Weapon-target assignment problem

exact and approximate solution algorithms

Andersen, Alexandre Colaers; Pavlikov, Konstantin; Toffolo, Túlio A. M.

Published in:
Annals of Operations Research

DOI:
10.1007/s10479-022-04525-6

Publication date:
2022

Document version:
Accepted manuscript

Citation for pulished version (APA):
Andersen, A. C., Pavlikov, K., & Toffolo, T. A. M. (2022). Weapon-target assignment problem: exact and approximate solution algorithms. *Annals of Operations Research*, *312*(2), 581-606. https://doi.org/10.1007/s10479-022-04525-6

Weapon-Target Assignment Problem: Exact and Approximate Solution Algorithms

Alexandre Colaers Andersen[1], Konstantin Pavlikov[1], Túlio A. M. Toffolo[2]

[1] *Department of Business and Economics, University of Southern Denmark*
Campusvej 55, 5230 Odense M, Denmark
[2] *Departament of Computing, Federal University of Ouro Preto*
Campus Universitário Morro do Cruzeiro, 35400-000 Ouro Preto/MG, Brazil

E-mails: alcoan@gmail.com, kpavlikov@mail.com, tulio@toffolo.com.br

First draft: August 2018, This draft: December 2021

Correspondence should be addressed to: Konstantin Pavlikov

Abstract

The Weapon-Target Assignment (WTA) problem aims to assign a set of weapons to a number of assets (targets), such that the expected value of survived targets is minimized. The WTA problem is a nonlinear combinatorial optimization problem known to be NP-hard. This paper applies several existing techniques to linearize the WTA problem. One linearization technique (Camm et al., 2002) approximates the nonlinear terms of the WTA problem via convex piecewise linear functions and provides heuristic solutions to the WTA problem. Such approximation problems are, though, relatively easy to solve from the computational point of view even for large-scale problem instances. Another approach proposed by O'Hanley et al. (2013) linearizes the WTA problem exactly at the expense of incorporating a significant number of additional variables and constraints, which makes many large-scale problem instances intractable. Motivated by the results of computational experiments with these existing solution approaches, a specialized new exact solution approach is developed, which is called branch-and-adjust. The proposed solution approach involves the compact piecewise linear convex under-approximation of the WTA objective function and solves the WTA problem exactly. The algorithm builds on top of any existing branch-and-cut or branch-and-bound algorithm and can be implemented using the tools provided by state of the art mixed integer linear programming solvers. Numerical experiments demonstrate that the proposed specialized algorithm is capable of handling very large scale problem instances with up to $1,500$ weapons and $1,000$ targets, obtaining solutions with optimality gaps of up to 2.0% within two hours of computational runtime.

Keywords: Branch-and-Adjust, Integer Programming, Weapon-Target Assignment, Probability Chains

---

1. Introduction

The problem of optimal location of weapons to assets (targets) is called the Weapon-Target Assignment (WTA) problem and is a particularly relevant problem for defense operations research. The problem belongs to a large family of nonlinear assignment problems, which consists of a variety of other problems, such as facility location problems (Camm et al., 2002;

O'Hanley et al., 2013), media allocation (Cetin and Esen, 2006), and radiation treatment (Esen et al., 2008) problems. This means that existing solution approaches developed and employed in one application domain may be applied to other problems within the family.

1.1. Literature Review

The WTA problem dates back to end 1950s, and has been widely studied since. Manne (1958) formally introduced the problem after it was informally described by Merrill Flood at The Princeton University Conference on Linear Programming on March 13-15, 1957. In his paper, Manne showed that it was possible to devise a linear programming formulation of the problem under the assumption of all available weapons being identical. Shortly after, den-Broeder Jr et al. (1959) developed an algorithm that yielded an optimum assignment to the problem presented by Manne by sequentially assigning weapons to targets that have the maximum marginal decrease in survival probability. This algorithm is referred to as the Maximum Marginal Return algorithm. Further investigation on the dimensionality and complexity of the problem was presented by Day (1966), where the author achieved a considerable reduction of the dimension by decomposing the allocation problem into smaller targeting problems that provide information for solving the larger targeting problem.

By the end of the 1960s, Matlin (1970) had presented a comprehensive review of the literature for this problem and its variations, categorizing the complexity of assumptions for various models. Supplementing this, Eckler and Burr (1972) and Murphey (2000) went into further details on its variations, and described offensive and defensive formulations of the WTA, as well as how a single assignment of all the weapons (known as the static WTA) extends to the dynamic WTA, when assignments are made at several discrete points in time. Chang et al. (1987) proposed an algorithm to obtain a near optimal solution for the large-scale weapon-target assignment problem. Assuming that at most a single weapon can be assigned to each target, a near optimal assignment of the full problem can be obtained. Wacholder (1989) carried out simulations of a neural network-based algorithm for the static WTA, which proved to converge towards solutions very close to global optima.

A suite of solution algorithms was presented by Metler et al. (1990), in which the problem is broken down into two-stage subproblems solved separately using heuristic algorithms. Subsequently, Ahuja et al. (2007) proposed several lower-bounding schemes as construction heuristics, improved with a very large-scale neighborhood search. Various other heuristic solution algorithms have been proposed for the WTA problem, see Sonuc et al. (2017) and references therein. Recently, an exact solution algorithm to the WTA problem based on the column generation idea appeared in Lu and Chen (2021). A comprehensive review of weapon-target assignment models and solution algorithms can be found in Kline et al. (2019).

1.2. Problem Description

Various formulations of the WTA problem are provided in the literature, each with slight modifications and different assumptions. Matlin (1970) presented a comprehensive review of the variety of WTA problems, in which he expressed the necessity of simplifying the complete model. He also explained how the assumptions made define the differences between the models. He divided the problem into four submodels: the weapon system, the target complex, the engagement, and the damage model. Within each of the submodels, various assumptions are made.

The weapon system submodel describes whether there is a single or multiple weapon types available. It also defines whether all weapons can reach every target, and if the weapon's damage is deterministic or probabilistic. *The target complex* is characterized by the types of targets. The type depends on whether a single weapon can attack a target and the values or weights of the targets. Targets may, for example, have equal or unequal values or be ordered by priority. *The damage* submodel determines whether the damage is partial or total (deterministic or probabilistic). Partial damage occurs when the target values may partially be accrued. Total damage assumption is used when the state of the target can be observed to either having survived or been destroyed after an attack. *The engagement* submodel defines the probability that a weapon destroys a target. This probability depends on the weapon and target, as well as on a defensive system, i.e., a system that may intercept the attack of the assigned weapon.

The formulation of the WTA problem may be perceived in either a defensive or offensive setting. For a defensive formulation, consider a scenario where one's assets are under attack. The assets need to be protected by intercepting the offensive weapons with one's defensive weapons. In such example, the objective may be to minimize the expected damage to one's assets, by assigning the defensive weapons to the missiles attacking the assets. Here it is assumed that every defensive weapon has a specified reliability, i.e., probability to successfully intercept an offensive weapon. Similarly, the non-intercepted weapon may destroy an asset with some other probability. Clearly for the simplest example of this problem, the defender may observe which assets are being attacked and find out the corresponding survival probabilities of the assets before assigning the defensive weapons to the offensive ones. This allows the defender to save the most valuable assets with higher certainty. Variations of this problem include situations without not knowing which assets are being attacked or not knowing the expected damage the non-intercepted weapons will do.

For an offensive formulation, consider having a range of available weapons of different types to be fired at some targets. Assume the values of each target as well as the probabilities of destroying each target with a single weapon of each type are provided. Then the objective is to determine which weapons to assign to which targets, such that the expected damage caused is maximized, without exceeding the number of available weapons. This is the problem this paper is based upon. A variation of this may include the assumption about uniform weapons, such that all weapons available are assumed to be the same. This assumption simplifies the solution process as demonstrated by denBroeder Jr et al. (1959). Another more realistic model may introduce appropriate ranges for every weapon, such that not all weapons may reach all targets.

In the above formulations for the defensive and offensive problems, it is implied that they are static. Given the probabilities, the problem is static if the value of the assets or targets are first observed, after which the weapons are assigned as to optimize the objective. Hence a single assignment for a single period in time is solved. In the dynamic WTA problem, however, assignments are made during multiple periods of time (Murphey, 2000). An example of such a problem is the *shoot-look-shoot* strategy, where the impact may be observed after a subset of the total available weapons are assigned, and before assigning the remaining weapons. This allows the attacker to observe whether the targets survive the first attack, and assign new weapons to the surviving priority targets, and adjust for any inaccurate shots.

1.3. Contributions of the Current Paper

This paper contributes to the literature in the following ways. Given that the WTA problem is a nonlinear integer optimization problem, the paper briefly describes how to apply several existing approaches to linearize the problem, compares their advantages and disadvantages. Most importantly, this paper further develops a specialized exact algorithm to solve the WTA problem using solely a compact convex under-approximation of the nonlinear objective function.

The main novel idea of the algorithm is to use a compact piecewise linear under-approximation of the WTA objective function to find the lower bounds and to guide branching, while resorting to the exact value of the nonlinear WTA objective for bounding in the branch-and-bound framework. Hence, the algorithm can be built on top of any branch-and-bound algorithm by introducing simple manual adjustments to the objective function at the incumbent nodes. Therefore, a more accurate name for the algorithm would be branch-and-adjust-and-bound, which we reduce to branch-and-adjust for brevity. The proposed solution approach can be implemented using state of the art mixed integer linear optimization software, and handles very large scale WTA problem instances. When it does not solve a problem instance to optimality within a specified time limit, it provides a very reasonable optimality guarantee in terms of a small optimality gap on the best solution obtained. Compared to the solution approach by Lu and Chen (2021), which was reported to handle problem instances with up to 400 weapons and 400 targets, the proposed algorithm in our experiments solved instances with up to 400 weapon types and 800 targets within minutes of computer time to optimality and significantly larger instances to suboptimality with small gaps within two hours of computer time. On top of that, the proposed solution approach is general and can be successfully applied in other application domains besides WTA.

The paper is organized as follows. Section 2 defines the WTA as a nonlinear optimization problem, Sections 3 and 4 describe two linearization approaches to the problem, Section 5 proposes a hybrid linearization approach that merges the two existing linearization approaches. Finally, Section 6 describes a new exact algorithm to solve the WTA problem and Section 7 presents results of all computational experiments solving WTA problem instances. Section 8 concludes.

2. Nonlinear Integer Programming Problem Formulation

This paper takes the perspective of the offensive WTA problem with focus on the static version. This is what will henceforth simply be referred to as the WTA problem. The WTA problem may formally be presented as a nonlinear integer programming optimization problem. Let $\mathcal{W} = \{1, \dots, m\}$ be the set of m different weapon types, and let $\mathcal{T} = \{1, \dots, n\}$ be the set of n targets to which the available weapons can be assigned to. Each target has an associated value (or weight), $w_j > 0$, that denotes relative importance of the target. Another integer parameter μ_i, $i \in \mathcal{W}$ denotes the number of weapons available for assignment for every weapon type. Denote by $p_{ij} \in (0, 1)$ the probability of destroying target j with a single weapon of type i. Finally, let the decision variable x_{ij} denote the number of weapons of type i assigned to target j. Assuming that weapons destroy the targets independently of each other, the probability that target j survives the assignment of x_{ij} weapons of type i is computed as

$$\mathbb{P}(\text{target } j \text{ survives weapon } i) = (1 - p_{ij})^{x_{ij}}.$$

Therefore, the survival probability of target j given the assignment of weapons to this target is determined by

$$\mathbb{P}(\text{target } j \text{ survives}) = \prod_{i \in \mathcal{W}} (1 - p_{ij})^{x_{ij}}. \tag{2.1}$$

The WTA problem aims to find an assignment of weapons to targets such that the expected value of survived targets is minimized, and can be formulated as follows (Manne, 1958):

$$\textbf{WTA:} \quad \min_{x_{ij}} \sum_{j \in \mathcal{T}} w_j \prod_{i \in \mathcal{W}} (1 - p_{ij})^{x_{ij}} \tag{2.2}$$

subject to

$$\sum_{j \in \mathcal{T}} x_{ij} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{2.3}$$

$$x_{ij} \in \mathbb{Z}_+, \qquad i \in \mathcal{W}, j \in \mathcal{T}, \tag{2.4}$$

where expression (2.2) denotes the objective to minimize the expected survived value of all targets, constraints (2.3) specify the maximum number of weapons utilized in the assignment for each weapon type. In addition, constraints (2.4) restrict decision variables to be non-negative integers.

3. Linear Approximation to WTA Problem

Arguably, the main challenge of the WTA problem is handling the nonlinear terms $\prod_{i \in \mathcal{W}}(1 - p_{ij})^{x_{ij}}$, i.e., the probability that target j survives an assignment of weapons. This section presents a linear approximation for this product initially proposed by Manne (1958) and investigated further by Camm et al. (2002) for a conceptually similar Nature Reserve Site Selection problem. Consider the following logarithmic transformation

$$z_j = \prod_{i \in \mathcal{W}} (1 - p_{ij})^{x_{ij}}, \qquad \ln(z_j) = \sum_{i \in \mathcal{W}} x_{ij} \ln(1 - p_{ij}), \tag{3.1}$$

so that the WTA problem can be equivalently reformulated as

$$\min_{x_{ij}} \quad \sum_{j \in \mathcal{T}} w_j e^{\ln(z_j)} \tag{3.2}$$

subject to

$$\ln(z_j) = \sum_{i \in \mathcal{W}} x_{ij} \ln(1 - p_{ij}), \qquad j \in \mathcal{T}, \tag{3.3}$$

$$\sum_{j \in \mathcal{T}} x_{ij} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{3.4}$$

$$x_{ij} \in \mathbb{Z}_+, \qquad i \in \mathcal{W}, j \in \mathcal{T}. \tag{3.5}$$

Let $\bar{f}(x) : \mathbb{R} \to \mathbb{R}$ be a convex piecewise linear function that approximates e^x from above. Then, the following program approximates the WTA problem from above:

$$\min_{x_{ij}} \quad \sum_{j \in \mathcal{T}} w_j \bar{f}(\ln(z_j)) \tag{3.6}$$

subject to

$$\ln(z_j) = \sum_{i \in \mathcal{W}} x_{ij} \ln(1 - p_{ij}), \qquad j \in \mathcal{T}, \tag{3.7}$$

$$\sum_{j \in \mathcal{T}} x_{ij} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{3.8}$$

$$x_{ij} \in \mathbb{Z}_+, \qquad i \in \mathcal{W},\ j \in \mathcal{T}, \tag{3.9}$$

where function $\bar{f}(x)$ is formally defined as follows (see Figure 1 for an illustration):

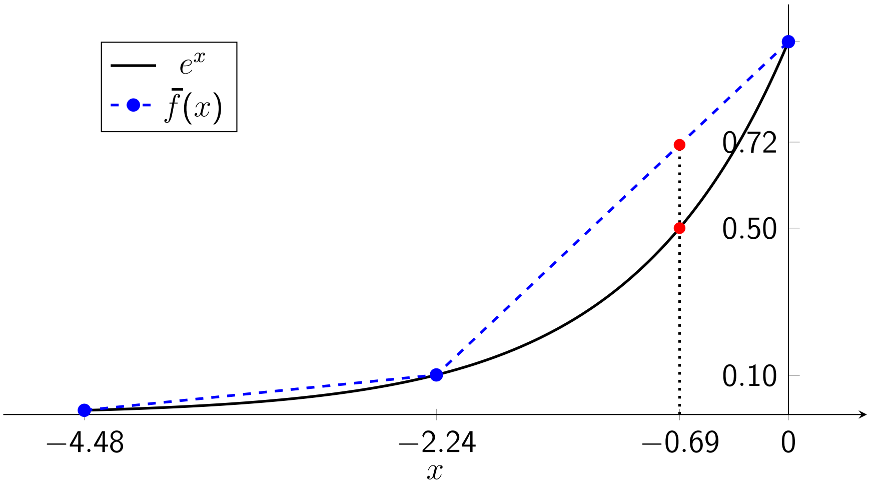

Figure 1: Plot of the e^x curve and its piecewise linear convex over-approximation $\bar{f}(x)$. The set of breakpoints in this example is chosen to be $\mathcal{B} = \{-4.48, -2.24, 0\}$.

$$\bar{f}(x) = \min_{\lambda_t} \sum_{t=1}^{|\mathcal{B}|} \lambda_t e^{b_t} \tag{3.10}$$

subject to

$$\sum_{t=1}^{|\mathcal{B}|} \lambda_t b_t = x, \tag{3.11}$$

$$\sum_{t=1}^{|\mathcal{B}|} \lambda_t = 1, \tag{3.12}$$

$$\lambda_t \geq 0, \qquad t = 1, \ldots, |\mathcal{B}|. \tag{3.13}$$

The set of breakpoints $\mathcal{B}$ to linearize the exponential function should be different from target to target. The main reason for this has to deal with defining the left point b_1 of the approximation domain for every target ($b_1 = -4.48$ in an example presented in Figure 1). The point b_1 corresponds to the smallest possible survival probability e^{b_1} for target j over all possible assignments. Clearly, this probability can be obtained if all available weapons of all weapon types are assigned to target j. Therefore, the smallest breakpoint for target j is defined as follows:

$$b_{j1} = \ln\left(\prod_{i \in \mathcal{W}} (1 - p_{ij})^{\mu_i}\right). \tag{3.14}$$

The number of breakpoints determines the accuracy of the approximation, at the cost of introducing additional variables. For each additional breakpoint b_t for a target, a new variable λ_t is introduced. Therefore, making a decision about the number of breakpoints and their locations is crucial for the computational runtime of an optimization algorithm for a problem instance. A straightforward way to define the breakpoints is to simply choose a number of uniformly distributed breakpoints. This way, the remaining breakpoints b_{jt}, $t > 1$ can be defined as follows:

$$b_{jt} = b_{j1} + (t-1)\frac{|b_{j1}|}{n-1}, \qquad t = 2, \ldots, n. \tag{3.15}$$

where $n = |\mathcal{B}|$ is the size of the breakpoint set. An example with $n = 3$ uniform breakpoints is presented in Figure 1.

The uniform breakpoint allocation approach is easy to implement and provides a good foundation for understanding the linear approximation method. However, this definition of breakpoints has some disadvantages: it may introduce unnecessary line segments where the exponential curve is already close to being linear. At the same time, it may fail to introduce breakpoints where the exponential function is far from being linear. In other words, the uniform breakpoint allocation may lead to excessive approximation error. Specifically, the approximation error, defined as $\bar{f}(x) - e^x$, is generally greater in the interval $(-2.24, 0)$ compared to the error in the interval $(-4.48, -2.24)$. For example, at $x = \ln(0.5) \approx -0.69$, the error of approximation of e^x by $\bar{f}(x)$ is roughly equal to 0.22, which is a rather significant number. At the same time, the approximation error is much smaller for any $x \in (-4.48, -2.24)$. The value 0.22 represents how much the survival probability of a target is overestimated by using $\bar{f}(x)$ compared to its true value, and this again illustrates why the problem given by (3.6)–(3.9) approximates WTA from above and in general does not lead to an optimal solution of the WTA problem. Of course, the maximum error over a segment decreases with the number of approximation segments we introduce, but the value of the maximum error will still be relatively large in segments close to 0. One way to deal with this issue is by introducing breakpoints with non-equal intervals as proposed by Camm et al. (2002), so that the maximum approximation error at every interval is at most δ. Consider a segment $[a, b]$ where e^x is approximated by the line $e^a + k(x-a)$ passing through the two points (a, e^a) and (b, e^b). Let $\delta_{[a,b]} = \max_{x \in [a,b]} e^a + k(x-a) - e^x$ be the maximum approximation error over the $[a, b]$ segment. Then, it is straightforward to demonstrate (by checking the necessary optimality condition) that

$$\delta_{[a,b]} = e^a + k(x^* - a) - e^{x^*}, \text{ such that } x^* : e^{x^*} = k = \frac{e^b - e^a}{b-a}. \tag{3.16}$$

With this result, breakpoints may be defined in such a way that the maximum error in every approximation segment is at most equal to some specified value δ:

$$b_1 = a, \tag{3.17}$$

$$b_t = \min(0, x_t), \qquad t = 2, \ldots, n, \tag{3.18}$$

where x_t is the solution to the system of the following conditions:

$$e^{b_{t-1}} + \frac{e^{x_t} - e^{b_{t-1}}}{x_t - b_{t-1}} \left(\ln\left(\frac{e^{x_t} - e^{b_{t-1}}}{x_t - b_{t-1}}\right) - b_{t-1} \right) - \frac{e^{x_t} - e^{b_{t-1}}}{x_t - b_{t-1}} = \delta, \tag{3.19}$$

$$x_t > b_{t-1}, \tag{3.20}$$

which can be solved numerically. Note that the number of breakpoints n is the outcome of this iterative procedure that depends on δ. Note also that the approximation error in the last interval $[b_{n-1}, b_n]$ may be less than δ if $x_n > 0$. An illustration of approximation with non-uniformly located breakpoints is presented in Figure 2.

Finally, when breakpoints are defined for every target by either of the methods, we obtain

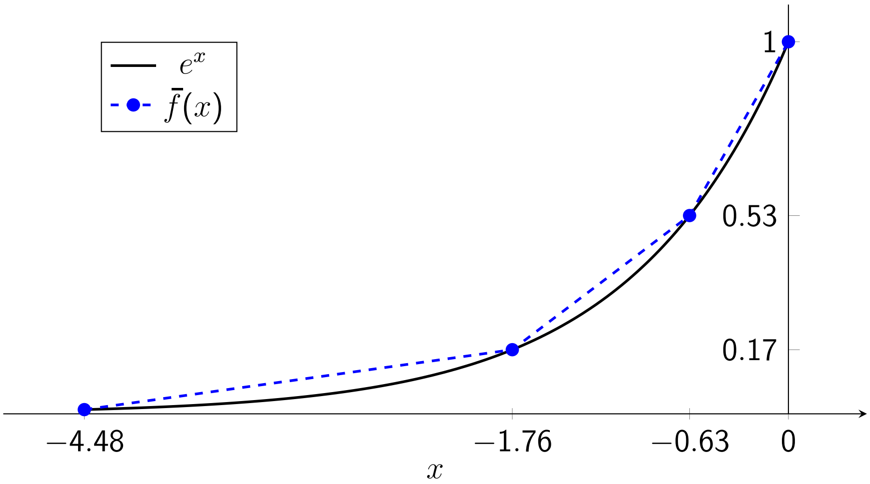

Figure 2: An example of non-uniform breakpoint location using the iterative procedure (3.19)–(3.20) with the parameter $\delta = 0.05$. Only 4 breakpoints are needed to obtain a convex piecewise linear approximation with the maximum error equal to 0.05.

the following mixed integer linear reformulation of the approximate WTA problem:

$$\textbf{WTA\_A:} \quad \min_{x_{ij},\, \lambda_{jt}} \quad \sum_{j \in \mathcal{T}} w_j \sum_{t=1}^{|\mathcal{B}_j|} \lambda_{jt} e^{b_{jt}} \tag{3.21}$$

subject to

$$\sum_{t=1}^{|\mathcal{B}_j|} \lambda_{jt} b_{jt} = \sum_{i \in \mathcal{W}} x_{ij} \ln(1 - p_{ij}), \qquad j \in \mathcal{T}, \tag{3.22}$$

$$\sum_{t=1}^{|\mathcal{B}_j|} \lambda_{jt} = 1, \qquad j \in \mathcal{T}, \tag{3.23}$$

$$\sum_{j \in \mathcal{T}} x_{ij} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{3.24}$$

$$\lambda_{jt} \geq 0, \qquad j \in \mathcal{T},\ t = 1, \ldots, |\mathcal{B}_j|, \tag{3.25}$$

$$x_{ij} \in \mathbb{Z}_+, \qquad i \in \mathcal{W},\ j \in \mathcal{T}. \tag{3.26}$$

Efficiency of this problem formulation and its approximation accuracy is illustrated using a computational study that we present in Section 7. The **WTA_A** problem can be solved even for rather large problem instances. Yet, **WTA_A** remains only an approximation to the original problem, which we will demonstrate by evaluating an optimal solution to the **WTA_A** problem using the nonlinear objective function (2.2). The difference between the original objective (2.2) and the approximate objective (3.21) can be arbitrary large, and solving the approximation problem provides no information concerning the quality of the solution in terms of the original objective. Hence, next sections propose several approaches to deal with the nonlinear WTA objective and solve the problem exactly.

4. Linearization Using Probability Chains

O'Hanley et al. (2013) proposed a general technique to linearize nonlinear terms of type

(2.1) exactly. Linearization is achieved by decomposing every nonlinear term with help of additional continuous variables and constraints, such that the value of some continuous variable corresponds to the value of the nonlinear term. In this section we describe this linearization technique involving the notion of probability chain.

4.1. Probability Chains for the Weapon-Target Assignment Problem

In order to apply the linearization technique proposed by O'Hanley et al. (2013), we first reformulate the WTA problem using binary assignment variables only. Let

$$x_{ijr} = \begin{cases} 1, & \text{if } r \text{ weapons of type } i \text{ are assigned to target } j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

With this notation, the WTA problem defined in Section 2 can be formulated as follows:

$$\min_{x_{ijr}} \sum_{j \in \mathcal{T}} w_j \prod_{i \in \mathcal{W}} \prod_{r=1}^{\mu_i} (1 - p_{ij})^{r x_{ijr}} \tag{4.2}$$

subject to

$$\sum_{j \in \mathcal{T}} \sum_{r=1}^{\mu_i} r x_{ijr} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{4.3}$$

$$\sum_{r=1}^{\mu_i} x_{ijr} \leq 1, \qquad i \in \mathcal{W},\ j \in \mathcal{T}, \tag{4.4}$$

$$x_{ijr} \in \{0, 1\}, \qquad i \in \mathcal{W},\ j \in \mathcal{T},\ r = 1, \dots, \mu_i, \tag{4.5}$$

where constraint (4.4) forces a proper assignment, i.e., that at most one binary assignment variable can be nonzero for every weapon-target pair. Next, two sets of auxiliary variables are introduced in order to form a *probability chain*:

z_{kj} = probability that target j survives the attack of the first k weapon types,
y_{kjr} = probability that target j is destroyed by r weapons of type k, after surviving the attack of the prior $k-1$ weapon types.

Consider the weapon type 1 first. If exactly r weapons of the type 1 are assigned to target j, then the probability for target j to be destroyed by this weapon type is equal to

$$y_{1jr} = 1 - (1 - p_{1j})^r,$$

with

$$y_{1jq} = 0, \qquad \text{for } q \neq r.$$

Hence, the probability that target j survives the attack of weapon type 1 only is equal to

$$z_{1j} = 1 - \sum_{r=1}^{\mu_1} (1 - (1 - p_{1j})^r) x_{1jr}.$$

If the target survives the attack of weapons of type 1, we advance to the weapon type 2 through what is called a probability chain. If exactly r weapons of type 2 are assigned to target j, then the probability for target j to be destroyed by this weapon type is equal to

$$y_{2jr} = z_{1j}(1 - (1 - p_{2j})^r),$$

and

$$y_{2jr} = 0,$$

if other than r number of weapons of type 2 is assigned to target j. Hence,

$$y_{2jr} = z_{1j}\left(1 - (1 - p_{2j})^r\right) x_{2jr}.$$

After linearization of the product $z_{1j}x_{2jr}$ (Glover, 1975), the probability of destroying the target by one of the r weapons of weapon type 2 can be obtained from the following set of inequalities:

$$y_{2jr} \leq (1 - (1 - p_{2j})^r)x_{2jr}, \qquad r = 1, \ldots, \mu_2, \tag{4.6}$$
$$y_{2jr} \leq (1 - (1 - p_{2j})^r)z_{1j}, \qquad r = 1, \ldots, \mu_2, \tag{4.7}$$
$$y_{2jr} \geq (1 - (1 - p_{2j})^r)(z_{1j} + x_{2jr} - 1), \qquad r = 1, \ldots, \mu_2. \tag{4.8}$$

The total probability of being destroyed by weapon type 2 is equal to the sum $\sum_{r=1}^{\mu_2} y_{2jr}$, therefore the probability to survive an attack of the first two weapon types is equal to:

$$z_{2j} = z_{1j} - \sum_{r=1}^{\mu_2} y_{2jr}.$$

Progressing further through the probability chain in the same way, we obtain the probability of surviving an attack of all weapons $z_{|\mathcal{W}|j}$ for every target j via a set of linear constraints. This allows writing down the following exact mixed integer linear programming reformulation of the WTA problem:

$$\textbf{WTA\_PC:} \quad \min_{x_{ijr},\, y_{kjr},\, z_{kj}} \sum_{j \in \mathcal{T}} w_j z_{|\mathcal{W}|j} \tag{4.9}$$

subject to

$$\sum_{j \in \mathcal{T}} \sum_{r=1}^{\mu_i} r x_{ijr} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{4.10}$$

$$\sum_{r=1}^{\mu_i} x_{ijr} \leq 1, \qquad i \in \mathcal{W},\ j \in \mathcal{T}, \tag{4.11}$$

$$z_{1j} = 1 - \sum_{r=1}^{\mu_1} (1 - (1 - p_{1j})^r) x_{1jr}, \qquad j \in \mathcal{T}, \tag{4.12}$$

$$\sum_{r=1}^{\mu_k} y_{kjr} + z_{kj} = z_{k-1,\,j}, \qquad k \in \mathcal{W}\backslash\{1\},\ j \in \mathcal{T}, \tag{4.13}$$

$$y_{kjr} \leq (1 - (1 - p_{kj})^r)x_{kjr}, \qquad k \in \mathcal{W}\backslash\{1\},\ j \in \mathcal{T},\ r = 1, \ldots, \mu_k, \tag{4.14}$$
$$y_{kjr} \leq (1 - (1 - p_{kj})^r)z_{k-1,\,j}, \qquad k \in \mathcal{W}\backslash\{1\},\ j \in \mathcal{T},\ r = 1, \ldots, \mu_k, \tag{4.15}$$
$$y_{kjr} \geq 0, \qquad k \in \mathcal{W}\backslash\{1\},\ j \in \mathcal{T},\ r = 1, \ldots, \mu_k, \tag{4.16}$$
$$z_{kj} \geq 0, \qquad k \in \mathcal{W},\ j \in \mathcal{T}, \tag{4.17}$$

$$x_{ijr} \in \{0, 1\}, \qquad i \in \mathcal{W},\ j \in \mathcal{T},\ r = 1, \ldots, \mu_i. \tag{4.18}$$

Note that the set of constraints (4.8) is not necessary given the nature of the objective function that essentially maximizes variables $y_{|\mathcal{W}|j}$ directly and other variables y_{kj}, $k < |\mathcal{W}|$, indirectly. For this reason we have omitted them from the **WTA_PC** formulation.

Example: To illustrate the concept of probability chain, a small example is provided. Consider the example depicted in Figure 3, which has three weapon types $\mathcal{W} = \{1, 2, 3\}$ that can be assigned to a single target $\mathcal{T} = \{j\}$. Let $\mu_1 = \mu_3 = 2$ and $\mu_2 = 3$. Consider a target that

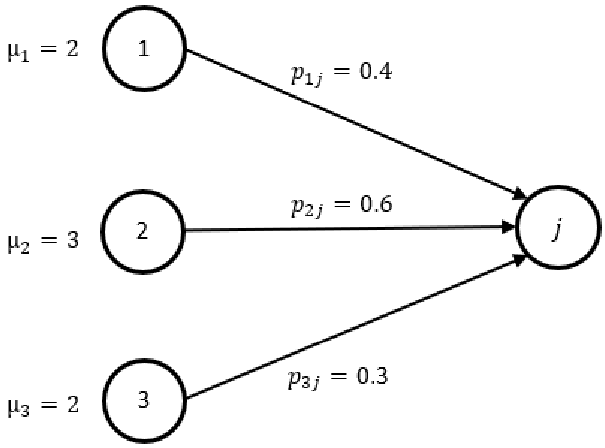

Figure 3: An example of the WTA problem with three weapon types and a single target, where 2 weapons of each of types 1 and 3 are available, together with 3 weapons of type 2. The destruction probabilities are equal to 0.4, 0.6, and 0.3 for weapons of types $i = 1, 2, 3$, respectively.

has been assigned to two weapons of type 1, three weapons of type 2, and a single weapon of type 3, i.e., $x_{1j2} = 1$, $x_{2j3} = 1$, and $x_{3j1} = 1$. Let $\begin{bmatrix} p_{1j} & p_{2j} & p_{3j} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.6 & 0.3 \end{bmatrix}$ denote the probabilities of destroying target j with a single weapon of type i. Given the assignment of weapons, it is straightforward to find the probability of survival for the target:

$$(1 - p_{1j})^2 (1 - p_{2j})^3 (1 - p_{3j}) = 0.6^2 \cdot 0.4^3 \cdot 0.7 = 0.016128. \tag{4.19}$$

Now the goal is to find the same survival probability using linear programming and the probability chain. Figure 4 provides an illustration of a single probability chain for target j and the considered assignment of weapons. The probability to survive the attack of the two weapons of the first weapon type is clearly equal to

$$z_{1j} = (1 - p_{1j})^2.$$

With that, using the equations of the probability chain, we can obtain the probability of surviving the attack of three weapons through the following linear programming problem:

$$\begin{aligned} &\min\ z_{3j} \\ &\text{subject to} \\ &z_{1j} = (1 - p_{1j})^2, \\ &y_{2j3} \le 1 - (1 - p_{2j})^3, \\ &y_{2j3} \le \left(1 - (1 - p_{2j})^3\right) z_{1j}, \end{aligned}$$

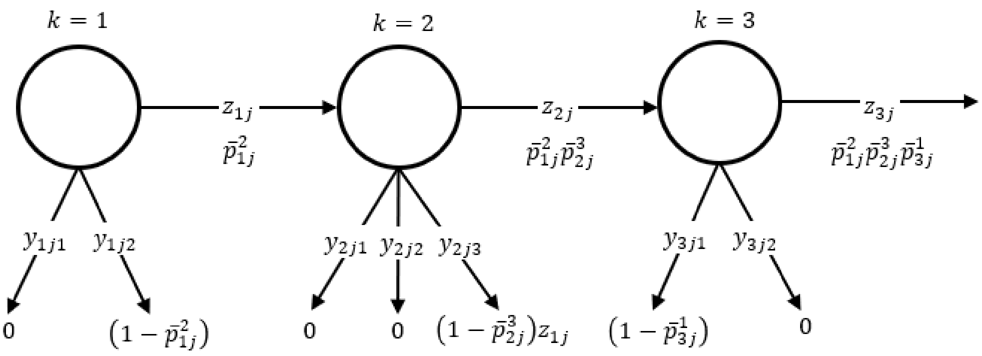

Figure 4: An example of probability chain for some target j and three weapon types $i \in \mathcal{W} = \{1, 2, 3\}$, where the first weapon type has both of its available weapons assigned, the second type has all of this three available weapons assigned, and the third types has one of its two available weapons assigned to the target. Notation $\bar{p}_{ij} = 1 - p_{ij}$ has been used for ease of presentation.

$$
\begin{aligned}
& z_{2j} = z_{1j} - y_{2j3}, \\
& y_{3j1} \leq p_{3j}, \\
& y_{3j1} \leq p_{3j} z_{2j}, \\
& z_{3j} = z_{2j} - y_{3j1}, \\
& y_{2j3},\ y_{3j1},\ z_{1j},\ z_{2j},\ z_{3j} \geq 0.
\end{aligned}
$$

The above formulation omits constraints (4.14) corresponding to variables x_{ijr} that are equal to 0, which make corresponding y variables equal to 0. Given the parameters of this simple example, $z_{1j} = 0.36$. Then, $y_{2j3} \leq 0.36 \times (1 - 0.064) = 0.33696$ and $z_{2j} \geq 0.36 - 0.33696 = 0.02304$. Finally, $y_{2j3} \leq 0.3 z_{2j}$ and $z_{3j} \geq z_{2j} - 0.3 z_{2j} = 0.7 z_{2j} \geq 0.7 \times 0.02304 = 0.016128$, which gives exactly the value of survival probability obtained in (4.19).

Results of computational experiments with formulation **WTA_PC** are presented in Section 7. We observe that only the small problem instances can be solved effectively, i.e., proving a small optimality gap of less than one percent, using the exact linearization via probability chains. Moreover, when the number of available weapons per weapon type increases, the formulation becomes rather useless given the allowed time limit of two hours. This is due to the weak linear programming relaxation of the **WTA_PC** formulation. In fact, it can be observed that the majority of LP relaxations results in lower bounds equal to or close to zero, which is a trivial lower bound value for non-negative objective functions. For this reason, next section combines the two already proposed approaches into one, such that it exactly linearizes the objective value using the same probability chain and has linear programming relaxation strengthened using a convex piecewise linear approximation of the objective function.

5. Probability Chains with Linear Approximation From Below

Probability chains introduced in the previous chapter provide an exact way to linearize the nonlinear product terms of the WTA problem. However, linear relaxations of the resulting models appear to be rather weak and restrict the dimensions of the problems that can be solved to optimality. At the same time, the simple linearization of the WTA problem from Section 3 was much more successful in solving the approximation problem instances to optimality. How can we combine the existing two approaches into one that possesses stronger linear relaxation and still exactly linearizes the WTA problem?

Let us return to the exponential function considered in Section 3. The piecewise linear convex function $\bar{f}(x)$ approximating the exponential function from above and its usage resulted in the over-approximation of the WTA objective as well. If we could solve the problem with the exponential function in the objective – we would solve the WTA problem. Naturally, if we use the piecewise linear under-approximation of the exponential function, we would under-approximate the WTA objective. O'Hanley et al. (2013) considered the following under-approximation piecewise linear convex function

$$\underline{f}(x) = \bar{f}(x) - \delta, \tag{5.1}$$

that is obtained by a vertical shift of the $\bar{f}(x)$ function constructed earlier. By definition of the approximation error $\delta \in (0,\, 1)$

$$\max_{x \in [b_1,\, 0]} \bar{f}(x) - e^x = \delta,$$

one can conclude that

$$\bar{f}(x) - e^x \leq \delta,$$

which is why

$$\bar{f}(x) - \delta \leq e^x.$$

Hence, function $\bar{f}(x) - \delta$ is indeed a convex function approximating the exponential function from below. Figure 5 provides an illustration of the function $\bar{f}(x) - \delta$. Note, however, that this approximation is not ideal. For example, for values of x close to -4.48, the approximation function takes negative values which does not make sense: the exponent is positive everywhere and that is what we would expect from the under-approximation as well. At the same time, since the maximum approximation error $\bar{f}(x)$ over the interval $[-0.63, 0]$ is less than δ, the under-approximation is too conservative in this interval:

$$\max_{x \in [-0.63,\, 0]} \bar{f}(x) - e^x < \delta \implies \min_{x \in [-0.63,\, 0]} e^x - \left(\bar{f}(x) - \delta\right) > 0.$$

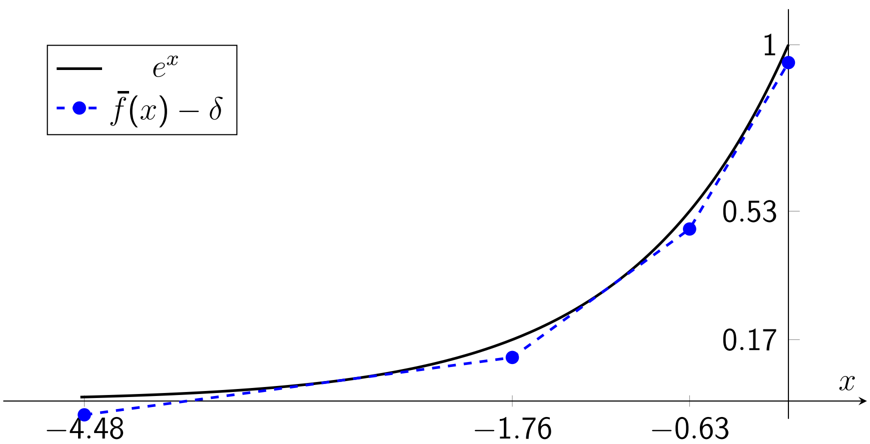

Figure 5: Approximation of the exponential function by a convex piecewise linear function $\underline{f}(x) = \bar{f}(x) - \delta$ from below.

We propose to construct the under-approximation in a similar way we constructed the over-approximation in the earlier section. Starting from the point b_1 defined in (3.14), we draw the tangent line to the function e^x at $x = b_1$. Then, b_2 is the point where the error of approximation becomes exactly equal to the predefined parameter $\delta > 0$:

$$\begin{aligned} & b_2 : \; e^{b_2} - e^{b_1} - e^{b_1}(b_2 - b_1) = \delta, \\ & b_2 > b_1. \end{aligned}$$

The above system can again be solved numerically for the point b_2. Given the point b_2, we draw another tangent line to the exponential function, which intersects it at point $(x_2^*, e^{x_2^*})$, $x_2^* > b_2$. This line can be defined as the line passing through the point $(b_2, e^{b_1} + e^{b_1}(b_2 - b_1))$ with the following slope:

$$\begin{aligned} & k_2 = e^{x_2^*}, \text{ where } e^{b_2} - \delta + e^{x_2^*}(x_2^* - b_2) = e^{x_2^*}, \\ & x_2^* > b_2. \end{aligned}$$

The next breakpoint, b_3, will be obtained when the approximation error again becomes equal to δ:

$$b_3 : \; e^{b_3} - \left(e^{b_2} - \delta + k_2(b_3 - b_2)\right) = \delta.$$

In general, given a breakpoint b_t, we can obtain the next breakpoint b_{t+1} solving the following system of equations:

$$b_{t+1} = \min(0, x), \tag{5.2}$$

$$x : \; e^x - \left(e^{b_t} - \delta + k_t(x - b_t)\right) = \delta, \tag{5.3}$$

$$k_t = e^{x_t^*}, \text{ where } e^{b_t} - \delta + e^{x_t^*}(x_t^* - b_t) = e^{x_t^*}, \; x_t^* > b_t. \tag{5.4}$$

An illustration of the approximation function obtained using this procedure is provided in Figure 6. The new under-approximation convex function can be formally defined as follows:

$$\underline{f}(x) = \min_{\lambda_t} \; \lambda_1 e^{b_1} + \sum_{t=2}^{|\mathcal{B}|-1} \lambda_t \left(e^{b_t} - \delta\right) + \lambda_{|\mathcal{B}|}, \tag{5.5}$$

$$\sum_{t=1}^{|\mathcal{B}|} \lambda_t b_t = x, \tag{5.6}$$

$$\sum_{t=1}^{|\mathcal{B}|} \lambda_t = 1, \tag{5.7}$$

$$\lambda_t \geq 0, \qquad t = 1, \ldots, |\mathcal{B}|. \tag{5.8}$$

Therefore, with help of the under-approximation function $\underline{f_j}(x)$ constructed for target j, the following expression

$$\prod_{i \in \mathcal{W}} (1 - p_{ij})^{x_{ij}} \geq \underline{f_j}\left(\sum_{i \in \mathcal{W}} x_{ij} \ln(1 - p_{ij})\right) \tag{5.9}$$

is a valid inequality for the WTA problem. The idea is to use this valid inequality as a strengthening additional constraint in the linear mixed integer problem reformulation. Specifically, the probability of target j to survive all assigned weapons is expressed as $z_{|\mathcal{W}|j}$, and the following

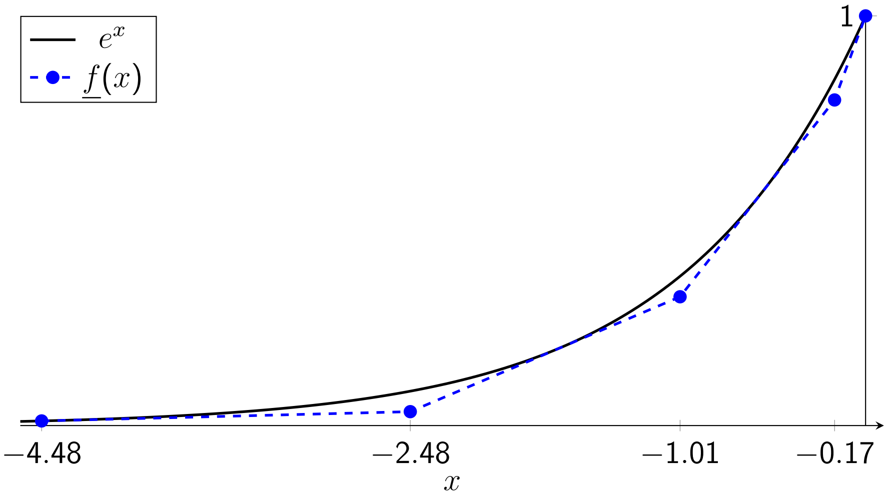

Figure 6: An outcome of the iterative procedure (5.2)–(5.3): function $\underline{f}(x)$ approximates e^x from below with approximation error parameter $\delta = 0.05$.

inequality

$$z_{|\mathcal{W}|j} \geq \underline{f_j}\left(\sum_{i\in\mathcal{W}} x_{ij} \ln(1-p_{ij})\right)$$

is valid and strengthens the probability chain linearization. Therefore, the set of strengthening constraints for the linear formulation of the WTA problem via probability chains are as follows:

$$z_{|\mathcal{W}|j} \geq \lambda_{j1}e^{b_{j1}} + \sum_{t=2}^{|\mathcal{B}_j|-1} \lambda_{jt}\left(e^{b_{jt}} - \delta\right) + \lambda_{j|\mathcal{B}_j|}, \qquad j \in \mathcal{T}, \tag{5.10}$$

$$\sum_{t=1}^{|\mathcal{B}_j|} \lambda_{jt} b_{jt} = \sum_{i\in\mathcal{W}} \ln(1-p_{ij}) \sum_{r=1}^{\mu_i} r x_{ijr}, \qquad j \in \mathcal{T}, \tag{5.11}$$

$$\sum_{t=1}^{|\mathcal{B}_j|} \lambda_{jt} = 1, \qquad j \in \mathcal{T}, \tag{5.12}$$

$$\lambda_{jt} \geq 0, \qquad j \in \mathcal{T},\ t = 1, \ldots, |\mathcal{B}_j|. \tag{5.13}$$

By adding the required strengthening constraints for every probability chain in the **WTA_PC** formulation, we obtain the improved version of the WTA problem formulation. The main advantage of this improvement is flexibility of managing the strength of the resulting formulations. By choosing a small value of parameter δ, the approximation function becomes closer to the exponential function, making the additional constraints (5.10) – (5.13) stronger. Section 7 presents the values of strengthened linear programming relaxations, which are significantly improved over those obtained using the probability chain model only. Computational performance is also significantly improved when considering the mixed integer models, yet, around a half of the considered problem instances have 100% of optimality gap after two hours of CPU time. Aiming at addressing these instances, next section proposes a specialized algorithm that solves the problem exactly using the piecewise linear under-approximation of the objective function only.

6. Branch-and-Adjust Algorithm Based on the Under-Approximation of the Objective Function

Motivated by the computational results of experiments solving the WTA problem via approximate and exact approaches, we observe that exact linearization the problem significantly increases the size of the corresponding problem formulation and hence the solution time. As a result, a problem instance that was tractable to be solved approximately becomes rather intractable after linearization using probability chains. In addition, when the problem is linearized exactly using probability chains and enhanced with inequalities (5.10) – (5.13), we observe a significant speed up in the solution times for small problem instances and increase of the size of problem instances that can be handled using that linearization within the time limit. For that reason, it can be argued that in the combined linearization approach described in the previous section, probability chains are responsible for modeling the original nonlinear objective function correctly and hence allow to solve the problem exactly, while the strength of the linear programming formulations is due to the piecewise linear under-approximation (5.10) – (5.13). Therefore, the motivation for development of the new algorithm is possibility to combine the advantages of both linearization approaches, i.e., use a compact approximate linearization in a way to solve the original WTA problem exactly without introducing a large number of additional variables and constraints to model the nonlinear objective function. The proposed solution approach exploits the possibility to avoid the probability chain linearization and work with compact picewise linear under-approximation only, given that for any feasible assignment of weapons to targets it is trivial to calculate its true nonlinear objective value, and use the new value to prune the nodes in the branch-and-bound tree. The approach is described in what follows.

Consider the following under-approximation to the original WTA problem

$$\textbf{WTA\_LA}: \quad \overset{*}{\mathsf{min}}_{x_{ij}} \quad \sum_{j \in \mathcal{T}} w_j \underline{f_j}(\ln(z_j)) \tag{6.1}$$

subject to

$$\ln(z_j) = \sum_{i \in \mathcal{W}} x_{ij} \ln(1 - p_{ij}), \qquad j \in \mathcal{T}, \tag{6.2}$$

$$\sum_{j \in \mathcal{T}} x_{ij} \leq \mu_i, \qquad i \in \mathcal{W}, \tag{6.3}$$

$$x_{ij} \in \mathbb{Z}_+, \qquad i \in \mathcal{W},\ j \in \mathcal{T}, \tag{6.4}$$

where $\overset{*}{\mathsf{min}}$ denotes the following specialized algorithm for minimizing the objective function. The above problem is solved by a standard branch-and-bound or a branch-and-cut algorithm, with simple adjustments only made at incumbent nodes. When an incumbent solution x^*, i.e., a valid assignment of targets to weapons, is obtained in the branch-and-bound tree, the value of the objective function is replaced by the potentially larger objective value of the original WTA problem:

$$\underline{\text{WTA}}(x^*) := \sum_{j \in \mathcal{T}} w_j \underline{f_j}\left(\sum_{i \in \mathcal{W}} x^*_{ij} \ln(1 - p_{ij})\right) \rightarrow \sum_{j \in \mathcal{T}} w_j \prod_{i \in \mathcal{W}} (1 - p_{ij})^{x^*_{ij}} =: \text{WTA}(x^*). \tag{6.5}$$

The replacement value is used as a new upper bound for pruning remaining nodes in the tree.

If the objective value (6.1) of a node in the branch-and-bound tree is greater than or equal to the true objective value $\text{WTA}(x^*_{ub})$ of the best incumbent solution x^*_{ub}, then this node can clearly be pruned since the WTA objective of any integer solution in that node will not be less than the objective value of the assignment x^*_{ub}. The algorithm stops when there are no nodes to explore in the tree and returns an assignment that is properly evaluated (adjusted) in terms of the original WTA problem objective function. Therefore, these adjustments result in an algorithm that solves the original WTA problem exactly using the lower approximation function only, avoiding complications associated with exact linearization of the WTA problem objective. The adjustment part of the algorithm also gives the branch-and-adjust name for the algorithm, which is a shorter and more convenient version than branch-and-adjust-and-bound that describes the algorithm more accurately. Note, however, that contrary to the case of standard branch-and-bound algorithm, an incumbent node of such adjusted branch-and-bound algorithm may need to be branched further whenever the objective value given by Eq. (6.1) is smaller than the current upper bound of the original WTA problem. In order to see why this is the case, consider an incumbent solution x^* of the **WTA_LA** problem with the corresponding objective value $\underline{\text{WTA}}(x^*)$. If this node is not pruned by optimality, i.e., if $\underline{\text{WTA}}(x^*)$ is strictly less than the current upper bound of the original WTA problem $\text{WTA}(x^*_{ub})$, the objective function of this solution is re-evaluated. Two outcomes of the re-evaluation process are possible:

- $\underline{\text{WTA}}(x^*) = \text{WTA}(x^*)$. In this case, the incumbent solution x^* provides a new upper bound to the WTA problem and this node can be pruned and not considered for further branching.

- $\underline{\text{WTA}}(x^*) < \text{WTA}(x^*)$. In this case, there exists a positive gap between the upper and lower bounds on the minimum WTA objective value in this branch. Depending on the relation between $\text{WTA}(x^*)$ and $\text{WTA}(x^*_{ub})$, solution x^* may or may not provide a new upper bound to the WTA problem. However, there may be another integer solution within this branch of the branch-and-bound tree with an objective value strictly smaller than $\text{WTA}(x^*)$ and $\text{WTA}(x^*_{ub})$. Hence, further branching of this node has to be performed to either find such a solution or prove that it does not exist.

Algorithm 1 presents the primary principles of the specialized algorithm we propose. Only one input is considered: the problem P_0, given by formulation (6.2) – (6.4). The algorithm begins by initializing the set of nodes L, lower bound lb, upper bound ub and best solution x^*_{ub} (line 1). Next, the lower bound is updated (line 2). The algorithm ends when there are no nodes to explore or when the standard optimality gap-based criteria for exiting the algorithm is satisfied (lines 3-4). At each iteration, a node (or a problem) P is selected and removed from set L (lines 5-6). The linear programming relaxation of node P is then solved (line 7), resulting in the solution $\bar{x}$. If the node is infeasible ($\bar{x} = \emptyset$) or if its linear programming relaxation has objective value $\underline{\text{WTA}}(\bar{x})$ greater than or equal to the current upper bound, the node is pruned (lines 8-9). Otherwise, the algorithm checks whether the node's solution $\bar{x}$ is integer or not (line 10). On one hand, if $\bar{x}$ is integer, it will be evaluated using the original objective function of the WTA problem. Whenever the resulting value $\text{WTA}(\bar{x})$ indicates that a better upper bound was obtained, both x^*_{ub} and ub are updated (lines 11-12). As mentioned above, it may be necessary to branch on this node even though it is an integer one. The unfixed variable with the smallest objective coefficient is selected (line 13) for branching. On the other hand, if $\bar{x}$ is not integer, the solver will be responsible for selecting a variable to branch (lines 14-15). Next, if an unfixed variable exists (line 16), two new branches are created (lines 17-20). Note that $\underline{v}$ and $\bar{v}$ represent two appropriate integer values for v; for instance, if $v = 1.5$, then $\underline{v} = 1$ and $\bar{v} = 2$. Finally, the procedure repeats (line 21).

Algorithm 1: Branch-and-Adjust algorithm based on the Under-Approximation of the Objective Function

Initialization:

1 $L \leftarrow \{P_0\}$, $lb \leftarrow 0$, $ub \leftarrow \infty$, $x^*_{ub} \leftarrow \emptyset$

Node selection:

2 $lb \leftarrow$ best known global lower bound so far
3 **if** $L = \emptyset$ **or** $ub - lb \leq \epsilon \times ub$ **then**
4   **return** x^*_{ub}
5 $P \leftarrow$ node in L with smallest lower bound value
6 $L \leftarrow L \backslash \{P\}$

Linear programming relaxation and pruning:

7 $\bar{x} \leftarrow$ optimal solution for the linear programming relaxation of P
8 **if** $\bar{x} = \emptyset$ **or** $\underline{\text{WTA}}(\bar{x}) \geq ub$ **then**
9   **go to Node selection**

Integrality test and variable selection:

10 **if** $\bar{x}$ is an integer solution **then**
11   **if** $\text{WTA}(\bar{x}) < ub$ **then**
12         $x^*_{ub} \leftarrow \bar{x}$, $ub \leftarrow \text{WTA}(\bar{x})$
13     $v \leftarrow$ unfixed variable in P with the smallest $w_j \times \ln(1 - p_{ij})$ objective coefficient
14 **else**
15     $v \leftarrow$ unfixed variable in P selected by the solver

Branching:

16 **if** $v \neq \emptyset$ **then**
17     $(\underline{v}, \bar{v}) \leftarrow$ integer values for variable v that will be used for branching
18     $P' \leftarrow$ new node with all constraints from P and ensuring $v \leq \underline{v}$
19     $P'' \leftarrow$ new node with all constraints from P and ensuring $v \geq \bar{v}$
20     $L \leftarrow L \cup \{P', P''\}$
21 **go to Node selection**

We highlight that the implementation of Algorithm 1 uses the commercial MIP solver CPLEX and three of its callback functions: *BranchCallback*, *BoundCallback* and *IncumbentCallback*[1]. Some internal steps of the algorithm are not detailed here. For instance, CPLEX will improve linear relaxations by adding cuts and run heuristics to produce solutions and improve the upper bound. Note, however, that all incumbent solutions are re-evaluated using the original WTA problem's objective function before their costs are considered by CPLEX.

The specialized algorithm may require solving a prohibitively large number of branch-and-bound nodes. This is mainly due to the natural approximation gap between **WTA_LA** and the original WTA problems, and is also aggravated by the necessity of branching integer

[1]For more information concerning the callback functions, the reader is directed to IBM and ILOG (2020).

nodes. Nevertheless, solving every linear relaxation is relatively easy and, at the same time, the algorithm does provide an optimality guarantee in terms of the valid optimality gap. The optimality gap for minimization problems with non-negative objective functions is defined as

$$\text{Gap} = \frac{\text{Upper bound} - \text{Lower bound}}{\text{Upper bound}} \times 100\%. \tag{6.6}$$

and is the natural outcome of using exact algorithms employed to solve mixed integer linear programming problems. The upper bound is the value of the objective function for the best assignment obtained by a solution algorithm. Next proposition shows validity of the optimality gap obtained by the branch-and-adjust algorithm for the WTA problem.

Proposition 6.1. *Optimality gap obtained using the under-approximation problem* **WTA_LA** *solved with the branch-and-adjust algorithm provides a valid optimality gap to the original WTA problem.*

Proof. First of all, note that the Upper bound in the definition of the optimality gap (6.6) for the **WTA_LA** problem always refers to the true value of the best solution obtained in terms of the original WTA problem, given that the branch-and-adjust algorithm evaluates integer solutions in terms of the original WTA objective function. To prove the proposition, we need to demonstrate that the Lower bound obtained by the branch-and-adjust algorithm based on the under-approximation problem **WTA_LA** is also a lower bound on the optimal objective function of the WTA problem. Consider a set of nodes $\mathcal{P} = \{P_1, \ldots, P_Q\}$ in the branch-and-bound tree of the **WTA_LA** problem. The lower bound based on **WTA_LA** problem is by definition

$$\text{Lower bound}(\textbf{WTA\_LA}) = \min_{P_i \in \mathcal{P}} \underline{\text{WTA}}(P_i), \tag{6.7}$$

where $\underline{\textbf{WTA}}(P_i)$ is the value of the linear programming relaxation of the problem **WTA_LA** over the node P_i. Given the relation

$$\underline{\text{WTA}}(P_i) \le \text{WTA}(P_i), \tag{6.8}$$

that is valid for every node P_i, we obtain that

$$\min_{P_i \in \mathcal{P}} \underline{\text{WTA}}(P_i) \le \min_{P_i \in \mathcal{P}} \text{WTA}(P_i) = \text{Lower bound}(\textbf{WTA}), \tag{6.9}$$

which justifies the claim that the lower bound based on Lower bound(**WTA_LA**) is a valid lower bound to the original WTA problem and finishes the proof of the proposition. □

Computational experiments reveal the very good performance of the proposed algorithm. It can be observed that small problems are solved to optimality within seconds of computer time just like when the exact probability chain-based linearization approach was used. However, larger problem instances, intractable for the exact linearization approach, become tractable using the proposed approach and approximation parameter δ small enough. Solutions provided by the algorithm within the same two hours of CPU time have at most 2.02% of optimality gap for very large problem instances. Such a result provides much better confidence in the quality of the solutions returned by the solver and compares very favorably to much larger optimality gaps obtained with exact linearization method and the same time limit. Detailed experiments are presented in the next section.

7. Computational Experiments

In this section we report computational experiments considering all presented formulations and algorithms using a set of randomly generated problem instances. Specifically, to generate a problem instance with $|\mathcal{W}|$ weapons types and $|\mathcal{T}|$ targets, we only need a set of probabilities

$$\{p_{ij} \in (0,\, 1),\, i \in \mathcal{W},\, j \in \mathcal{T}\}$$

that we obtain by generating $|\mathcal{W}| \times |\mathcal{T}|$ samples of a uniform random variable on $(0,\, 1)$. Then, we assume in our computational study that targets have random weights uniformly distributed between 1 and 100. In addition, the number of available weapons μ_i for each weapon type is assumed to be the same for every weapon type and is denoted simply by μ.

All experiments were executed on an Intel® Xeon® CPU E5620 2.40GHz computer with 128GB of RAM memory running CentOS Linux 7.7.1908. All formulations and the specialized algorithm (Section 6) were implemented[2] and solved using Python 3.7.6 and CPLEX Optimizer 12.10. CPLEX was our choice because it offers additional callback functions which were essential to implement the proposed algorithm. Moreover, CPLEX is currently one of the best-performing commercial mixed integer programming solvers available. Every experiment was executed in the sequential mode (single-threaded) with the CPU time limit of two hours. Optimality gaps for all problem instances are also reported. In many cases, it is not possible to solve a problem instance to optimality within the available time limit. Then, the value of optimality gap serves as performance guarantee to the best solution obtained by an algorithm. Note that we use the symbol $\circledast$ to report optimality gaps that are below the solver default MIP gap tolerance equal to 10^{-4}. In practice, the solver stops when the optimality gap is smaller or equal than this value because it considers the solution to be optimal (within the tolerance). In either case, whether a WTA problem instance is solved to optimality within the time limit or not, we report the best objective value that the solver was able to obtain.

The remainder of this section is organized as follows. Section 7.1 provides results obtained for the approximate WTA problem. Section 7.2 presents results of experiments with formulations that utilize probability chains. The impact of the strengthening constraints is evaluated. Finally, Section 7.3 investigates performance of the specialized algorithm proposed.

7.1. Approximate WTA problem

Table 1 presents results of solving the instances of **WTA_A** problem with various numbers of weapon types and targets. Every problem instance is solved twice, with the approximation error parameter δ equal to 0.001 and 0.0005. Because of solving the approximation problem, it is important to report the true objective (2.2) for the best assignment obtained by the solver. As we can see, the approximation problem is rather efficient in terms of computational time and provides reasonably good solutions to the WTA problem. As expected, approximations with the smaller value of the maximum approximation error parameter δ result in better solutions in terms of the true objective values. However, optimality of such solutions is not guaranteed and the difference between the approximation objective value and the true objective value may be significant even if the value of δ is small.

[2]Source code available at `https://github.com/tuliotoffolo/wta`

Instance			δ	Approx. Obj.	Lower Bound	Gap	CPU (sec.)	True Objective
$\lvert\mathcal{W}\rvert$	$\lvert\mathcal{T}\rvert$	μ						
200	400	1	0.001	4,979.22	4,979.13	⊛	260.4	4,970.57
		2		127.19	127.19	⊛	239.9	115.65
		3		17.74	17.68	0.34%	7,200.1	11.18
250	500	1		6,701.14	6,700.89	⊛	353.1	6,689.89
		2		129.92	129.92	⊛	367.8	114.43
		3		20.85	20.73	0.58%	7,200.3	11.51
300	600	1		8,151.67	8,151.44	⊛	1,123.9	8,137.15
		2		137.88	137.87	⊛	923.7	118.75
		3		25.65	25.39	1.01%	7,225.1	13.74
350	700	1		9,065.90	9,065.31	⊛	1,861.0	9,050.35
		2		140.97	140.96	⊛	1,746.0	121.37
		3		24.00	23.78	0.92%	7,203.6	11.54
200	400	1	0.0005	4,975.09	4,975.09	⊛	235.3	4,970.40
		2		121.78	121.77	⊛	264.9	115.51
		3		17.37	17.33	0.23%	7,200.3	10.96
250	500	1		6,696.14	6,695.80	⊛	508.6	6,690.05
		2		122.34	122.33	⊛	690.7	114.42
		3		20.42	20.36	0.29%	7,200.1	11.40
300	600	1		8,144.25	8,143.83	⊛	1,122.0	8,136.99
		2		127.95	127.94	⊛	648.1	118.58
		3		24.81	23.35	5.88%	7,200.3	13.62
350	700	1		9,058.22	9,058.07	⊛	5,713.2	9,049.88
		2		132.21	132.20	⊛	2,385.4	121.05
		3		24.41	24.25	0.66%	7,200.6	11.27

Table 1: Computational results of the **WTA_A** approximation model.

7.2. Probability chains

Table 2 presents results of computational experiments with the **WTA_PC** formulation. As we can see, no problem instance was solved to optimality within the time limit. Moreover, increasing the number of available weapons for each weapon type, the corresponding optimization problems become less tractable: even for the smallest problem instance with 200 weapon types and 400 targets with 2 weapons per weapon type, the optimality gap is over 97%. When the number of available weapons per weapon type is equal to 3, then it is not possible to obtain any meaningful (strictly positive) lower bound to the objective function, which is by definition non-negative. This observation suggests that linear relaxations of probability chains are rather weak. The last two columns of Table 2 provide values of linear programming relaxations and corresponding CPU times of the **WTA_PC** formulation for the same problem instances, and we indeed observe that linear programming relaxations are weak (equal to or slightly greater than zero) for most of the problem instances.

We experiment further with improving the quality of linear programming relaxations of the probability chain-based models and study how this helps solving WTA problems to optimality. In our computational experiments we study the effect of strengthening constraints (5.10) – (5.13) on the efficiency of the **WTA_PC** formulation. As before, the time limit of two hours was employed in all experiments. Same values of the parameter δ were employed, 0.001 and 0.0005. First, we can compare the values of linear programming relaxations of the improved formulation to the linear programming relaxations of the standard formulation via probability chains. The results are presented in the last two columns of Table 3 and can be compared to

Instance $\|\mathcal{W}\|$	$\|\mathcal{T}\|$	μ	Objective	Lower Bound	Gap	CPU (sec.)	LP relaxation	CPU (sec.)
		1	4,971.29	4,934.01	0.75%	7,200.1	4,929.00	21.2
200	400	2	125.66	3.17	97.47%	7,200.5	2.69	117.4
		3	569.23	0.00	100.00%	7,200.4	0.00	140.6
		1	6,691.84	6,658.53	0.50%	7,200.7	6,657.78	32.8
250	500	2	123.23	5.19	95.79%	7,201.7	4.99	292.4
		3	25,106.52	0.00	100.00%	7,200.7	0.00	328.2
		1	8,137.97	8,104.91	0.41%	7,200.2	8,104.36	88.2
300	600	2	30,427.93	2.67	99.99%	7,202.5	2.67	373.0
		3	30,427.42	0.00	100.00%	7,200.2	0.00	694.4
		1	35,498.66	0.00	100.00%	7,201.0	9,011.31	123.8
350	700	2	35,488.67	0.00	100.00%	7,200.2	2.63	641.7
		3	35,460.24	0.00	100.00%	7,204.4	0.00	1,241.5

Table 2: Computational results of the **WTA_PC** formulation.

results in Table 2. Linear programming relaxations of models with strengthening inequalities are significantly tighter compared to models without such constraints. This is especially true for problem instances with the parameter μ, the number of available weapons for every weapon type, being greater than one. As we expected, the relaxations become even tighter when the value of the parameter δ decreases to 0.0005.

Instance $\|\mathcal{W}\|$	$\|\mathcal{T}\|$	μ	δ	Objective	Lower Bound	Gap	CPU (sec.)	LP relaxation	CPU (sec.)
		1		4,970.36	4,969.87	⊛	796.3	4,965.98	27.3
200	400	2		115.92	106.57	8.06%	7,201.8	93.94	424.7
		3		19,800.00	0.00	100.00%	7,200.9	0.00	956.8
		1		6,689.83	6,689.83	⊛	1,757.8	6,686.15	59.0
250	500	2		114.63	103.05	10.10%	7,200.2	92.52	1,395.3
		3	0.001	25,235.00	0.00	100.00%	7,205.7	0.00	2,446.5
		1		8,136.88	8,135.38	0.02%	7,200.1	8,132.37	105.3
300	600	2		120.77	102.19	15.39%	7,200.3	93.04	1,923.1
		3		30,478.00	0.00	100.00%	7,201.1	0.00	3,760.6
		1		9,049.87	9,049.87	⊛	4,954.6	9,044.46	219.0
350	700	2		35,543.00	0.00	100.00%	7,201.4	93.19	6,426.9
		3		35,543.00	0.00	100.00%	7,201.4	0.00	5,830.3
		1		4,970.36	4,969.88	⊛	800.7	4,966.99	24.1
200	400	2		115.46	110.28	4.48%	7,200.1	98.94	453.0
		3		592.03	0.00	100.00%	7,390.5	0.00	1,230.4
		1		6,689.83	6,689.83	⊛	944.0	6,687.24	57.6
250	500	2		114.58	108.29	5.49%	7,201.1	99.74	986.3
		3	0.0005	25,235.00	0.00	100.00%	7,200.5	0.00	787.7
		1		8,136.88	8,136.07	⊛	1,532.3	8,133.83	107.4
300	600	2		120.73	110.03	8.86%	7,201.0	101.66	1,733.0
		3		30,478.00	0.00	100.00%	7,201.0	–	–
		1		9,049.82	9,048.68	0.01%	7,200.2	9,046.03	181.0
350	700	2		35,543.00	0.00	100.00%	7,201.4	102.22	3,153.1
		3		35,543.00	0.00	100.00%	7,201.4	–	–

Table 3: Computational results of the **WTA_PC** formulation with strengthening constraints (5.10) – (5.13).

Next, we investigate how the additional constraints affect the CPU time of solving the mixed integer linear programming formulations. Table 3 provides the results of experiments. First of all, notice that the improved formulation enables solving three problem instances to optimality. When optimality is not proven within the time limit, the optimality gap significantly decreases compared to the gaps of roughly $90\% - 100\%$ of instances reported in Table 2. Moreover, the optimality gap decreases even more when the parameter δ is set to 0.0005 instead of 0.001. Overall, we can say that problem instances with up to 300 weapon types and 600 targets, one weapon per each weapon type, can be solved within minutes of computer time. More challenging instances with two weapons available for each weapon type can now be solved with rather small optimality gaps of up to 9% within two hours of computer time. Still, most challenging problem instances, with $\mu = 3$, could not be solved to optimality and even no reasonable bound was obtained for instances with $|\mathcal{W}| = 200$ and $|\mathcal{T}| = 400$. Since a more accurate under-approximation requires more breakpoints, decreasing parameter δ can be done up to a point. The size of problem formulation with $|\mathcal{W}| = 350$, $|\mathcal{T}| = 700$ and $\mu = 3$ becomes so large that even the linear programming relaxation could not be solved within the time limit. For this reason, we do not experiment with smaller values of parameter δ for the probability chain-based linearization model.

7.3. Branch-and-Adjust algorithm

Table 4 presents results of the computational experiments with the same set of problem instances using the branch-and-adjust algorithm proposed in Section 6. Here we observe the

Instance $\|\mathcal{W}\|$	$\|\mathcal{T}\|$	μ	δ	Objective	Lower Bound	Gap	CPU (sec.)
		1		4,970.36	4,967.84	0.05%	7,200.0
200	400	2		115.47	112.03	2.98%	7,200.0
		3		11.81	7.56	36.00%	7,200.1
		1		6,689.83	6,686.75	0.05%	7,200.1
250	500	2		114.38	109.91	3.91%	7,200.1
		3	0.0005	12.43	7.92	36.26%	7,200.0
		1		8,136.90	8,133.21	0.05%	7,200.1
300	600	2		118.55	113.04	4.65%	7,200.1
		3		14.82	9.72	34.38%	7,200.0
		1		9,049.83	9,045.05	0.05%	7,200.3
350	700	2		121.06	114.74	5.22%	7,200.1
		3		12.07	6.76	44.00%	7,200.0
		1		4,970.36	4,969.87	⊛	86.3
200	400	2		115.45	114.86	0.51%	7,200.0
		3		10.63	9.68	8.98%	7,200.0
		1		6,689.83	6,689.16	⊛	392.2
250	500	2		114.35	113.53	0.72%	7,200.2
		3	0.0001	11.10	9.85	11.23%	7,200.2
		1		8,136.88	8,136.07	⊛	358.9
300	600	2		118.55	117.54	0.85%	7,200.1
		3		13.51	11.86	12.16%	7,200.1
		1		9,049.83	9,048.86	0.01%	7,200.1
350	700	2		121.24	119.71	1.27%	7,200.1
		3		11.36	9.54	16.03%	7,200.1

Table 4: Computational results of the **WTA_LA** formulation solved via the branch-and-adjust approach.

significantly improved quality of solutions obtained within the same time limit, both in terms of objective values and optimality gaps. Because of the compact size of the problem formulation involving piecewise linear approximations, we start experiments with the approximation quality parameter $\delta = 0.0005$. It is noticeable how the value of parameter δ plays a crucial role in the performance of the approach. The best results were obtained with $\delta = 0.0001$.

Given the strong results obtained even for the largest instances presented in Table 4, we generated even larger and thus more challenging problem instances to further evaluate the proposed algorithm that was executed with an even smaller value of the parameter δ equal to 0.00001. Table 5 presents results of experiments with old and new problem instances. Note how the algorithm execution results in very small gaps (not exceeding 2.02%) even for very large instances with up to 1,500 weapons of 500 weapon types and 1,000 targets.

Instance $\|\mathcal{W}\|$	$\|\mathcal{T}\|$	μ	δ	Objective	Lower Bound	Gap	CPU (sec.)
200	400	1	0.00001	4,970.67	4,970.17	⊛	419.9
		2		115.45	115.41	0.04%	7,200.2
		3		10.58	10.51	0.66%	7,200.1
250	500	1		6,689.96	6,689.29	⊛	541.1
		2		114.35	114.26	0.08%	7,200.2
		3		10.96	10.86	0.94%	7,200.1
300	600	1		8,136.95	8,136.48	⊛	1,859.5
		2		118.54	118.44	0.09%	7,200.2
		3		13.10	12.98	0.90%	7,200.1
350	700	1		9,049.87	9,049.52	⊛	2,414.4
		2		121.12	120.66	0.38%	7,200.2
		3		10.83	10.68	1.41%	7,200.1
400	800	1		11,491.08	11,489.94	⊛	1,844.9
		2		129.33	129.07	0.20%	7,200.2
		3		11.98	11.77	1.76%	7,200.2
450	900	1		12,521.67	12,520.43	⊛	4,736.4
		2		134.25	133.80	0.34%	7,200.3
		3		12.50	12.28	1.75%	7,200.2
500	1,000	1		13,141.45	13,139.04	0.02%	7,200.4
		2		124.36	123.19	0.94%	7,200.4
		3		12.08	11.83	2.02%	7,200.2

Table 5: Computational results of the **WTA_LA** formulation solved via the branch-and-adjust approach for large scale problem instances.

8. Conclusion

This paper considered the static version of the weapon-target assignment problem. Each weapon is assumed to destroy a target with a specified probability. Under the assumption that weapons perform independently of each other, the assignment problem seeks for an optimal allocation of weapons to targets that minimizes the expected value of survived targets. The optimal assignment problem is a nonlinear combinatorial optimization problem, which is computationally challenging to solve. The paper presented three approaches from the literature to linearize the problem and convert it to mixed integer linear optimization problems, for which efficient optimization solvers exist and can be directly employed. The first approach can only be used as an approximation since the nonlinear terms of the original problem are over-approximated by convex piecewise linear functions. The second approach linearizes the objective function of the WTA problem exactly, yet, it suffers from a weak linear programming relaxation, which negatively impacts the solution time of the assignment problem. The third linearization approach combines the other two in a more efficient way: it exactly linearizes the objective function of the WTA problem while strengthening the linear programming formulation using a set of convex piecewise linear under-approximation functions. The strengthening precision can be easily managed by changing a corresponding parameter. Finally, a specialized exact algorithm is proposed, which avoids the challenge of growing size of the corresponding LP models. The algorithm uses the compact and flexible piecewise linear under-approximation functions and adjusts the value of every incumbent solution obtained by the solver by its true value in terms of the original WTA objective function. Computational experiments demonstrate the efficiency of the considered solution methods. We observe that the improved probability chains-based linearization approach is efficient for solving problems to optimality with up to 350 weapon types with one weapon per each type. With a larger number of weapons available for each weapon type, the problems become challenging for solving to optimality. The proposed algorithm solves significantly larger problem instances to provable near optimality. The proposed branch-and-adjust algorithm for instances with up to $1,500$ weapons of 500 weapon types and $1,000$ targets finds solutions with corresponding optimality gaps of just slightly more than 2.0% within two hours of computer time.

References

Ahuja, R.K., Kumar, A., Jha, K.C., Orlin, J.B., 2007. Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem. Operations Research 55, 1136–1146.

Camm, J.D., Norman, S.K., Polasky, S., Solow, A.R., 2002. Nature Reserve Site Selection to Maximize Expected Species Covered. Operations Research 50, 946–955.

Cetin, E., Esen, S.T., 2006. A Weapon-Target Assignment Approach to Media Allocation. Applied Mathematics and Computation 175, 1266–1275.

Chang, S.c., James, R.M., Shaw, J.J., 1987. Assignment Algorithm for Kinetic Energy Weapons in Boost Phase Defence, in: Decision and Control, 1987. 26th IEEE Conference on, IEEE. pp. 1678–1683.

Day, R.H., 1966. Allocating Weapons to Target Complexes by Means of Nonlinear Programming. Operations Research 14, 992–1013.

Eckler, A.R., Burr, S.A., 1972. Mathematical Models of Target Coverage and Missile Allocation. Technical Report. MILITARY OPERATIONS RESEARCH SOCIETY ALEXANDRIA VA.

Esen, Ö., Çetin, E., Esen, S.T., 2008. A Mathematical Immunochemoradiotherapy Model: A Multiobjective Approach. Nonlinear Analysis: Real World Applications 9, 511–517.

Glover, F., 1975. Improved Linear Integer Programming Formulations of Nonlinear Integer Problems. Management Science 22, 455–460.

IBM, ILOG, 2020. CPLEX Optimization Studio User Manual.

denBroeder Jr, G., Ellison, R., Emerling, L., 1959. On Optimum Target Assignments. Operations Research 7, 322–326.

Kline, A., Ahner, D., Hill, R., 2019. The Weapon-Target Assignment Problem. Computers & Operations Research 105, 226–236.

Lu, Y., Chen, D.Z., 2021. A New Exact Algorithm for the Weapon-Target Assignment Problem. Omega 98, 102138.

Manne, A.S., 1958. A Target-Assignment Problem. Operations Research 6, 346–351.

Matlin, S., 1970. A Review of the Literature on the Missile-Allocation Problem. Operations Research 18, 334–373.

Metler, W.A., Preston, F.L., Hofmann, J., 1990. A Suite of Weapon Assignment Algorithms for a SDI Mid-Course Battle Manager. Technical Report. NAVAL RESEARCH LAB WASHINGTON DC.

Murphey, R.A., 2000. Target-Based Weapon Target Assignment Problems, in: Nonlinear Assignment Problems. Springer, pp. 39–53.

O'Hanley, J.R., Scaparra, M.P., García, S., 2013. Probability Chains: A General Linearization Technique for Modeling Reliability in Facility Location and Related Problems. European Journal of Operational Research 230, 63–75.

Sonuc, E., Sen, B., Bayir, S., 2017. A Parallel Simulated Annealing Algorithm for Weapon-Target Assignment Problem. International Journal of Advanced Computer Science and Applications 8, 87–92.

Wacholder, E., 1989. A Neural Network-Based Optimization Algorithm for the Static Weapon-Target Assignment Problem. ORSA Journal on Computing 1, 232–246.