

# Inteligencia Artificial Avanzada

## Informe 1, Weapon Target Assignment Problem

Jaime Zamorano

15 de abril de 2019

### Resumen

En este informe se revisan investigaciones anteriores con respecto al Weapon Target Assignment Problem, en el cual se intenta asignar armas a ciertos objetivos para poder derribarlos. La asignación se hace en base a la probabilidad que tiene un arma de derribar un objetivo. Específicamente se investigó el estado del arte (como otros autores atacaron el problema) y se realizó una heurística greedy para resolver el problema.

## 1. Introducción

En este documento se abordó la temática del Weapon Target Assignment Problem, un problema np-completo. Se busca recabar información de investigaciones pasadas y ver como otros abordaron el problema. El documento se divide en definición del problema, donde se define la problemática, Estado del arte, donde se ven investigaciones anteriores y se programa un algoritmo greedy para resolver el problema para resolver el problema.

## 2. Definición del Problema

En el Weapon target Assignment Problem (conocido como WTA en la literatura) es un problema que busca asignar armas para derribar objetivos. Se tienen una cierta cantidad  $i$  de tipos de armas diferentes, cada uno compuesto de  $M_i$  armas, y del mismo modo, se tienen  $j$  tipos de objetivos diferentes con  $V_j$  objetivos por tipo. Para cada tipo de arma  $i$  existe una probabilidad  $P_{ij}$  de que derribe al objetivo de tipo  $j$ . Se puede asignar mas de un arma a un tipo de objetivo. La misión es minimizar la cantidad de objetivos supervivientes (lo que minimiza el daño que los objetivos hacen en la base), esto sujeto a que no se pueden asignar mas armas en total de las que ya existen (en total y por cada tipo diferente), sin embargo, no es necesario que todos los objetivos sean atacados (es una restricción blanda, que se toma dependiendo de como un autor resuelva el problema).

Este problema tiene 2 variaciones conocidas [2]. En el WTA estático, se conocen todos los objetivos que llegaran y se atacan todos con las armas en una sola ronda". En el WTA dinámico, se desconocen los objetivos que puedan llegar y estos vienen en "bloques" de tiempo (en  $t_1$  llegan  $X$ , en  $t_2$  llegan  $Y$ , etc). El problema que se desarrolla en este paper es el WTA estático.

### 3. Estado del Arte

Se revisará el como 6 autores diferentes atacan al Weapon Target Assignment Problem, cada autor tiene ciertas formas diferentes de ver el problema en cuanto a las restricciones del problema.

#### Redes neuronales

E. Wacholder planteó esto para defensa balística de misiles(BMD). Lo que se realiza, es usar una función de energía que representa restricciones del problema, y se busca minimizar la energía de toda la función(se optimiza de una forma similar a los multiplicadores de lagrange), luego se realiza el método de gradiente ascendente para pasar la información entre neuronas, estas luego deciden si se alcanza un criterio de convergencia o si se itera de nuevo. Los resultados arrojaron que el algoritmo producía resultados cerca del mínimo y que funciona bien para casos mas complejos, además es fácil de implementar en un circuito.[2]

#### Algoritmo genético

Zne-Jung-Lee y otros propusieron este algoritmo. Se resuelve la variante estática. Lo que se busca es el mismo "proceso" que los algoritmos genéticos, generar población, proceso de selección, cruzamiento, mutación y obtener una nueva población. La gran diferencia, es que la población no se selecciona de manera aleatoria entre todas las posibles transformaciones, si no que usa una heurística greedy para la generación(quitando algo de la aleatoriedad), dejando la parte de diversificación a la mutación(cosa de escapar de los óptimos locales que greedy casi siempre encuentra). En cuanto a resultados, este algoritmo tuvo un mejor rendimiento en comparación a algoritmos genéticos clásicos.[3]

#### Branch & Bound

Fue propuesto por Ravindra K. Ahuja y otros. Debido a la escasez de métodos exactos para resolver el WTA, ellos proponen(entre otras técnicas) un Branch & Bound. Este consiste en guardar memoria solo viendo la rama anterior y siguiente, explora usando depth first search(DFS). Los resultados arrojaron que Branch & Bound no fue una buena idea debido a que tuvieron un mal rendimiento.[5]

#### Simulated Annealing

Fue propuesto por Emrullah Sonuc y otros. Ellos trabajaron bajo el supuesto de que había un arma para cada objetivo(las instancias no consideraron cantidades para cada tipo) de distinto tipo cada una, usando la variación estática de WTA. El algoritmo Simulated Annealing consiste en tener una solución inicial, a partir de ella moverse de forma aleatoria a otra, si esta es de mayor calidad se mueve a esta, si no, una probabilidad basada en el temperatura del sistema(una variable matemática) determinada si el algoritmo puede avanzar a la solución de peor calidad. En cada iteración la temperatura baja en cierto factor y se vuelve a iterar. Los resultados indican que en general SA es un buen algoritmo para resolver el problema (tiene buenos resultados sobre la media de la literatura). [6]

#### Lagrange Relaxation

Propuesto por MingFang Ni y otros. Se aplicaron multiplicadores de lagrange para relajar algunas de las restricciones del problema. En el modelo matemático, plantean como restricción fuerte un parámetro  $t_j$ , que es la cantidad mínima de armas que se deben asignar a un objetivo j. Se plantea que al aplicar multiplicadores al problema, este se puede dividir en 2 subproblemas, y uno de estos subproblemas pueden

dividirse en mas subproblemas(que pueden resolverse todos paralelamente), y si uno encuentra una solución posible, todos encuentran una solución posible. Sus resultados arrojaron que para problemas pequeños la solución se encuentra rápido y después se mejora levemente(algo así como el gradiente descendente) y para instancias grandes funciona de manera decente con un buen tiempo de procesamiento.[1]

### Programación dinámica

Fue propuesto por Topi Sikanen, esta hecho para resolver la variante dinámica del WTA. Consiste en ocupar ventanas de tiempo discretas, donde una función se va incrementando. Luego se toman decisiones por cada ventana de tiempo(que están en isolación) basadas en optimización de Bellman. Para reducir la dimensionalidad del problema se usan ciertas políticas de look-ahead. Los resultados son buenos y comparables con el estándar de la literatura. [4]

## 4. Modelo Matemático

El modelo a ocupar del problema es el WTA estático.

### Parámetros

$I$  : tipos de armas diferentes  
 $J$  : tipos de objetivos diferentes  
 $M_i$  : cantidad de armas del tipo i  
 $V_j$  : cantidad de objetivos del tipo j  
 $P_{ij}$  : Probabilidad de que el arma de tipo i derribe al objetivo de tipo j  
 $Q_{ij}$  : Probabilidad de supervivencia del objetivo j al ataque del tipo i.  
 Se define  $Q_{ij}$  como :  $1 - P_{ij} \forall j$

### Variables

$X_{ij}$  : Entero que determina la cantidad de armas del tipo i que se asignan a los objetivos de tipo j.

### Función Objetivo

Para que los objetivos de un cierto tipo sobrevivan todas las armas deben fallar su tiro. Esto en teoría de probabilidades es multiplicar sucesivamente las probabilidades de supervivencia (ya que las probabilidades son independientes). El valor esperado de supervivientes entonces queda definido como la multiplicación entre la cantidad de objetivos y su probabilidad de supervivencia  $Q_{ij}$ . La función objetivo entonces busca minimizar dicha cantidad, y esta queda definida como:

$$MIN \sum_j^J V_j * \prod_i^I Q_{ij}^{X_{ij}} \quad (1)$$

Sujeto a:

$$\sum_i^I X_{ij} \leq M_i \forall j \quad (2)$$

(Notar que la restricción para que estén todas las armas asignadas es blanda)

$$X_{ij} \geq 0 \forall i, j \quad (3)$$

Notar que para este paper, la restricción de que todos los objetivos deben ser atacados, es blanda.

## 5. Descripción del algoritmo

El algoritmo greedy programado funciona de la siguiente manera.

1. Recolectar los datos necesarios como lo son: tipos de armas i, tipos de objetivos j, cantidad de armas por tipo M(lista), cantidad de objetivos por tipo V(lista) y la matriz de probabilidades P (probabilidad de que un arma i derribe a j).
2. Crear una matriz de dimensiones  $i \times j$  que este inicializada en 0(representación).
3. Usar V como SolucionActual y M como Restantes.
4. De la solución actual tomar el índice del mayor, guardar como indexj.(función miope).
5. Buscar en P en la columna indexj el índice de la probabilidad mas alta y guardar como indexi(función miope).
6. Si Restantes en la posición indexi no es 0, entonces asignar en la solución [indexi,indexj], si no, seguir iterando el paso anterior.
7. Multiplicar SolucionActual[indexj] por la probablidad (1-probabilidadMáxima) y guardar ahí mismo.
8. Restarle 1 a Restantes[indexi]
9. Resetear variables
10. Si la suma de todos los restantes es mayor a 0, repetir desde el paso 4, si no, avanzar al siguiente paso.
11. Evaluar la solución(función de evaluación)

## 6. Experimentos

Para la fecha de este entregable, se manejan 13 instancias diferentes. Estas están estructuradas de la siguiente forma:

Un número i que indica cantidad de tipos de armas.

Un número j que indica cantidad de tipos de objetivos.

i líneas que indican la cantidad de armas por tipo

j líneas que indican la cantidad de objetivos por tipo

$i \times j$  líneas que representan la probabilidad de que un arma i derribe un objetivo j( 11,12,13...21,22,.....,ij)

Para probar el algoritmo greedy, se utilizaron 13 instancias: wta3, wta5, wta10, wta20, wta30, wta40, wta50, wta60, wta70, wta80, wta90, wta100 y wta200, las cuales, como el nombre dice, tienen wtaX tipos de armas y objetivos, donde las cantidad  $M_i$  son un orden de magnitud mas pequeñas que  $V_i$ .

## 7. Resultados

Estos fueron los resultados obtenidos por el algoritmo greedy.

Instancia	Objetivos iniciales	Objetivos finales
wta3	90	11.88
wta5	377	0.16
wta10	719	0.00
wta20	1103	0.02
wta30	2089	0.01
wta40	2758	0.02
wta50	3167	0.00
wta60	3841	0.01
wta70	4631	0.03
wta80	5010	0.04
wta90	5591	0.07
wta100	6579	0.02
wta200	12461	0.05

Se puede ver que el algoritmo greedy redujo en gran medida el número de objetivos. Los buenos resultados del algoritmo pueden deberse a las probabilidades, ya que si una de estas es muy grande se reduce en gran medida uno de los objetivos solo asignando un arma. Para futuras instancias quizás sea considerable disminuir el número de armas o hacer que los tipos de armas sean menores que los tipos de objetivos.

## 8. Conclusiones

El estado del arte muestra que este problema tiene muchas variantes dependiendo de como se quiera ver, lo que puede relajar ciertas restricciones que para otros autores eran duras, sin mencionar que también se puede resolver el WTA dinámico (el cual es un poco mas complejo). Al final del día cada algoritmo desarrolla el problema de la forma que mejor le venga. Esta falta de consenso en la literatura puede ser la causante de la poca cantidad de instancias en la web, ya que muchos autores han recurrido a hacer unas propias.

En cuanto al algoritmo greedy desarrollado, este funciona bien debido a la naturaleza de las instancias, quizás reducir los tipos de armas y desbalancear la matriz de probabilidades pueda generar instancias mas complicadas en las que el algoritmo greedy no sea suficiente, pero por el momento, funciona.

## Referencias

- [1] MINGFANG NI, ZHANKE YU, FENG MA & XINRONG WU. *A Lagrange Relaxation Method for Solving Weapon-Target Assignment Problem*. China.Institute of Communication Engineering, PLA University of Science and Technology.2011

- [2] E. WACHOLDER. *A Neural Network — Based Optimization Algorithm for the Weapon-Target Assignment Problem*. USA . OAK RIDGE NATIONAL LABORATORY. 1989
- [3] ZNE-JUNG LEE, SHUN-FENG SU, MEMBER, IEEE, & CHOU-YUAN LEE. *Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms With Greedy Eugenics*. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 33, NO. 1 . 2003
- [4] TOPI SIKANEN. *Solving Weapon Target Assignment Problem with Dynamic Programming* 55670A TFY N. 2008
- [5] RAVINDRA K. AHUJA, ARVIND KUMAR, KRISHNA JHA & JAMES B. ORLIN. *Exact and Heuristic Methods for the Weapon Target Assignment Problem*. MIT Sloan School of Management. 2003
- [6] EMRULLAH SONUC, BAHA SEN & SAFAK BAYIR. *A Parallel Simulated Annealing Algorithm for Weapon-Target Assignment Problem*. Turkey.(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 4. 2017