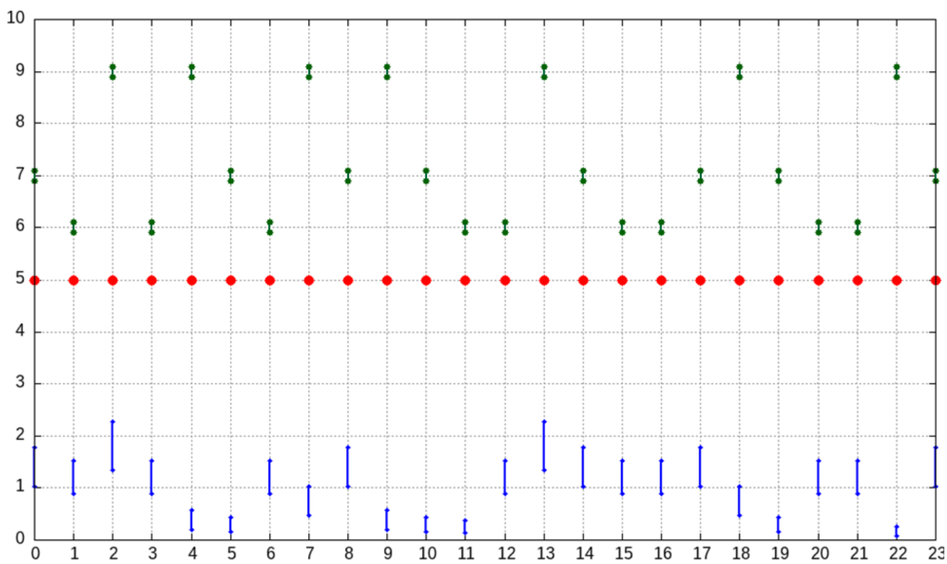

Figure 4: Initial injury intervals (green segments), residual injury intervals (blue segments), maximum injury constraints (red dots).

4.4 Impact of uncertainty about one wounded

We further examined how the pessimistic estimate of an initial injury level (represented by the worst-case bound $\overline{v_j}$) could influence caregiver assignments. To explore this, for $m = 12$ and $n = 12$, and with an initial value $\overline{v_j} = 0$, we focused on wounded person 2 and progressively increased its pessimistic bound $\overline{v_j}$ by a given step δ. In Figure 5, we show the variations of $\overline{v_j}$ (left), $\overline{v_j} + \delta$ with δ ranging from 0 to 58, as well as the resulting assignments (right).

As expected, wounded 2 was **selected more frequently** (in green) as the uncertainty increased: once from the extra margin $\delta = 2$, twice from 10, three times from 34, and four times from 57. The thresholds where an additional assignment occurred are highlighted in orange.

Interestingly, at some uncertainty values (e.g., in blue: 34, 35, 55, and 56), the total number of assignments (here 3) remained unchanged, although the specific caregivers involved differed. This suggests that the resolution can **switch between alternative solutions** without increasing the overall caregiver effort.

To illustrate this effect, Figure 6 shows the worst-case residual injury (y-axis, in green) and the best-case residual injury (y-axis, in purple) as a function of the worst-case initial injury for wounded 2 (x-axis).

As anticipated, the addition of a caregiver reduces the worst-case injury (i.e., the thresholds discussed earlier).

However, between such transitions, increasing uncertainty does not trigger new assignments, which leads to a gradual rise in the worst-case residual injury. Eventually, a saturation effect occurs: although the worst-case injury keeps worsening, the number of caregivers allocated to wounded 2 begins to decline, reflecting the limitation of caregiver resources.

The lower bound of the injuries (in purple), on the other hand, decreases after each assignment and does not increase between assignments. This behavior is expected, since the lower bound is never increased, while each assignment directly contributes to reducing its value.

4.5 Execution times and parallelism

We present the execution times as a function of the variables m (caregivers) and n (wounded), examine the algorithmic complexity, and study the efficiency of the parallelization.

4.5.1 Execution times

In the 3D plot of Figure 7, execution times of our algorithm are shown as a function of m (right-hand axis) and n (left-hand axis). We observe that execution times increase significantly with both parameters, as expected due to the combinatorial nature of our assignment problem. However, the growth rate is greater with respect to m, indicating that the algorithm becomes harder to solve when the number of caregivers increases.

δ	**Assignments as 12-tuple**
0	$(\emptyset, 5, 9, 8, 7, 9, 5, 11, 3, 1, 10, 4)$
1	$(\emptyset, 5, 9, 8, 7, 9, 5, 11, 3, 1, 10, 4)$
2	$(2, 5, 9, 8, 7, 9, 5, 11, 3, 1, 10, 4)$
3	$(2, 5, 9, 8, 7, 9, 5, 11, 3, 1, 10, 4)$
...	...
8	$(2, 5, 9, 8, 7, 9, 5, 11, 3, 1, 10, 4)$
9	$(2, 5, 9, 8, 7, 9, 5, 11, 3, 1, 10, 4)$
10	$(4, 5, 9, 8, 7, 9, 10, 11, 3, 1, 2, 2)$
11	$(4, 5, 9, 8, 7, 9, 10, 11, 3, 1, 2, 2)$
...	...
23	$(4, 5, 9, 8, 7, 9, 10, 11, 3, 1, 2, 2)$
24	$(4, 5, 9, 8, 7, 9, 10, 11, 3, 1, 2, 2)$
25	$(2, 5, 9, 2, 7, 9, 5, 11, 3, 1, 10, 4)$
26	$(2, 5, 9, 2, 7, 9, 5, 11, 3, 1, 10, 4)$
...	...
33	$(2, 5, 9, 2, 7, 9, 5, 11, 3, 1, 10, 4)$
34	$(2, 5, 9, 2, 7, 2, 5, 11, 3, 1, 10, 4)$
35	$(2, 5, 2, 2, 7, 9, 5, 11, 3, 1, 10, 4)$
...	...
55	$(2, 5, 2, 2, 7, 9, 5, 11, 3, 1, 10, 4)$
56	$(2, 5, 9, 2, 7, 2, 5, 11, 3, 1, 10, 4)$
57	$(2, 2, 2, 2, 7, 9, 5, 11, 3, 1, 10, 4)$
58	$(2, 2, 2, 2, 7, 9, 5, 11, 3, 1, 10, 4)$

Figure 5: Evolution of assignments A as a function of δ

To further illustrate this trend, Figure 8 shows 2D execution curves for several fixed values of m, as a function of n. We observe that execution times grow faster with n when m is large. Again, this is consistent with the exponential increase in the size of the search space when both dimensions grow.

Figure 9 shows two theoretical curves plotted to frame the measurements: a lower bound proportional to $m\,n$, and an upper bound proportional to $(n+1)^m$. These curves provide an **empirical enclosure** of the complexity, since all measured points lie between them. This indicates that the computation becomes significantly harder as m increases, with a growth lying between linear and exponential.

4.5.2 Reduced by parallelism

To leverage multicore embedded architectures for real-time decision-making, we parallelized the exploration by distributing the first-level assignment of the initial caregiver across multiple threads. Each thread independently explores a different

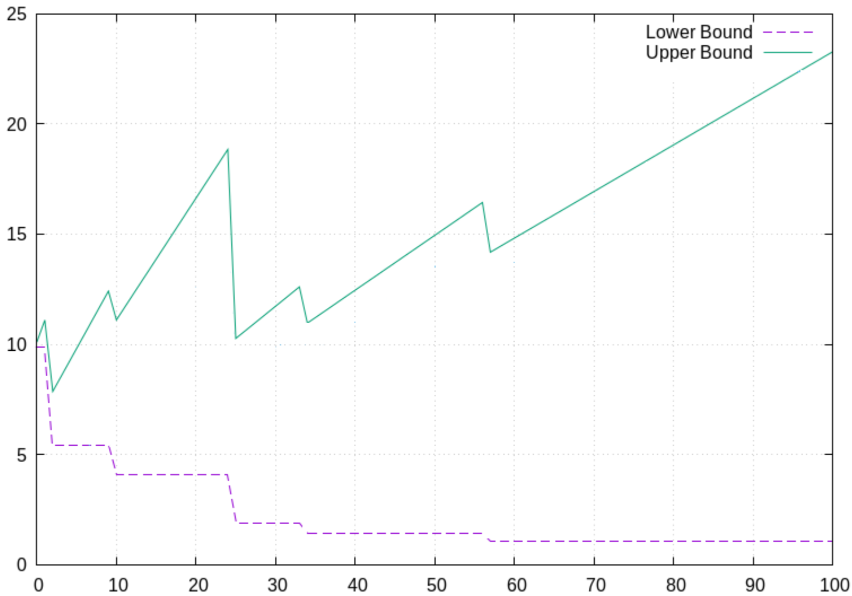

Figure 6: Worst-case residual injury vs uncertainty on wounded person 2

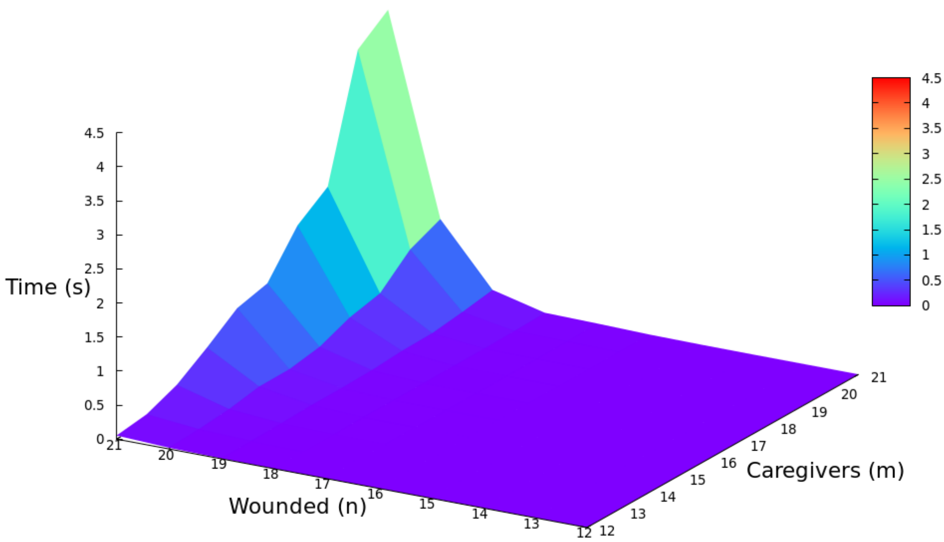

Figure 7: Execution time (in seconds) as a function of the number of caregivers m and wounded n

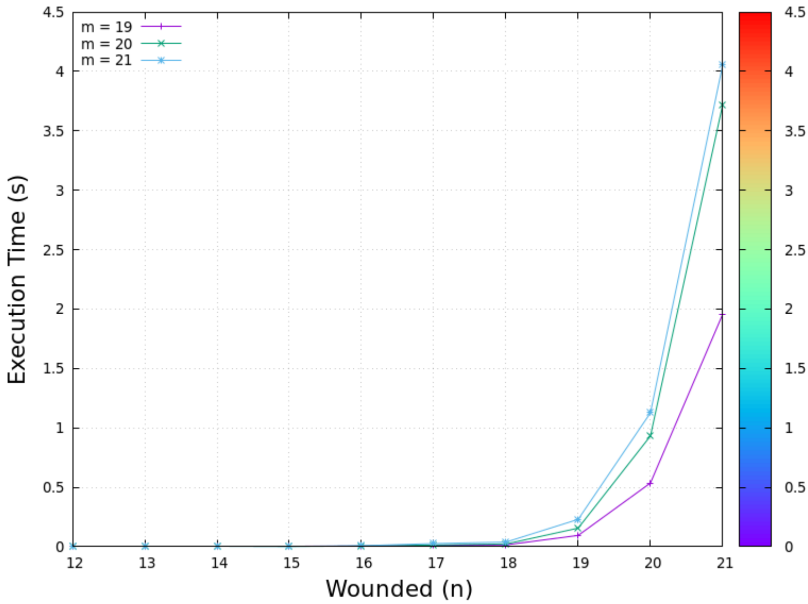

Figure 8: Execution time vs number of wounded n, for three different numbers of caregivers m

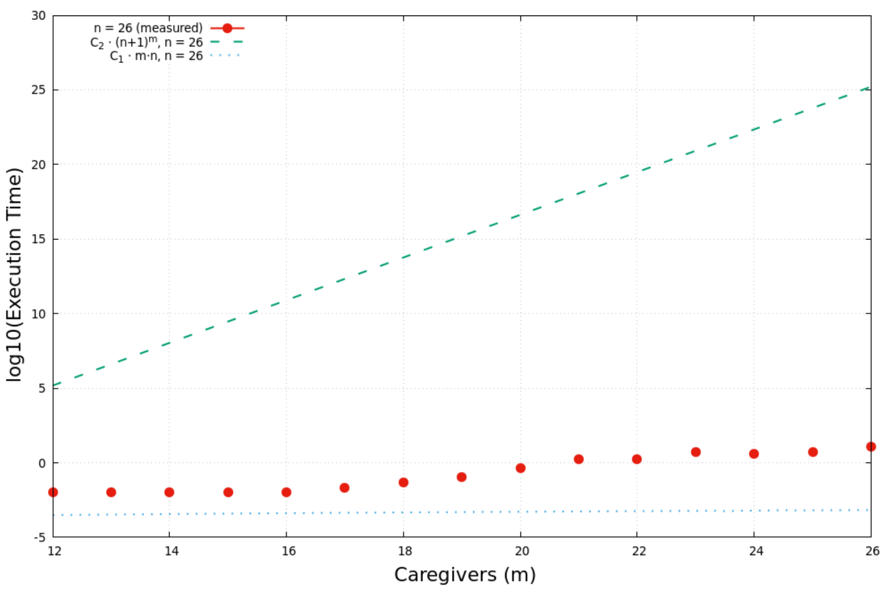

Figure 9: Empirical complexity vs number of caregivers m

initial choice, while the remaining recursive search proceeds sequentially within the thread. This preserves the structure of the algorithm while enabling substantial speedups under strict timing constraints. Figure 10 illustrates the performance gains from this parallelization on an instance with m=36 caregivers and n=18 wounded. On the Figure 10, execution time is shown as a function of the number of cores. It decreases significantly up to 8 cores, then more gradually up to 16 cores. While the execution times decreases with the number of cores, it eventually saturates. This is expected, as the deeper levels of the recursive search rely on sequential backtracking, bounding, and constraint checks that are harder to parallelize efficiently.

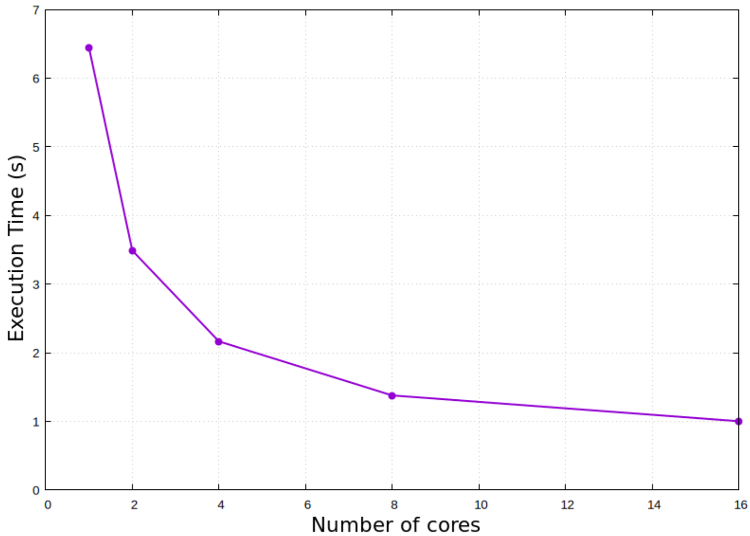

Figure 10: Execution time vs number of cores

5 Conclusion and future works

In this paper, we address a humanitarian scenario in which caregivers are assigned to the wounded, and we show how it corresponds to an instance of the IWTAP. Our problem is characterized by being **unbalanced**, **many-to-one**, with a **multiplicative objective function** defined under interval uncertainty. The state of the art showed that this type of problem had not yet been solved. We solved it using a parallelized Branch and Bound algorithm that directly manipulates the intervals. In an example of resolving this problem, we showed that we ensure a rigorous propagation of uncertainty over the parameters in residual injury intervals. The analysis also highlighted the impact of increasing pessimism on a single wounded individual's injury level, and how uncertainty may trigger changes in assignments.

As future work, we aim to investigate the interdependence between planning and assignment. For example, caregiver movements may be linked to assignments, as illustrated in Figure 11. The process depends on an initial state, which itself results from a planning phase, while planning is in turn influenced by assignment outcomes. This creates a loop of mutual dependence, further complicated by uncertainty. This interplay will be explored in the context of planning under uncertainty, integrating Hierarchical Task Network planning, assignment, and uncertainty into a coherent decision-making framework.

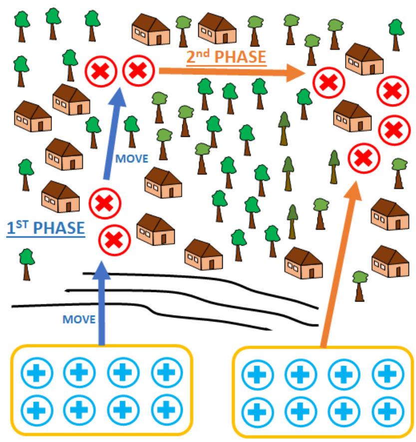

Figure 11: Planning and assignments

References

[1] Achenie, L. E. K. and Sinha, M. Interval global optimization in solvent design. *Reliable Computing*, 9(5):317–338, 2003. DOI: 10.1023/A:1025158512652.

[2] Buvaneshwari, T. and Anuradha, D. Solving bi-objective interval assignment problem using genetic approach. *Advances in Mathematics: Scientific Journal*, 10:759–768, 2021. DOI: 10.37418/amsj.10.2.7.

[3] Buvaneshwari, T. and Anuradha, D. On solving bi-objective interval valued neutrosophic assignment problem. *Neutrosophic Sets and Systems*, 64:212–229, 2024. DOI: 10.5281/zenodo.10729919.

[4] Çetin, E. A queuing theoretical model for anticancer tool selection. In *International Mathematical Forum*, Volume 2-54, pages 2675–2685, 2007. DOI: 10.12988/imf.2007.07238.

[5] Cetin, E. and Esen, S. T. A weapon–target assignment approach to media allocation. *Applied Mathematics and Computation*, 175(2):1266–1275, 2006. DOI: 10.1016/j.amc.2005.08.041.

[6] Chenouard, R. and El-Sehiemy, R. A. An interval branch and bound global optimization algorithm for parameter estimation of three photovoltaic models. *Energy Conversion and Management*, 205:112400, 2020. DOI: 10.1016/j.enconman.2019.112400.

[7] Elsisy, M. A., Elsaadany, A. S., and El Sayed, M. A. Using interval operations in the hungarian method to solve the fuzzy assignment problem and its application in the rehabilitation problem of valuable buildings in egypt. *Complexity*, 2020(1):9207650, 2020. DOI: 10.1155/2020/9207650.

[8] Fahle, T., Schamberger, S., and Sellmann, M. Symmetry breaking. In *International Conference on Principles and Practice of Constraint Programming*, pages 93–107. Springer, 2001. DOI: 10.1007/3-540-45578-7_7.

[9] Ferson, S. and Ginzburg, L. R. Different methods are needed to propagate ignorance and variability. *Reliability Engineering & System Safety*, 54(2-3):133–144, 1996. DOI: 10.1016/S0951-8320(96)00071-3.

[10] Fränzle, M., Herde, C., Teige, T., Ratschan, S., and Schubert, T. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modelling and Computation*, 1(3-4):209–236, 2006. DOI: DOI:10.3233/SAT190012.

[11] Gelenbe, E., Timotheou, S., and Nicholson, D. Fast distributed near-optimum assignment of assets to tasks. *The Computer Journal*, 53(9):1360–1369, 2010. DOI: 10.1016/j.cor.2016.07.005.

[12] Gong, D.-w. and Zhang, Y. Particle swarm optimisation for multi-project location problems with interval profits. *International Journal of Modelling, Identification and Control*, 8(4):335–343, 2009. URL: https://www.inderscienceonline.com/doi/abs/10.1504/IJMIC.2009.03008.

[13] Hansen, E. and Walster, G. W. *Global optimization using interval analysis: revised and expanded*, Volume 264. CRC press, 2003. DOI: 10.1604/9780824740597.

[14] IBEX Team. IBEX, 2007-2020. URL: https://ibex-team.github.io/ibex-lib/.

[15] Ishibuchi, H. and Tanaka, H. Multiobjective programming in optimization of the interval objective function. *European Journal of Operational Research*, 48(2):219–225, 1990. DOI: 10.1016/0377-2217(90)90375-L.

[16] Jiang, C., Han, X., Liu, G. R., and Liu, G. A nonlinear interval number programming method for uncertain optimization problems. *European Journal of Operational Research*, 188(1):1–13, 2008. DOI: 10.1016/j.ejor.2007.03.031.

[17] Jiang, C., Zheng, J., and Han, X. Probability-interval hybrid uncertainty analysis for structures with both aleatory and epistemic uncertainties: A review. *Structural and Multidisciplinary Optimization*, 57(6):2485–2502, 2018. DOI: 10.1007/s00158-017-1864-4.

[18] Kagade, K. and Bajaj, V. Fuzzy method for solving multi-objective assignment problem with interval cost. *Journal of Statistics and Mathematics*, 1(1):1, 2010. URL: https://bioinfopublication.org/pages/article.php?id=BIA0001583.

[19] Kreinovich, V. Decision making under interval uncertainty (and beyond). In *Human-centric decision-making models for social sciences*, pages 163–193. Springer, 2013. DOI: 10.1007/978-3-642-39307-5_8.

[20] Majumdar, S. Interval linear assignment problems. *Universal Journal of Applied Mathematics*, 1(1):14–16, 2013. DOI: 10.13189/ujam.2013.010103.

[21] Manne, A. S. A target-assignment problem. *Operations Research*, 6(3):346–351, 1958. DOI: 10.1287/opre.6.3.346.

[22] Moore, R. E., Kearfott, R. B., and Cloud, M. J. *Introduction to interval analysis*. SIAM, 2009. DOI: 10.2307/40590421.

[23] Naidoo, S. Applying military techniques used in threat evaluation and weapon assignment to resource allocation for emergency response — A literature survey, 2008. Presented as Seminar for Operations Research Society of South Africa. URL: https://www.researchgate.net/publication/228356931_Applying_Military_Techniques_used_in_Threat_Evaluation_and_Weapon_Assignment_to_Resource_Allocation_for_Emergency_Response-A_Literature_Survey.

[24] Nasseri, H., Darvishi Salokolaei, D., and Yazdani, A. A new approach for solving grey assignment problems. *Control and Optimization in Applied Mathematics*, 2(1):15–28, 2017. URL: https://mathco.journals.pnu.ac.ir/article_4820.html.

[25] Onay, O. A mathematical approach to neuromarketing: A weapon–target assignment model. *International Journal of Academic Research in Business and Social Sciences*, 6(1):164–173, 2016. DOI: 10.6007/IJARBSS/v6-i1/1986.

[26] Ramesh, G. and Ganesan, K. Assignment problem with generalized interval arithmetic. *International Journal of Scientific and Engineering Research*, 6(3):82–85, 2015.

[27] Ramesh, G., Sudha, G., and Ganesan, K. Method of finding an optimal solution for interval balanced and unbalanced assignment problem. In *IOP Conference Series: Materials Science and Engineering*, Volume 912-6, page 062031. IOP Publishing, 2020. DOI: 10.1088/1757-899X/912/6/062031.

[28] Salehi, K. An approach for solving multi-objective assignment problem with interval parameters. *Management Science Letters*, 4(9):2155–2160, 2014. DOI: 10.5267/j.msl.2014.7.031.

[29] Sinha, M., Achenie, L. E. K., and Gani, R. Blanket wash solvent blend design using interval analysis. *Industrial & Engineering Chemistry Research*, 42(3):516–527, 2003. DOI: 10.1016/B978-0-323-95931-5.00006-3.