Efficient Heuristic Approaches to the Weapon Target Assignment Problem

Azad M. Madni, Ph.D.*
Intelligent Systems Technology, Inc., Los Angeles, CA 90066

Mircea Andrecut, Ph.D.†
Institute for Biocomplexity and Informatics, University of Calgary, 2500 University Drive NW, Calgary, Alberta, T2N 1N4, Canada.

The Weapon Target Assignment (WTA) problem is a fundamental defense application of operations research. The problem consists of optimally assigning a given number of weapons to a set of targets so that the post-engagement total expected survival value of the targets is minimized. Since the WTA problem is known to be NP-complete, there are no exact methods to solve it. This paper presents two innovative heuristic algorithms based on Simulated Annealing and Threshold Accepting methods to solve the general WTA problem. Our computational results show that by using these algorithms, relatively large instances of the WTA problem can be solved near-optimally in a few seconds on a standard PC.

Introduction

The Weapon Target Assignment (WTA) consists of optimally assigning a finite number of weapons to a finite number of targets so that the total expected survival value of the targets after the engagement is minimized. Research on the WTA problem dates back to the 1950s and 1960s,1,2,3,4,5. A comprehensive review of the literature on the WTA problem is given in,4,6. Several exact algorithms have been proposed in the literature to solve two special cases of the WTA problem: (a) when all the weapons are identical,2 or (b) when the targets can be attacked by at most one weapon,7,8. A major result, obtained by Lloyd and Witsenhausen,9, is that the general WTA problem is NP-complete. This finding implies that the computation time of any optimal algorithm for this problem will grow exponentially with the size of the problem. Thus, there are no exact methods for the general WTA problem even for small problems. Some of the heuristic algorithms proposed to solve the WTA problem are based on linear integer programming,6,7, nonlinear network flow,6,7, neural networks,10 and genetic algorithms,11,12,13. However, in the absence of exact algorithms to solve the WTA problem, no estimates can be made on the quality of solutions produced by such heuristics.

* President and CEO, Intelligent Systems Technology Inc., 12122 Victoria Avenue, Los Angeles, CA 90066, Associate Fellow.
† Senior Research Associate, Institute for Biocomplexity and Informatics, University of Calgary, 2500 University Drive NW, Calgary, Alberta, T2N 1N4, Canada.
In this paper, we present two heuristic algorithms to solve the WTA problem. The first is based on the Simulated Annealing (SA) method and the second on the Threshold Accepting (TA) method. We selected these algorithms for study because of the need for scalability and speed for WTA problems. In particular, we were interested in the performance of these algorithms when the number of targets is greater than the number of weapons and when they are not. Our numerical results show that relatively large dimensionality WTA problems can be solved near-optimally by both these algorithms. In particular, the results show that SA offers slightly better solutions when the numbers of targets are strictly greater than the number of weapons. The solutions are equivalent when the number of targets are less than or equal to the number of weapons.

**Formulation of the WTA Problem**

At the outset, we briefly describe the WTA problem. This problem involves an offense that possesses a large but finite number of missiles that can be launched against the assets of the defense. The defense employs a finite number of interceptors (i.e., weapons) to destroy the incoming missiles (threats) and thereby protect its assets. In other words, the incoming missiles are the targets that the interceptors seek to destroy. These incoming missiles are assigned values by the defense. The interceptors (i.e., weapons) can engage at most one attacking missile launched by the offense. The engagement of a target by a weapon is modeled as a stochastic event, with a probability of kill assigned to each weapon-target pair (this is probability that the interceptor weapon will destroy the missile target if assigned to it). The engagement of a weapon-target pair is independent of all other weapons and targets. It is also assumed that a particular target may be engaged by more than one weapon. The objective is to assign weapons to targets so as to minimize the total expected value of the surviving targets. This is an integer optimization problem in that fractional weapon assignments are not allowed. Also, it is a large-scale problem because, in general, the number of weapons and targets is large, which makes enumeration techniques impractical.

The parameters and variables of the WTA problem are:

- $N$, the number of targets (offense).
- $M$, the number of weapons (defense).
- $v_n$, the value of targets, $n = 0, \ldots, N - 1$. 
• $P_{nm}$, the probability of kill (the probability that weapon $m$ destroys target $n$ if it is assigned to it) 
$n = 0, \ldots, N - 1$, $m = 0, \ldots, M - 1$.

• The decision variable is the $N \times M$ matrix $x = [x_{nm}]$, where:

$$x_{nm} = \begin{cases} 1 & \text{if weapon } n \text{ is assigned to target } m \\ 0 & \text{otherwise} \end{cases}.$$  \hspace{1cm} (1)

The probability that target $n$ is not destroyed by weapon $m$ is $(1 - P_{nm})^{x_{nm}}$. Since the engagement of a weapon-target pair is independent of all other weapons-targets pairs, the probability that target $m$ survives after all weapons have been fired is given by:

$$f_n(x) = \prod_{m=0}^{M-1} (1 - P_{nm})^{x_{nm}}.$$  \hspace{1cm} (2)

Thus, the total expected value of the surviving targets is given by:

$$F(x) = \sum_{n=0}^{N-1} v_n f_n(x).$$  \hspace{1cm} (3)

Therefore, the WTA problem can be stated as following:

$$\min_x F(x) \qquad \text{subject to} \quad \sum_{n=0}^{N-1} x_{nm} = 1, \ m = 0, \ldots, M - 1.$$  \hspace{1cm} (4)

The constraints are due to the fact that each weapon must be assigned to exactly one target. Also, it can easily be shown that the objective function $F(x) : \{0,1\}^{NM} \rightarrow R$ is convex,\cite{17}.

**Maximum Marginal Return for the Related WTA Problem**

It is interesting to note that the WTA problem can be solved exactly by assuming a single class of weapons\cite{18}.

For this case, the kill probabilities are independent of the weapons, i.e., $P_{nm} = P_n$. This assumption is valid if the defense has only one type of weapons and all the weapons are located in the same area, such that the geometry and time of intercept are the same for all weapon-target pairs. For this particular case, the Maximum Marginal Return (MMR) is a polynomial time algorithm, guaranteed to find the optimal solution\cite{2}.
The MMR algorithm can be also applied to a relaxed version of the WTA problem to find a lower bound for the general WTA problem. Replacing \( p_{nm} \) with \( q_n = \max_m p_{nm} \) we can formulate the relaxed WTA (R-WTA) problem as follows:

\[
\min \sum_{n=0}^{N-1} v_n (1 - q_n)^{x_n} \quad \text{subject to} \quad \sum_{n=0}^{N-1} x_n = M, \tag{5}
\]

where \( x_n \) is the number of weapons assigned to target \( n \). Note that now the objective function is convex and separable, which make the problem tractable for exact MMR algorithm. MMR is a greedy algorithm, in which the weapons are assigned sequentially to the target for which the reduction in the objective cost is maximum. The algorithm terminates when all weapons are assigned. The pseudo-code of the MMR algorithm applied to the R-WTA problem is given in Figure 1. The algorithm begins by initializing \( q_n = \max_m p_{nm} \), \( \sigma_n = v_n q_n \) and \( x_n = 0 \), for all \( n = 0, ..., N-1 \). The marginal return of adding an additional weapon to the target \( k \) is given by:

\[
\sigma_k = v_k [(1 - q_k)^{x_k} - (1 - q_k)^{x_k+1}] = v_k q_k (1 - q_k)^{x_k}. \tag{6}
\]

Since initially \( x_n = 0 \) for all \( n = 0, ..., N-1 \), the initial marginal return is:

\[
\sigma_n = v_n q_n. \tag{7}
\]

The MMR is a greedy algorithm in that the weapons are always assigned to the target \( k \) with the maximal marginal return:

\[
k = \arg \max_n (\sigma_n). \tag{8}
\]

This algorithm is repeated until all weapons have been assigned to targets.
Simulated Annealing Method

In this section, we present a sub-optimal solution to the general WTA problem that employs Simulated Annealing (SA), a heuristic method developed in statistical physics. In general, this method has produced extremely good solutions for hard optimization problem instances encountered in statistical physics (e.g., traveling salesmen problem and spin glasses).

SA is a generic, probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space,\textsuperscript{20,21,22,23,24}. Each step of the SA algorithm replaces the current solution by a random solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter $\theta$ (called the temperature), which gradually decreases during the process. This dependency is such that the current solution changes almost randomly when $\theta$ is large, but trends increasingly downhill as $\theta$ goes to zero. The allowance for uphill moves saves the method from getting “stuck” in local minima, the bane of greedier methods. To apply the SA method to a specific problem, one must specify: (i) the state space; (ii) the candidate solution selection method (which enumerates the candidates for the next state); (iii) the annealing schedule; and (iv) the probability transition function. It is important to realize that these choices can have a significant impact on the method’s effectiveness.

initialize $p, v$;
for $n = 0 : N - 1$ do
   
   $q_n \leftarrow \max_m \{p_{nm}\}$;
   $\sigma_n \leftarrow v_n q_n$;
   $x_n \leftarrow 0$;

for $m = 0 : M - 1$ do
   
   $k = \arg\max_n (\sigma_n)$;
   $x_k \leftarrow x_k + 1$;
   $\sigma_k \leftarrow v_k q_k (1 - q_k)^{\frac{1}{\theta}}$;

return $x$;

Fig. 1 The NMR algorithm formulated for the R-WTA problem.
In the case of the general WTA problem, the state space corresponds to \( x = [x_{nm}] \in \{0,1\}^{NM} \). We begin the computation with a feasible solution:

\[
x_{nm} = \delta(n, m \mod(N)),
\]

where

\[
\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}
\]

is the Dirac function, and \( x \mod(y) \) is the modulo operation. This way every weapon is assigned to exactly one target, which is the requirement of the constraint in the general WTA problem (4). We also consider an integer variable: \( \xi_m = m \mod(N), \quad m = 0,\ldots,M - 1 \), such that we have:

\[
x_{nm} = \delta(n, m \mod(N)) = \delta(n, \xi_m).
\]

One can see that \( \xi_m \) gives the target index, to which the weapon \( m \) is assigned.

Let us denote by \( x \) the solution corresponding to the current step of the algorithm. A candidate solution \( x' \) is found by randomly selecting a weapon \( m \) and randomly choosing another target \( i \) for this weapon. The return involved in this transition is:

\[
\Delta(x, x') = F(x') - F(x).
\]

The transition \( x \leftarrow x' \) is accepted with a probability given by the probability transition function discussed next.

The initial temperature must be large enough to make the uphill and downhill transition probabilities about the same. The temperature must then decrease so that it is nearly zero, at the end of the allotted time. A popular choice is the exponential schedule, where the temperature decreases by a fixed factor \( \lambda \) at each iteration. If \( \theta_i \) and \( \theta_f \) are the initial and, respectively, the final temperature then

\[
\lambda = (\theta_f / \theta_i)^{1/T},
\]

where \( T \) is the final number of iterations. For each iteration \( t = 0,\ldots,T - 1 \), one performs a fixed number of transition steps \( s = 0,\ldots,S - 1 \), such that the equilibrium is reached at that temperature.

The probability of making the transition from the current state \( x \) to the candidate new state \( x' \) is given by the Boltzmann distribution:
\[ P(\Delta(x, x'), \theta) = \begin{cases} 1 & \text{if } \Delta(x, x') \leq 0 \\ \exp(-\Delta(x, x')/\theta) & \text{if } \Delta(x, x') > 0 \end{cases} \] (13)

Let us analyze the return \( \Delta(x, x') \) in more detail. The initial total expected value of the surviving targets is given by:

\[
F(x) = \sum_{n=0}^{N-1} v_n \prod_{k=0}^{M-1} (1 - p_{nk})^{x_{nk}} \\
+ v_{\bar{x}_n} \prod_{k=0}^{M-1} (1 - p_{\bar{x}_nk})^{x_{\bar{x}_nk}} \\
+ v_{i} \prod_{k=0}^{M-1} (1 - p_{ik})^{x_{i}}
\] (14)

The final total expected value of the surviving targets is given by:

\[
F(x') = \sum_{n=0}^{N-1} v_n \prod_{k=0}^{M-1} (1 - p_{nk})^{x'_{nk}} \\
+ v_{\bar{x}_n} \prod_{k=0}^{M-1} (1 - p_{\bar{x}_nk})^{x'_{\bar{x}_nk}} (1 - p_{\bar{x}_nk})^{x_{\bar{x}_nm} - x_{\bar{x}_nm}} \\
+ v_{i} \prod_{k=0}^{M-1} (1 - p_{ik})^{x'_{i}} (1 - p_{im})^{x_{im} - x_{im}}
\] (15)

Therefore we have:

\[
\Delta(x, x') = v_{\bar{x}_n} [(1 - p_{\bar{x}_nm})^{x_{nm} - x_{\bar{x}_nm}} - 1] \prod_{k=0}^{M-1} (1 - p_{\bar{x}_nk})^{x_{\bar{x}_nk}} + v_{i} [(1 - p_{im})^{x_{im} - x_{im}} - 1] \prod_{k=0}^{M-1} (1 - p_{ik})^{x_{i}}
\] (16)

This expression is more efficient to calculate because it only uses the two involved targets, \( \bar{x}_m \) and \( i \). Thus, if the transition is successful, then the target for weapon \( m \) is updated as \( \bar{x}_m \leftarrow i \). Also, the values of the involved \( x \) elements are exchanged: \( \alpha \leftarrow x_{i,m}, \ x_{i,m} \leftarrow x_{\bar{x}_m}, x_{\bar{x}_m} \leftarrow \alpha \). The pseudo-code of our SA method is given in Figure 2.
Threshold Accepting Method

Threshold Accepting (TA) method is a deterministic analog of the SA method, \(^{25,26}\). It is a refined search procedure which escapes local minima by accepting solutions which are not worse by more than a given threshold. The algorithm is deterministic in the sense that the number of iterations are fixed and the neighborhood is explored with a fixed number of steps during each iteration. Analogous to the SA method, the threshold is decreased successively until it reaches the value of zero in the last iteration. The main difference between SA and TA is that for SA the threshold is modeled as a random variable, while for TA the threshold is deterministic. The TA algorithm has three advantages: (1) it is easy to parameterize; (2) it is robust to changes in problem characteristics; and (3) it works well for many hard problem instances. An extensive introduction to the TA method is given by Winker, \(^{26}\).

For the case of the WTA problem, we consider that the threshold corresponds to the temperature defined for the SA
method. Obviously, this choice satisfies the requirements since $\theta \rightarrow \theta_f \approx 0$. The pseudo-code of our TA method is given in Figure 3.

```plaintext
initialize $\theta_i$, $\theta_j$, $T$, $p$, $v$;
$\theta \leftarrow \theta_i$; $\lambda \leftarrow (\theta_j / \theta_i)^{1/T}$;
for $n = 0: N-1$ do
  { for $m = 0: M-1$ do
    { $\xi_m = m \text{ mod}(N)$; $x_m = \delta(n, \xi_m)$; }
  }
for $t = 0: T-1$ do
  { for $s = 0: S-1$ do
    { $m \leftarrow \text{rand}(M)$; $i \leftarrow \text{rand}(N)$; $\Delta \leftarrow \Delta(x, x')$;
      if $\Delta \leq \theta$ then
        { $\alpha \leftarrow x_{i,m}$; $x_{i,m} \leftarrow x_{\xi_m,m}$
          $x_{\xi_m,m} \leftarrow \alpha$; $\xi_m \leftarrow i$;
        }
    $\theta \leftarrow \lambda \theta$;
    }
return $x$.
```

Fig. 3 The TA algorithm formulated for the WTA problem.

Experimentation and Numerical Results

R-WTA Case

In the first experiment, we considered the R-WTA problem. The R-WTA instances were obtained by first generating $V_n$ and $Q_n$ as uniform distributed random variables on the interval $[0, 1]$, and then setting $p_{nm} = q_n$, for all $n = 0, ..., N-1$ and $m = 0, ..., M-1$.

The goal of this experiment was to compare the MMR, SA and TA methods. Since the MMR algorithm gives an exact solution to the R-WTA problem, we were interested in determining whether or not the results calculated with the SA and TA methods converged to the optimum assignment found by the MMR algorithm. To this end, we carried out extensive simulations for $25 \leq N, M \leq 200$ using the following parameters: $T = 100$, $S = 100$, $\theta_f = 1$, $\theta_i = 10^{-5}$. It is important to note that the SA and TA results converged to the optimum assignment.
(calculated with the MMR method) for all considered R-WTA instances. In Figure 4, we present three such examples of the results from numerical simulations.

![Graphs](image)

Fig. 4 Simulation results for the R-WTA problem: (a) The annealing schedule used for the SA and TA algorithms; (b) \( N = 100, M = 50 \); (c) \( N = 50, M = 100 \); (d) \( N = 100, M = 100 \); (SA = Red, TA = Blue, MMR=Black (optimum assignment)).

One can see from Figure 4 that in about 50 steps, both the SA and TA algorithms converge to the optimum assignment found by the MMR algorithm (the black base line on the figures). As a measure of convergence, we considered the time dependence of \( F \), which is the total expected value of the surviving targets at time step \( t \). These results show that the SA and TA methods are accurate and appropriate for moderate size instances of R-WTA problems. Obviously, the MMR method is faster for large instances \( N, M > 200 \), since its complexity is only \( O(N + M \log N) \).

WTA Case

Since the MMR algorithm cannot be applied to the more general WTA instances, we are only interested in comparing the results provided by the SA and TA algorithms. For the same reason, we also consider an extreme case of the TA algorithm, where the threshold is kept constant all the time at \( \theta = 0 \). We denote this algorithm as TA(0).

As before, the target values \( v_n \) and the probabilities \( p_{nm} \) were generated by uniform random variables on the interval \([0,1]\) Representative examples of the simulations are presented in Figure 5.
Fig. 5 Simulation results for the WTA problem: (a) The annealing schedule used for the SA and TA algorithms; (b) \( N = 100, M = 50 \); (c) \( N = 50, M = 100 \); (d) \( N = 100, M = 100 \); \( \text{SA} = \text{Red}, \text{TA} = \text{Blue}, \text{TA}(0) = \text{Black} \).

It is interesting to note that the greedy \( \text{TA}(0) \) algorithm converges very rapidly to a pretty good local minimum. In contrast, SA and TA algorithms converge more slowly; however, their solution is superior to the one calculated by \( \text{TA}(0) \). As such, \( \text{TA}(0) \) can be regarded as a fast alternative to SA and TA, when there is high demand for speed. Nevertheless, for the investigated range, \( 25 \leq N, M \leq 200 \), both the SA and TA methods can solve a general WTA instance in only a few seconds on a standard PC. For the instance solved in Figure 4, one can see that SA gives a slightly better solution than TA. To elucidate which method is better, we averaged the results over 100 general WTA instances. The results obtained are presented in Figure 6. One can see that indeed SA offers a slightly better solution when \( N > M \), however SA and TA solutions become equivalent for \( N \leq M \).
Fig. 6 Simulation results averaged over 100 general WTA instances: (a) The annealing schedule used for the SA and TA algorithms; (b) $N = 100$, $M = 50$; (c) $N = 50$, $M = 100$; (d) $N = 100$, $M = 100$; SA = Red, TA = Blue, TA(0) = Black.

Conclusions

The general WTA problem is known to be NP-complete and, therefore, there are no exact methods to solve it. In this paper, we presented two heuristic algorithms to solve the WTA problem. These algorithms are based on the Simulated Annealing and the Threshold Accepting methods, which were originally developed for solving hard combinatorial optimization problems frequently encountered in statistical physics. Our numerical results have shown that relatively large size instances ($25 \leq N, M \leq 200$) of the general WTA problem can be solved near-optimally by these algorithms, in only a few seconds on a standard PC. Furthermore, these algorithms have the added advantage that they can be easily parallelized. When such parallelization is possible, then parallel computing is the preferred form of competition. (The specific hardware architecture used to support parallelism could be multi-core or multi-processor computers having multiple processing elements within a single machine). In other words, the processing elements can be diverse and include resources such as a single computer with multiple processors, several networked computers, specialized hardware, or any combination of these. Assuming that there are $K$ processors available, one can run in parallel the same WTA instance on each processor $k = 1, \ldots, K$. The best solution is given by $x^* = \min\{x^{(1)}, \ldots, x^{(k)}\}$, where $x^{(k)}$ is the local minimum of the results computed by each processor. Thus, the performance of these methods increases by simply adding more processors. Furthermore, the
complexity of the algorithm is quite small, requiring only successive calculations of $\Delta(x, x')$ for a fixed number of time steps $T$. This reduced complexity, coupled with large scale parallelization provide extremely good solutions for large instances of the WTA problem.

References


