

Literature Review of Weapon Target Assignment problem

Guanda Li

Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and Systems Science,
Chinese Academy of Sciences

December 20, 2022

Contents

- 1 Introduction
- 2 Formulations and algorithms
 - Static Weapon Target Assignment problem
 - Dynamic Weapon Target Assignment problem
 - Recent Developments
- 3 Outer Approximation
- 4 Columnn Generation
- 5 Future Research Plan

Introduction

Background

- **Weapon Target Assignment problem** is a problem in the military field
- The basic problem is to consider using m weapons to attack n targets, in order to minimize the weighted survival probability of all targets.
- **The problem is proved to be NP-complete**, and is often formulated as a nonlinear model.
- Before 2015, it took 16.2 hours to compute on a problem with 80 weapons and 80 targets.

History

- Give the basic formulation of WTA Problem (1958)
- linear approximations to problems, solve simplified forms of problems, or small-scale problems.
- Give the dynamic WTA Problem (1985).
- Improve the model of static and dynamic WTA problems and use heuristic algorithms to solve them.
- equivalent linear formulations of the WTA problem and solve it exactly.

Basic formulation

- $I = \{1, \dots, m\}$, Weapon set.
- $J = \{1, \dots, n\}$, Target set.
- $p_{ij} \in [0, 1]$, probability that i hits j .
- V_j , The weight of the target j .
- x_{ij} , whether weapon i attack j .

$$\begin{aligned}
 \max \quad & \sum_{j=1}^n (1 - V_j) \left(1 - \prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right) \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S0}$$

Basic formulation

- $I = \{1, \dots, m\}$, Weapon set.
- $J = \{1, \dots, n\}$, Target set.
- $p_{ij} \in [0, 1]$, probability that i hits j .
- V_j , The weight of the target j .
- x_{ij} , whether weapon i attack j .

$$\min \sum_{j=1}^n V_j \left(\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right) \quad (1)$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \quad (2)$$

$$x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I. \quad (3)$$

Formulations and Algorithms

Category

- Static WTA problems.
- Dynamic WTA problems
- In recent years, some new problem forms have also be propose. such as adding more combat elements or using multi-objective programs.

Notation

- I , weapon set.
- J , target set.
- n , the number of targets.
- m , the number of weapon types. If limiting to one of each type, this variable indicates the number of weapons.
- w_i , the number of class i weapon
- p_{ij} , The probability that weapon i hits target j .
- q_{ij} , The probability that weapon i misses target j .
- x_{ij} , The number of weapons of type i assigned to target j . If each weapon is limited to only one, this variable is a 0-1 variable.
- V_j , value of target j .
- c_{ij} , The cost of assigning weapon i to target j

static WTA model 1

- Compared to the basic model, allows w_i weapons for each weapon type.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j \left(\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right) \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq w_i \quad \forall i \in I, \\
 & x_{ij} \in \mathbb{Z}^+ \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S1.1}$$

static WTA model 1

- Transform $1 - p_{ij}$ to q_{ij} to make the formulation simpler.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j \left(\prod_{i=1}^m q_j^{x_{ij}} \right) \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq w_i \quad \forall i \in I, \\
 & x_{ij} \in \mathbb{Z}^+ \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S1.2}$$

static WTA model 2

- For Convenience, Assume that any weapon has the same probability of hitting the same target.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j q_j^{\sum_{i=1}^m x_{ij}} \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq w_i \quad \forall i \in I, \\
 & x_{ij} \in \mathbb{Z}^+ \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S2}$$

static WTA model 3

- Logarithm of the objective function in S1

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j \exp \left(\sum_{i=1}^m x_{ij} \ln(1 - p_{ij}) \right) \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq w_i \quad \forall i \in I, \\
 & x_{ij} \in \mathbb{Z}^+ \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S3.1}$$

static WTA model 3

- replaced the objective function with y_j to get the following model.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j e^{y_j} \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq w_i \quad \forall i \in I, \\
 & \sum_{i=1}^m \ln(1 - p_{ij}) x_{ij} = y_j \quad \forall j \in J \\
 & x_{ij} \in \mathbb{Z}^+ \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S3.2}$$

- The model is equivalent to model S1, but it is more convenient for the solver to calculate.
- [Kline et al.\(2017b\)](#) points out that use the commercial solver BARON to solve the problem, this form can increase the correct rate by 21%

static WTA model 4

- limit the x_{ij} to binary variable.
- For $(1 - p_{ij}x_{ij}) = (1 - p_{ij})^{x_{ij}}$, $x \in \{0, 1\}$, The problem can be transformed into the following form.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j \left(\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right) \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S4}$$

- This conversion will cause the relaxation of the problem from a convex form to a non-convex form.

static WTA model 5

- Transform the problem into a knapsack problem

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n \sum_{x=1}^m c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \\
 & \sum_{i=1}^m x_{ij} \leq 1 \quad \forall j \in J \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S5}$$

- This method removes the coupling when multiple weapons strike the same weapon, transforms the difficult objective function into a linear function.

static WTA model 6

- This is attacking model.
- there are K assets we have to protect, and n targets try to destroy these assets.
- The probability with target j will destroy k is γ_{jk} .

$$\begin{aligned}
 \min \quad & \sum_{k=1}^K K a_k \prod_{j=1}^{n_k} \left[\gamma_{ij} \prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right] \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \\
 & \sum_{k=1}^K n_k = n \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S6}$$

dynamic WTA model 1 : shoot-look-shoot

- the model assumes that it is possible to observe the target situationn after a strike and rearrange weapons on targets that have not yet been destroyed
- In this model, no new targets will appear in the new time stage.
- Repeatedly attack the targets until all targets are destroyed or the number of iterations reaches the upper limit

dynamic WTA model 2 : multi-stages

- The model assumes that targets emergence will occur in different stages
- the given number of targets is engaged in first stage, the new targets that appear in later stages are given in the form of probability distributions
- Due to the high complexity of the model, heuristic solutions are mainly used.

Sensor WTA problems

- Considering the real scene, it is necessary to introduce sensors to identify targets to be attacked, and similarly, elements such as command posts also need to be introduced.
- The following is a simpler model: S represents the set of sensors, and o represents the number of sensors

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n V_j \left(\prod_{i=1}^m \prod_{s=1}^o (1 - p_{isj})^{x_{isj}} \right) & (\text{Sensor}) \\
 \text{s.t.} \quad & \sum_{j=1}^n \sum_{s=1}^o x_{isj} \leq 1 \quad \forall i \in I, \\
 & \sum_{j=1}^n \sum_{i=1}^m x_{isj} \leq 1 \quad \forall s \in S, \\
 & x_{isj} \in \{0, 1\} \quad \forall j \in J, i \in I, s \in S.
 \end{aligned}$$

- The two constraints are each weapon and each sensor can only be assigned to one target.

Multi-objective programs

- In the real scene, not only the probability of destroying the target needs to be considered
- Introduced objectives include minimizing cost or minimizing time
- minimizing cost: Using weapon i on target j requires a fixed cost
- minimizing time: minimize combat time in order to achieve combat objectives
- combine multiple goals or Simultaneously optimize multiple objectives.

Outer Approximation

basic formulation

- $I = \{1, \dots, m\}$, Weapon set.
- $J = \{1, \dots, n\}$, Target set.
- $p_{ij} \in [0, 1]$, probability that i hits j .
- V_j , The weight of the target j .
- x_{ij} , whether weapon i attack j .

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n a_j \left(\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right) \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I.
 \end{aligned} \tag{S0}$$

formulation

$$\prod_{i=1}^m (1 - p_{ij}x_{ij}) \iff \prod_{i=1}^m (1 - p_{ij})^{x_{ij}}, x \in \{0, 1\}$$

- When restricting x to be a binary variable, the objective function values are the same.

The convexity of the objective function

consider the second formulation

$$f_j(x) = \prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \quad \forall j \in J$$

Hessian matrix H :

$$\frac{\partial f_j(x)}{\partial x_{aj} \partial x_{bj}} = \ln(1 - p_{aj}) \ln(1 - p_{bj}) f_j(x)$$

$$H = f(x) \begin{bmatrix} \ln(1 - p_{1j}) \ln(1 - p_{1j}) & \ln(1 - p_{1j}) \ln(1 - p_{2j}) & \cdots & \ln(1 - p_{1j}) \ln(1 - p_{mj}) \\ \ln(1 - p_{2j}) \ln(1 - p_{1j}) & \ln(1 - p_{2j}) \ln(1 - p_{2j}) & \cdots & \ln(1 - p_{2j}) \ln(1 - p_{mj}) \\ \vdots & \vdots & \ddots & \vdots \\ \ln(1 - p_{mj}) \ln(1 - p_{1j}) & \ln(1 - p_{mj}) \ln(1 - p_{2j}) & \cdots & \ln(1 - p_{mj}) \ln(1 - p_{mj}) \end{bmatrix}$$

The convexity of the objective function

we assume that:

$$l = [\ln(1 - p_{1j}) \quad \ln(1 - p_{2j}) \quad \cdots \quad \ln(1 - p_{mj})]$$

the Hessian matrix is :

$$H = f(x)l \cdot l^T$$

It can be seen that the matrix is a rank-one matrix, that is, the columns are linearly related, so the determinant is 0 so the original function is convex.

transformed model

objective function

$$\sum_{j=1}^n a_j \left(\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right)$$

by introducing auxiliary variables, the nonlinear term can be transformed from the objective function into the constraint.

if we take η_j as the auxiliary variables to $\prod_{i=1}^m (1 - p_{ij})^{x_{ij}}$ the model can be transformed to

$$\min \quad \sum_{j=1}^n a_j \eta_j \quad (S0')$$

$$\text{s.t.} \quad \eta_j \geq \prod_{i=1}^m (1 - p_{ij})^{x_{ij}}, \quad \forall j \in J$$

$$\sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I,$$

$$x_{ij} \in \{0, 1\} \quad \forall j \in J \quad i \in I.$$

Basic idea of outer approximation

If $f(x)$ is a convex function, then for any point x^* in the feasible region, we have

$$f(x) \geq f(x^*) + \nabla f(x^*)(x - x^*)$$

Therefore, if the constraint of the original problem is $\eta \geq f(x)$, then

$$\eta \geq f(x) \geq f(x^*) + \nabla f(x^*)(x - x^*)$$

must be correct

- That is, for any given point, a linear constraint can be introduced to ensure that the feasible region satisfies the constraint.
- The outer approximation method hopes to take advantage of the integer characteristics and replace a nonlinear constraint by introducing multiple linear constraints, but guarantees that the problems before and after the transformation are equivalent.

constraint formulation of outer approximation

take $f_j(x) = \prod_{i=1}^m (1 - p_{ij})^{x_{ij}}$ $j \in J$, then for any given \bar{x} in feasible domain, we have

$$\begin{aligned}\nabla f(\bar{x})(x - \bar{x}) &= f(\bar{x}) \sum_{i=1}^m \ln(1 - p_{ij})(x_{ij} - \bar{x}_{ij}) \\ &= f(\bar{x}) \sum_{i=1}^m \ln(1 - p_{ij})x_{ij} - f(\bar{x}) \sum_{i=1}^m \ln(1 - p_{ij})\bar{x}_{ij}\end{aligned}$$

- The first term is the linear combination of variables, and the second term is a constant when \bar{x} is given.
- We denote this constraint as the outer approximation constraint.

outer approximation model

Let X donates the set of all integer feasible solutions in the problem, the model can be described as.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n a_j \eta_j \\
 \text{s.t.} \quad & \eta_j \geq f(\bar{x}) \sum_{i=1}^m \ln(1 - p_{ij})(x_{ij} - \bar{x}_{ij}) + f(\bar{x}), \quad \forall j \in J, \bar{x} \in X \\
 & \sum_{i=1}^m x_{ij} \leq 1 \quad \forall j \in J, \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in J \quad i \in I.
 \end{aligned} \tag{OA}$$

the number of outer approximation constraints is very large, which has caused certain difficulties in solving the problem. Therefore, the restricted model is considered, that is, only some constraints are considered.

Idea of outer approximation method

There are too many outer approximation constraints, only a subset of them is considered : $\hat{X} \subseteq X$

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n a_j \eta_j \\
 \text{s.t.} \quad & \eta_j \geq f(\bar{x}) \sum_{i=1}^m \ln(1 - p_{ij})(x_{ij} - \bar{x}_{ij}) + f(\bar{x}), \quad j \in J, \bar{x} \in X \\
 & \sum_{i=1}^m x_{ij} \leq 1 \quad j \in J, \quad x_{ij} \in \{0, 1\} \quad j \in J \quad i \in I
 \end{aligned}$$

- ❶ Remove all outer approximation constraints and give the current optimal solution x, η
- ❷ Check whether the current optimal solution can satisfy all the outer approximation constraints. if true, the iteration terminates and end the execution.
- ❸ Otherwise, choose to outer approximation constraint that has been violate most and add it to the restricted model, resolve the model and go to step 2.

choose violate constraint

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n a_j \eta_j \\
 \text{s.t.} \quad & \eta_j \geq f(\bar{x}) \sum_{i=1}^m \ln(1 - p_{ij})(x_{ij} - x_{ij}^-) + f(\bar{x}), \quad j \in J, \bar{x} \in X \\
 & \sum_{i=1}^m x_{ij} \leq 1 \quad j \in J, \quad x_{ij} \in \{0, 1\} \quad j \in J \quad i \in I
 \end{aligned}$$

- Unless an optimal solution has been found, the solution to the restricted problem must bring a violation of the outer approximation constraint.
- It can be intuitively understood that a violation means that η_j is outside the feasible region.
- When obtaining a solution that x is in the feasible region, if $\eta < f_j(x)$, this causes infeasible. the infeasible point can be excluded by adding outer approximation constraint.

Columnn Generation

Basic Idea

- Some linear programming problems have too many columns (variables), making it difficult to solve
- Use only some of the variables at the beginning of the algorithm and assume all the other variables are 0
- Variables that have the potential to improve the objective function are iteratively added to the model.
- Once it can be proven that adding new variables will no longer improve the value of the objective function, the iterative process is terminated and an optimal solution is obtained.

How to use column generation in WTA Problem

- **Basic Idea:** By listing all the weapon assignment scenarios S , the problem is transformed into a linear programming problem.
- Assuming that there are m weapons, for any target j , each weapon can choose to attack j or not attack j , so there are 2^m different attack schemes.
- **an example:** There are a total of 8 weapons, and the attack plan using No. 1, 3, and 6 weapons is recorded as $s_{[1,0,1,0,0,1,0,0]}$, $|S| = 2^8 = 256$
- n_{si} : binary variable, indicates whether to enable weapon i in the s th scene. In the above example, $n_{s1} = 1$, $n_{s2} = 0$
- $q_{js} = a_j \prod_{i=1}^m (1 - n_{si} \cdot p_{ij})$: weighted probability of the plan s to hit the target j , For example, q_{3s} is to use the No. 1, 3, and 6 weapons to hit the target 3 and multiply the probability of destroying the target j by the weight of the target j .

transformed formulation

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} q_{js} y_{js} & (CG) \\
 \text{s.t.} \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} n_{si} y_{js} \leq 1 & \forall i \in I \\
 & \sum_{s=0}^{2^m-1} y_{js} = 1 & \forall j \in J \\
 & y_{js} \in \{0, 1\} & \forall j \in J, s \in S
 \end{aligned}$$

- $I = \{1, \dots, m\}$, weapon set
- $J = \{1, \dots, n\}$, target set
- $S = \{1, \dots, 2^m\}$, scene set
- n_{si} : weather scene s use weapon i
- y_{js} : Whether use s to attack target j
- q_{js} : weighted destruction probability of using s to j

transformed formulation continuous

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} q_{js} y_{js} & (CG) \\
 \text{s.t.} \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} n_{si} y_{js} \leq 1 & \forall i \in I \\
 & \sum_{s=0}^{2^m-1} y_{js} = 1 & \forall j \in J \\
 & y_{js} \in \{0, 1\} & \forall j \in J, s \in S
 \end{aligned}$$

- **objective function**: minimize the weighted destruction probabilities.
- **first constraint**: Each weapon can only attack one target.
- **second constraint**: assign exactly one scene for each target

column enumeration

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} q_{js} y_{js} && \text{(CG)} \\
 \text{s.t.} \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} n_{si} y_{js} \leq 1 && \forall i \in I \\
 & \sum_{s=0}^{2^m-1} y_{js} = 1 && \forall j \in J \\
 & y_{js} \in \{0, 1\} && \forall j \in J, s \in S
 \end{aligned}$$

- This idea is from [Lu\(2021\)](#)
- The basic idea of the column enumeration method is to enumerate all columns in a smarter way
- Two techniques are used in the article: weapon number bounding and weapon domination
- weapon number bounding: Scenarios with too few or too many weapons are give no improvement to the objective function.

LP Relaxation

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} q_{js} y_{js} && \text{(CG-LP)} \\
 \text{s.t.} \quad & \sum_{j=1}^n \sum_{s=0}^{2^m-1} n_{si} y_{js} \leq 1 && \forall i \in I \\
 & \sum_{s=0}^{2^m-1} y_{js} = 1 && \forall j \in J \\
 & y_{js} \geq 0 && \forall j \in J, s \in S
 \end{aligned}$$

- LP Relaxation, the third constraint can be changed into $y_{js} \geq 0$
- linear program contains $n \times 2^m$ columns
- Only some of the columns (variables) are taken, because in the optimal solution of linear programming, at most $m + n$ variables are not 0, that is, the dual constraints corresponding to other variables are not active.

Dual Problem

$$\begin{aligned}
 \max \quad & \sum_{i=1}^m u_i + \sum_{j=1}^n v_j && \text{(CG-Dual)} \\
 \text{s.t.} \quad & \sum_{i=1}^m x_{is} u_i + v_j \leq q_{js} \quad \forall (s, j) \in X \\
 & u_i \leq 0, \quad v_j \text{ free}
 \end{aligned}$$

- If $X = \{(s, j) | s \in S, j \in J\}$, each element in X corresponds to a constraint.
-
- Selecting some columns in the original problem is equivalent to selecting some rows in the dual problem.

Restricted Dual Problem

$$\begin{aligned}
 \max \quad & \sum_{i=1}^m u_i + \sum_{j=1}^n v_j && \text{(CG-Dual-R)} \\
 \text{s.t.} \quad & \sum_{i=1}^m x_{is} u_i + v_j \leq q_{js} \quad \forall (s, j) \in \hat{X} \\
 & u_i \leq 0, \quad v_j \text{ free}
 \end{aligned}$$

- Only select some constraints, that is, only consider the constraints generated by the some (s, j) in $\hat{X} \subseteq X$
- Solve to get u^*, v^* and bring into the original dual, if all the constraints are satisfied, it must be the optimal solution.
- Otherwise, choose the constraint that violates the most, that is, the largest $\sum_{i=1}^m x_{is} u_i + v_j > q_{js}$.

subproblem

$$\begin{aligned}
\min \quad & a_j \prod_{i=1}^m (1 - p_{ij}x_{is}) - \sum_{i=1}^m x_{is}u_i - v_j \\
\text{s.t.} \quad & j \in J, \quad s \in S
\end{aligned} \tag{CG-sub}$$

- choose the constraint that violates the most, that is, the largest $\sum_{i=1}^m x_{is}u_i + v_j > q_{js}$.
- using the definition of q_{js} to get the above sub-problems.
- The problem can be separated and then solved for each given target j .
- It is intuitive that using each weapon to attack requires a cost, we try to balance the cost of the weapon and the probability of destroying the target.

motivation to use column generation

- Huge improvement in computational result for column enumerations
- The article on column enumeration mentions that column generation subproblems is hard to solve due to non-linearity
- The outer approximation method can be used in subproblems

Future Research Plan

Future Research Plan

- Implement the column generation method.
- Add sensors to problem formulation.

Alternate applications

$$\min \sum_{j=1}^n V_j \left(\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right) \quad (4)$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \in I, \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad \forall j \in J, i \in I. \quad (6)$$

- This model can not only be applied to the problem of WTA, but also some assignment problems with limited resources.

Thank you!