DYNAMIC WEAPON-TARGET ASSIGNMENT PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRAH GÜNSEL

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2008

Approval of the thesis:

DYNAMIC WEAPON-TARGET ASSIGNMENT PROBLEM

submitted by **EMRAH GÜNSEL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ___________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ___________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. M. Kemal Leblebicioğlu ___________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Mübeccel Demirekler ___________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu ___________
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. İlkay Ulusoy ___________
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Afşar Saranlı ___________
Electrical and Electronics Engineering Dept., METU

Dr. Hüseyin Yavuz ___________
Defense Systems Technologies Group, ASELSAN

Date: ___________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Emrah Günsel

Signature :

ABSTRACT

DYNAMIC WEAPON-TARGET ASSIGNMENT PROBLEM

Günsel, Emrah

M. Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. M. Kemal Leblebicioğlu

September 2008, 113 pages

The Weapon-Target Assignment (WTA) problem is a fundamental problem arising in defense-related applications of operations research. Optimizing the WTA is about the selection of the most appropriate weapon for each target in the problem. Basically the aim is to have the maximum effect on targets. Different algorithms; branch and bound (B&B), genetic algorithm (GA), variable neighborhood search (VNS), are used to solve this problem. In this thesis, a more complex version of this problem is defined and adapted to fire support automation (Command Control Communication Computer Intelligence, C4I) systems. For each target, a weapon with appropriate ammunition, fuel, timing, status, risk is moved to an appropriate ammunitions, economy of fuel, risk analysis and time scheduling are all integrated into the solution. B&B, GA and VNS are used to solve static and dynamic WTA problem. Simulations have shown that GA and VNS are the best suited methods to solve the WTA problem.

Keywords: Optimization, Genetic, Search, Weapon Target Assignment.

ÖZ

DİNAMİK HEDEF SİLAH TAHSİS PROBLEMİ

Günsel, Emrah

Yüksek Lisans, Elektrik Elektronik Mühendisligi Bölümü

Tez Yöneticisi: Prof. Dr. M. Kemal Leblebicioğlu

Eylül 2008, 113 sayfa

Hedef silah tahsisinde (HST/WTA) optimizasyon ihtiyacı günümüz savunma sanayi uygulamalarında giderek artmaktadır. Hedef silah tahsisinin optimizasyonu, her bir hedefe en uygun silahın tahsis edilmesiyle ilgilidir. Genelde amaç hedeflerdeki tahribi arttırmaktır. Bu problemler için dallan ve sıçra (D&S/B&B), genetik algoritma (GA), değişken komşu arama (DKA/VNS) yöntemleri kullanılabilmektedir. Bu tezde, HST'nin çok daha karmaşık bir versiyonu tanımlanmış ve ateş destek otomasyon sistemleri için uyarlanmıştır. Her hedef için en uygun mühimmat miktarı, yakıt miktarı, zamanlaması, durumu ve risk seviyesi olan silah en uygun mevziiye intikal ettirilmiş ve görevin icrası sağlanmıştır. Hedef kıymeti, mühimmat ekonomisi, yakıt ekonomisi, risk analizi ve zamanlama da çözüme dahil edilmiştir. D&S, GA ve DKA yöntemleri statik ve dinamik HST çözümünde kullanılmıştır. Yapılan benzetim çalışmaları GA ve DKA yöntemlerinin HST problemini çözmek için çok uygun olduğunu göstermiştir.

Anahtar Kelimeler: Optimizasyon, Genetik, Tarama, Hedef Silah Tahsis.

To My Country

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. M. Kemal Leblebicioğlu for his encouragements, guidance, advice, criticism and insight throughout the research.

I would like to thank ASELSAN Inc. for facilities provided for the completion of this thesis.

I would like to forward my appreciation to all my friends and colleagues who contributed to my thesis with their continuous encouragement.

I would also like to express my profound appreciation to my family for their continuous support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AA	: Ammunition Availability
AC	: Assignment Cost
B&B	: Branch and Bound
BFS	: Breadth First Search
COP	: Constant of Penalty
C4I	: Command Control Communication Computer Intelligence
DFS	: Depth First Search
DLS	: Depth Limited Search
ED	: Effects Desired
FA	: Fuel Availability
FOW	: Fuel of Weapon
FC	: Fuel Cost
FP	: Fuel Price
GA	: Genetic Algorithm
GUI	: Graphical User Interface
LO	: Local Optimum
MA	: Mission Availability
MR	: Mission Risk
NA	: Number of Ammunitions
NOR	: Number of Rounds
NOT	: Number of Targets
NS	: Neighborhood Search
OLI	: Operational Lethality Index
PC	: Position Coordinates
RA	: Range Availability
RD	: Range Difference

SP : Shot Period
TA : Time Availability
TC : Target Coordinates
TO : Time Operational
TPE : Target Protection Effect
TPYE : Target Priority Effect
TTE : Target Type Effect
TTS : Time to Shoot
TTSP : Time to Shoot Position
TS : Target Size
TV : Target Value
QAP : Quadratic Assignment Problem
VNS : Variable Neighborhood Search
WA : Weapon Accuracy
WAC : Weapon Ammunition Cost
WAP : Weapon Ammunition Price
WC : Weapon Coordinates
WFE : Weapon Fuel Expense
WMA : Weapon Moving Ability
WOLI : Weapon OLI
WR : Weapon Range
WS : Weapon Status
WT : Weapon Type
WTA : Weapon Target Assignment
WTPA : Weapon Target Position Assignment
X : Assignment List of Missions
X_{ij} : j'th attribute of i'th mission of assignment list
T : Target List
W : Weapon List
P : Position List
W_{xx} : Weight of XX

CHAPTER 1

INTRODUCTION

1.1 What is WTA?

In the 21th century; Command, Control, Communication, Computer and Intelligence (C4I) systems help the defense-related problems and increase the capability of the limited sources under any combat situations. One of these problems is the Weapon Target Assignment (WTA). WTA problem is a fundamental problem arising in fire support automation (C4I) systems [12]. Different kinds of targets appearing in the combat area are to be assigned to the appropriate weapons. The time to decide the assignment is very important for decreasing the effects of the target on friendly units. On the other hand, using the optimum assignment increases the remaining armed forces and potential to win the combat.

This problem consists of optimally assigning W weapons (W is the number of weapons) to T targets (T is the number of targets) so that the total expected damage on the targets after all engagements (assignments) is maximum. Meanwhile, all off the targets should be assigned with a limited number of weapons. In the literature, different versions of these problems are studied.

1.2 What Has Been Done for WTA?

The WTA problem is generally formulated as a nonlinear integer programming problem [12] and is known to be NP-complete [3, 4]. Various methods of combinatorial optimization to solve NP-complete problems have been reported in the literature [1, 7].

For combinatorial optimization problems, an efficient allocation of limited resources to meet desired objectives is studied when the values of some or all of the variables are restricted to be integer. Constraints on basic resources restrict the possible alternatives that are considered feasible. Still, in most of these kinds of problems, there are many possible alternatives to consider and one overall goal determines which of these alternatives the best is. Combinatorial optimization models are often referred to as integer programming models where programming refers to planning so that these are models used in planning where some or all of the decisions can take on only a finite number of alternative possibilities [2]. Namely, all of these alternatives are members of the set of the solution space and it can be divided into two according to the constraints of the problem. The set of points which all the constraints are satisfied is denoted as feasible and the set of others is denoted as infeasible [8].

The methods used usually results in exponential computational complexities. As a consequence, it is difficult to solve these types of problems directly while the numbers of targets or weapons are large [3]. Note that NP-complete problems do have solutions, but they are all known to be exponential algorithms so far. A common mistake is to confuse NP-complete problems with unsolvable problems [5].

On the other hand, there is no method for the WTA problem which can produce the global optimum solution even for small size problems (for instance, with 20 weapons and 20 targets). Although several heuristic methods can be proposed to solve the WTA problem; due to the absence of a global optimum method, no estimates are available on the quality of solutions produced by such heuristics [12].

In the literature, different kinds of algorithms, for example branch and bound (B&B) [11, 14, 15], genetic algorithm (GA) [6, 9, 10], variable neighborhood search (VNS) [25, 26, 28] and some other methods are used to solve this problem.

WTA is also a Quadratic Assignment Problem (QAP) (the term “Quadratic” comes from formulating the problem as an optimization problem with a quadratic objective function [15]). The QAP is the problem of finding an assignment of limited number of sources with limited number of targets to minimize / maximize the evaluated cost values. B&B algorithms are to date known to be one of the most effective global solution procedures for QAP [11, 14]. On the other hand, GA is widely used as a search algorithm in various applications and also demonstrated satisfactory performances [6, 9, 10].

VNS is another meta-heuristic for combinatorial optimization. As its name suggests, this meta-heuristic systematically explores different neighborhood structures. The main idea underlying VNS is that a local optimum relative to a certain neighborhood structure is not necessarily an optimum relative to another neighborhood structure. For this reason, escaping from a local optimum can be done by changing the neighborhood structure. Despite being a relatively recent development, it is reported that VNS has been successfully applied to a wide variety of optimization problems such as vehicle routing, project scheduling, and automatic discovery of theorems, graph coloring and the synthesis of radar poly-phase codes [28].

1.3 Contributions of the Thesis for WTA Problem

In this thesis, a more complex version of WTA problem is defined and adapted to fire support automation (C4I) systems. In this problem, not only the weapons are assigned to the targets, but also positions are assigned to the weapons.

For each target, a weapon with appropriate ammunition, fuel, timing, status, risk is moved to an appropriate position to shoot the target. So, this problem can be said to be a “Weapon Target Position Assignment (WTPA) Problem”. Some of the critical parameters of weapons are obtained from a source called Operational Lethality Index (OLI) [13]. OLI gives the effectiveness and basic properties of weapons in combat area. These constants associated with weapons and their missions have been modified slightly to keep the secure information. Target value, mission risk, economy of ammunitions, economy of fuel, risk analysis and time scheduling are also evaluated and are all integrated into the solution as in real applications.

Also, WTA problems are studied in two versions: static and dynamic. In the static version, all the inputs to the problem are fixed; that is, all targets are known, all weapons are known, all the positions are known, and all weapons are engaged to the targets from the positions in a single stage. The dynamic version of the problem is a multi-stage problem where some weapons are engaged to the targets from the positions at the first stage, the outcome of this engagement is assessed and strategy for the next stage is determined [12]. In this thesis, the static WTA problem is studied first; and then the same algorithms are used to deal with a simplified version of the dynamic WTA problem. B&B, GA and VNS are used to solve static and dynamic WTA problems.

1.4 Comparison of this Thesis and Literature

- In this thesis, WTA problem is improved as WTPA problem. So, not only the weapons but also position for each target is decided.
- WTA problem is developed as a real application for fire support automation systems. Additionally, in order to increase the effect on targets, real constraints specific to this area which are listed below is also used for the evaluations.
    - Moving time, shooting time and target time,
    - The limited sources (ammunitions, fuel),
    - The risks (being detected and shot) of missions.
- For WTPA problem solution, not only the assignment but also the full mission information is listed.
    - Assignment of weapon,
    - Assignment of position,
    - Fuel information,
    - Ammunition information,
    - Timing information,
    - Availability information.
- WTPA solution time is improved by:
    - Heuristics,
    - Problem specific coding.
- WTPA problem is solved not only statically but also dynamically. So, a result obtained previously can be used in new assignments. In this way, in a dynamic environment a better solution is obtained in a shorter time. Every change in conditions (time, targets status changes, weapons status changes, positions changes) results in a new assignment list. In chapter 2, B&B will be presented as a method of solution for static assignments and VNS will be presented as a method of solution for dynamic assignments.

1.5 Outline of the Thesis

- In chapter 2, problem formulation, integrated parameters and feasibility constraints are given.
- In chapter 3, B&B and VNS methods are given with information for classical search techniques and all related details.
- In chapter 4, GA method is given with general information and the implementation details.
- In chapter 5, application environment, the simulation program and user interface (GUI) is given with details. This chapter can be used as a user guide for the software.
- In chapter 6, simulations of codes developed are given in detail to compare the developed results.

These results can be improved using a better computer. In this study, all the simulations are done with the same computer to be able to compare the results. It has been observed that GA and VNS are best suited methods to solve the WTPA problem. Using these methods it is possible to attack moderately large size (up to 50 weapons and 50 targets) WTPA problems and obtain almost optimal solutions in fairly large number of instances (up to 100 weapons and 100 targets) within a few (1-2) minutes.

- In chapter 7, a summary of the thesis, conclusions and the future work is given.

CHAPTER 2

PROBLEM FORMULATION

In this thesis, an optimization problem associated with the assignments of a series of weapons, targets and (missions) positions has been dealt with. Each assignment is checked for feasibility and results in an assignment score.

The assignment list is stored in a matrix "X_{ij}". "X_{ij}" contains all the mission information of mission X_i. X_{i1} is the target ID, X_{i2} is the weapon ID, X_{i3} is the position ID and X_{i4} is the assignment cost. Other attributes of the mission are denoted as a function of X_i, which represents the corresponding matrix element or information of X_i. For example "X_{iED}" means the effect desired for the mission X_i (can be found by X_{i1} = target ID and the target list T), or "X_{iRD}" means the range difference for the mission X_i (can be found by the formula (1)).

The WTPA problem includes the assignments which are basically defined with the items below and the optimization of the assignment according to the related information.

2.1.1 Weapon Information

Weapons' attributes are listed in a weapon list ("W_{ij}") and each element of this list (i'th weapon) contains the information below:

- ("W_{i1}") Unit Reference Number,
- ("W_{i2}") Weapon Type,
- ("W_{i3}") Weapon Position (latitude, longitude, elevation),
- ("W_{i4}") Weapon Status (not operational, operational/ready, unknown, moving),
- ("W_{i5}") Weapon Range (by OLI method [13]),
- ("W_{i6}") Shot Period (by OLI method [13]),
- ("W_{i7}") Moving Ability (by OLI method [13]),
- ("W_{i8}") Weapon OLI (by OLI method [13]),
- ("W_{i9}") Time Operational,
- ("W_{i10}") Time to Position for Shooting (TTSP),
- ("W_{i11}") Weapon Accuracy (by OLI method [13]),
- ("W_{i12}") Number of Ammunitions,
- ("W_{i13}") Weapon Fuel Expense,
- ("W_{i14}") Fuel of Weapon,
- ("W_{i15}") Weapon Ammunition Price.

2.1.2 Target Information

Targets' attributes are listed in a target list ("T_{ij}") and each element of this list (i'th target) contains the information below:

- ("T_{i1}") Target ID,
- ("T_{i2}") Target Generic Type (personnel, weapon, mortar, artillery, armor, vehicles, rocket/missile, supply dump, command center, equipment, building, terrain feature, assembly area, air defense artillery, bridge),
- ("T_{i3}") Degree of Protection (unknown, prone, standing, dug in, prone/overhead, cover),

- ("T_{i4}") Number of Targets,
- ("T_{i5}") Target Position (latitude, longitude, elevation),
- ("T_{i6}") Fire Mission Priority (normal, urgent, priority),
- ("T_{i7}") Effects Desired,
- ("T_{i8}") Target Size (area),
- ("T_{i9}") Time to Shoot.

2.1.3 Position (of Mission) Information

Positions' attributes are listed in a position list ("P_{ij}") and each element of this list (i'th position) contains the information below:

- ("P_{i1}") Latitude,
- ("P_{i2}") Longitude,
- ("P_{i3}") Elevation.

2.1.4 Time (of Mission) Information

Schedule of fire mission evaluation contains the time periods listed below:

- Movement (according to the distance between position of the weapon and position of the mission),
- Deployment (standard time of each weapon to deploy on the mission position),
- Tour of duty (time required for the rounds to be completed).

2.1.5 Risk (of Mission) Information

Risk factor attributes are listed below:

- Risk of movement (risk of being recognized while moving),
- Risk of mission (risk of being recognized while firing).

2.1.6 Fuel (of Weapon and Mission) Information

Fuel factor contains the fuel capacity of each weapon type (func.(“W_{i2}”)) and according to the movements of these weapon types (func.(“W_{i13}”)), remaining fuel (“W_{i14}”) is evaluated for possible new mission evaluations.

2.1.7 Ammunition Information

Ammunition factor contains the ammunition quantity capacity of each weapon type and according to the missions completed; remaining ammunition quantity (“W_{i12}”) is evaluated for possible new mission evaluations.

2.1.8 Mission (Assignment) Evaluation

Every fire mission (“X_i”) is assumed to include one weapon (“X_{i2}”), one position (“X_{i3}”) and one target (“X_{i1}”). Since the combat area is divided into discrete positions for the weapons for missions, possible positions are also predefined for the WTA problem. So, each mission should include one weapon that should go to one position and attack one target at a specific time and this assignment should obey the availability restrictions (that is described in detail in “Constraints” section).

2.1.9 Assumptions

Only “tactical fire control” (WTPA) is studied in this thesis. “Technical fire control” (the ballistic evaluation algorithms) is not included. If a fire mission assignment fails because of a technical problem / insufficiency, the fire support unit should reject fire mission.

2.1.10 Constraints

Basic feasibility constraints of the problem are range, ammunition, time and fuel availability.

- For each assignment, weapon range should be compared with the distance between mission position coordinates (PC) and the target coordinates (TC). If weapon range (WR) is greater than the mission, then it is said to be range available (RA).

$$X_{iRD} = (X_{iWR} - |X_{iTC} - X_{iPC}|) \quad (1)$$

$$X_{iRD} \geq 0 \rightarrow X_{iRA} = 1, \quad 0\ otherwise$$

(2)

- For each assignment, number of rounds (NOR) should be evaluated, which is directly proportional to the target type effect (TTE), effects desired (ED) for that target, number of targets (NOT), target size (TS)(surface area of the target, every ammunition has a maximum effect area), target protection effect (TPE) (every combat element protects itself against ammunitions), and inversely proportional to the weapon accuracy (WA) (which is a function of range in between weapon and target and weapon type) and weapon OLI (WOLI) (operational lethality index which represents the effect power of the weapon and value of the weapon) [13].

$$X_{iNOR} = [(X_{iTTE} \times X_{iED} \times X_{iNOT} \times X_{iTS} \times X_{iTPE} \times X_{iRD})/(X_{iWOLI} \times X_{iWA} \times X_{iWR})] \quad (3)$$

If the number of rounds is smaller than the available number of ammunitions (NA) for that weapon at the mission time, the mission is said to be ammunition available (AA).

$[X_{iNA} - X_{iNOR}] \geq 0 \rightarrow X_{iAA} = 1, \;\; 0 \; otherwise$

(4)

- Time availability (TA), is another critical point for the assignment. The weapon status (WS), time to position for shooting (TTSP) (movement of the weapon to the available position which depends on the distance and the weapon moving ability), latest time to shoot target (TTS) are taken into account to be ready at the time to shoot that target.

$$X_{iTTSP} = |X_{iWC} - X_{iPC}| \div X_{iWMA} \quad (5)$$

$$[(X_{iTTSP} - X_{iTO} - X_{iTTS}) \geq 0] \wedge [X_{iWS} = 1] \rightarrow X_{iTA} = 1, \;\; 0 \; otherwise \quad (6)$$

- Since movements are taken into account, fuel availability (FA) should also be considered; so weapon fuel expense (WFE) for the movement and time of movement (TTSP) is also compared with the current fuel of weapon (FOW).

$$(X_{iFOW} - X_{iTTSP} \times X_{iWFE}) \geq 0 \rightarrow X_{iFA} = 1, \;\; 0 \; otherwise \quad (7)$$

Other limits are:

$$X_{iWC} = (c_1, c_2, c_3), where \; c_1, c_2 \in [0, \; 999999], c_3 \in [0, \; 9999] \quad (8)$$

$$X_{iPC} = (c_1, c_2, c_3), where \; c_1, c_2 \in [0, \; 999999], c_3 \in [0, \; 9999] \quad (9)$$

$$X_{iTC} = (c_1, c_2, c_3), where \; c_1, c_2 \in [0, \; 999999], c_3 \in [0, \; 9999] \quad (10)$$

$$X_{iTA} \in \{0, \; 1\} \quad (11)$$

$$X_{iRA} \in \{0, \; 1\} \quad (12)$$

$$X_{iFA} \in \{0, \; 1\} \quad (13)$$

$$X_{iAA} \in \{0,\ 1\} \quad (14)$$

$$X_{iED} \in [0,\ 100] \quad (15)$$

$$X_{iNOT} \in [0,\ 99] \quad (16)$$

$$X_{iTS} \in [0,\ 9999] \quad (17)$$

$$X_{iTTS} \in [0,\ 9999] \quad (18)$$

$$X_{iFOW} \in [0,\ 1000] \quad (19)$$

2.1.11 Cost Function Evaluation

An assignment score (profit) is evaluated for assignments. This score includes fuel, risk, ammunition and time costs. Target value is evaluated as a profit and is obtained according to the target value that is directly proportional to the target type, effects desired, number of targets, target priority. Score of the mission (that is to be maximized) is evaluated by subtracting the costs from the profits. The total score after these operations will be called as the score of the assignment (AC).

If all of the feasibility checks are succeeded, mission is said to be available (MA). If a mission is not available, then a constant of penalty (COP) term is created (effectively) by not inserting the assignment profit into the score. Infeasibility is also used as decision criteria for search algorithms; for example if a mission is not feasible, further searches through that mission are stopped also.

Evaluation of the score for each assignment is determined by the following formulas:

$$X_{iRD} = (X_{iWR} - |X_{iTC} - X_{iPC}|) \quad (1\text{-}20)$$

$$X_{iRD} \geq 0 \rightarrow X_{iRA} = 1,\ \ 0\ otherwise \quad (2\text{-}21)$$

$$X_{iNOR} = [(X_{iTTE} \times X_{iED} \times X_{iNOT} \times X_{iTS} \times X_{iTPE} \times X_{iRD})/(X_{iWOLI} \times X_{iWA} \times X_{iWR})] \quad (3\text{-}22)$$

$$[X_{iNA} - X_{iNOR}] \geq 0 \rightarrow X_{iAA} = 1,\ \ 0\ otherwise \quad (4\text{-}23)$$

$$X_{iTTSP} = |X_{iWC} - X_{iPC}| \div X_{iWMA} \quad (5\text{-}24)$$

$$[(X_{iTTSP} - X_{iTO} - X_{iTTS}) \geq 0] \wedge [X_{iWS} = 1] \rightarrow X_{iTA} = 1,\ \ 0\ otherwise \quad (6\text{-}25)$$

$$(X_{iFOW} - X_{iTTSP} \times X_{iWFE}) \geq 0\ \rightarrow X_{iFA} = 1,\ \ 0\ otherwise \quad (7\text{-}26)$$

$$X_{iTV} = X_{iTTE} \times X_{iED} \times X_{iNOT} \times X_{iTYPE} \quad (27)$$

$$X_{iMR} = (X_{iNOR} \times X_{iSP} + X_{iTTSP}) \times X_{iWOLI} \div X_{iWMA} \quad (28)$$

$$X_{iFC} = (|X_{iWC} - X_{iPC}|) \times X_{iFP} \div X_{iWFE} \quad (29)$$

$$X_{iWAC} = X_{iNOR} \times X_{iWAP} \quad (30)$$

$$X_{iTA} \times X_{iRA} \times X_{iAA} \times X_{iFA} = 1\ \leftrightarrow X_{iMA} = 1 \quad (31)$$

$$X_{iAC} = (X_{iTV} - X_{iFC} - X_{iWAC} - X_{iMR}) + X_{iMA} \times COP \quad (32)$$

The score function of the WTPA that is to be maximized is the sum of all assignments' scores:

$$\sum_{i=1}^{T}\{(w_{TV}.X_{iTV} - w_{FC}.X_{iFC} - w_{WAC}.X_{iWAC} - w_{MR}.X_{iMR}) + X_{iMA} \times COP\} \quad (33)$$

subject to the equations 20 to 32 and the assumption that all targets are assigned to only one weapon (34):

$$\forall X_{k1} = t_k,\ X_{i1} \neq t_k \quad st. i \in [1,\ T] - \{k\} \quad (34)$$

In eq. (33), “w” is the weight of each element and “X_{ij}” contains all the mission information of mission X_i. For example X_{i1} is the target ID, X_{i2} is the weapon ID, X_{i3} is the position ID and X_{i4} is the assignment cost. w (the user defined constants) can take values between 0 and 9.

2.1.12 Static and Dynamic Evaluation of WTPA

Static WTPA problem is the WTPA problem with static inputs at an instant. So, it is solved before a combat to organize the fire support. After the combat started, the conditions and characteristic values of targets and weapons start changing. Since the targets are activated (started moving), timing differs in time, as well. Dynamic WTPA problem should be considered for this case, which is nothing but solving the static WTPA problem, according to changing inputs, starting from a previously obtained solution. So, static WTA is used (with static inputs) before the combat starts, on the contrary dynamic WTA is used (with dynamic inputs) after the combat starts.

CHAPTER 3

SOLUTION TECHNIQUES 1: BRANCH & BOUND

3.1 Classical Search Techniques

Any problem to be solved should be represented first as a mathematical model. After the mathematical model is constructed, problem can be solved by any search technique. There are different search techniques that use an explicit search tree that is generated by an initial state and the successor function which together defines the state space. When a state can be reached from different paths, a search graph is obtained rather than a search tree.

The root of the search tree is said to be a search node corresponding to the initial state. The first step in a search is to test if this is a goal state. Starting from the initial state, if it is not a goal state, some other states are considered. This is done by expanding the current state; that is, applying the successor function to the current state, thereby generating a new set of states. Each combinatorial optimization problem contains a different number of alternatives for each expansion. Also the sequence of search for similar alternatives is critical. This is the essence of search, following up one option first and putting the others aside for later consideration, which may or may not lead the algorithm to an optimal solution. Choosing, testing and expanding continue until either a solution is found or there are no more nodes to be expanded [16].

There are many ways to represent nodes, but here a node is assumed to be a data structure with five components. These are the state (the state in the state space to which the node corresponds), the parent node (the node in the search tree that has generated this node), the action (the action that was applied to the parent to generate the node), the path cost (the cost of the path from the initial state to the node) and the depth (the number of steps along the path from the initial state).

It is important to remember the distinction between the nodes and the states. A node is a bookkeeping data structure used to represent the search tree. A state corresponds to a configuration of the world. Thus, nodes are particular paths, as defined by the parent node pointers whereas states are not. Furthermore, two different nodes may contain the same world state, if the state is generated via two different search paths.

The collection of nodes that have been generated but not yet expanded need to be represented: These nodes are called the fringe. Each element of the fringe is a leaf node, that is, a node with no successors in the tree [16].

The fringe could be a set of nodes. The search strategy would be a function that selects the next node to be expanded from this set. The strategy function might be checking every element of the set to choose the best one. So, this strategy could be computationally expensive.

The output of a problem-solving algorithm is either failure or a solution. After several recursive evaluations, a final solution should come out. The performance of the algorithm will be evaluated according to the following four criteria; completeness (is the algorithm guaranteed to find a solution when there is one?), optimality (does the strategy find the optimal solution?), time complexity (how long does it take to find the solution?) and space complexity (how much memory is needed to perform the search?) [16].

If the algorithm have no additional information about the states beyond the problem definition, generating new successors and checking the goal is repeated until the end. These kinds of algorithms are called unified search algorithms. Possible unified search algorithms are:

- Breath-First,
- Depth-First,
- Depth-Limited,
- Iterative Deepening Depth-First.

On the other hand, if an algorithm is able to select better candidates, it is called a heuristic search algorithm. Possible heuristic search algorithms are:

- Tabu,
- Greedy,
- A(*) star.

3.1.1 Unified Search Algorithms

Unified search algorithms are generally called as blind search algorithms. These algorithms search every node in the search tree to find out all possible solutions. So, unified search algorithms are complete but they are not optimal [16, 17]. Time and space complexities of unified search algorithms are also worse than heuristic algorithms.

3.1.1.1 Breadth-First Search (BFS)

BFS starts at a starting node (no assignment), which is at level 0. In the first stage, the algorithms visit all nodes at the first level. In the second stage, the algorithm visits all nodes at the second level. The new nodes, which are adjacent to the previous level node, are all searched and the algorithm goes on like this. The BFS traversal terminates when every node has been visited [18]. BFS is optimal as it returns the path with the fewest steps. Not necessarily the shortest route, but fewest steps. If there is a one-step answer, it will find it since it searches one- step answers first and if there is not, it will find it since it look for a two step answer, and so on. Even if the graph goes on a loop and if it is known that the goal is at depth d, algorithm expands it out depth by depth until depth d is reached. From the viewpoint of space complexity, the algorithm has to save only one layer, but exponential growth is such that there are as many nodes in the last processed layer as the sum of nodes of all previous layers [17].

BFS algorithm is not useful for WTPA type problems, because the goal is assumed to be at leaf nodes; so BFS requires very large amount of space and time complexity.

3.1.1.2 Depth First Search (DFS)

DFS starts at a specific node S (initial node) in the search graph, which becomes current node. The algorithm traverses the graph by any edge (u, v) incident to the current node u. If the edge (u, v) leads to an already visited node v, then it backtracks to the current node u. If, on other hand, edge (u, v) leads to an unvisited node v, then it goes to v and v becomes the current node. The algorithm proceeds in this manner until it reaches to a "dead end". At this point it starts backtracking. The process terminates when backtracking leads back to the starting node [18]. DFS will not necessarily find the shortest path even in search trees with dead ends. Trees with recursive loops may also cause indefinite search. From the viewpoint of space complexity, the algorithm only has to save the maximum number of children that a node can have (assuming no loop is repeated in the search). From the viewpoint of time complexity, time does not depend on how close the goal is to the starting node (in the worst case). The search could go all the way through the graph before it finds the simplest path, depending on the order that it adds the children to the agenda (which is unspecified by the algorithm). So, if it is a graph where goals can be in any level, timing is worse; on the other hand if goals are known to be at the leaf nodes, timing is better according to BFS [17].

DFS is a more useful algorithm for WTPA type problems with respect to BFS, but without heuristics, it is not capable enough to solve complex problems in a limited time.

3.1.1.3 Depth-Limited Search (DLS)

DLS starts at a specific depth in a search tree, which becomes the current depth. Then the algorithm searches through deeper levels until the limit of search is reached. At this point the algorithm starts backtracking to the initial node. The process terminates when backtracking leads back to the starting node. This method is a subset of DFS, but differs in the limit of search. The differences are the desirable space and time complexity, avoidance of infinite search and the possibility of finding no solution. So, DLS has some advantages compared to DFS but it is yet not capable enough to solve the complex problems in a limited time and also is not a complete search compared to DFS.

Since DLS is not a complete search algorithm and it is not effective without heuristics, it is not applicable for WTPA problems.

3.1.1.4 Iterative Deepening

Iterative deepening is applied as DLS and if the algorithm cannot find the goal, it continues starting from the last depth through the new limit of the search. So, the search goes on until the solution is found. This algorithm recovers the disadvantages of DLS and also has the advantages of it.

Space and time complexity is minimized compared to DFS and the search becomes complete.

Iterative deepening is not a useful algorithm for WTPA problems either; because of the large amount of space usage and time requirement.

3.1.2 Heuristic Search Algorithms

Unified search algorithms all need to be improved to solve NP-complete problems. Therefore, different kinds of heuristics should be used to decrease the time to reach a solution for a WTPA problem. A heuristic function maps a state onto an estimate of the lowest cost to the goal from that state [17].

H (n) = estimated cost of the best path from node n to goal node [16]

Some of the heuristic algorithms from literature are explained below:

3.1.2.1 Tabu Search

A tabu search algorithm uses the possible sub-solutions and proceeds by transiting from one sub-solution to another. All the possible sub-solutions of a global solution are examined and the best non forbidden sub-solutions are selected. Note that these selections may decrease the quality of the solution. A tabu list stores all the previously exploited selections which are now forbidden [20].

This heuristic algorithm may decrease the complexity of WTPA type problems for both space and time. Some nodes can be defined as forbidden and taken to the tabu list. Avoiding the repetitions of these nodes the number of weapons, positions or targets decrease and this decreases the computations exponentially.

3.1.2.2 Greedy (Best-First) Search

Greedy search tries to expand the node that is closest to the goal; therefore it first expands the node which scores best in some pre-evaluation (heuristic) function. In assignment problems, algorithm uses a greedy search evaluation function which expands nodes that have the least estimated cost or best score of the current nodes, regardless of the cost so far [16, 19].

H (n) = estimated cost of the best path from node n to the goal node [16].

Greedy search is useful for WTPA type problems since the decision of which candidate to start with effects the time for finding the first goal. Starting with the best candidate will on the other hand increase the complexity of the search and result in extra evaluations in every level of the search. Greedy search algorithm can be represented as in Figure 3-1:

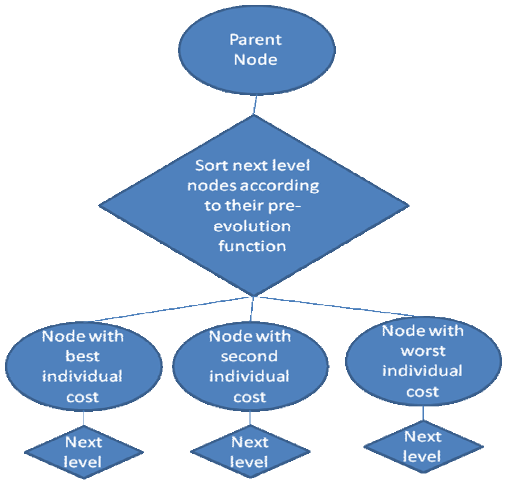


Figure 3-1: Greedy Search

On the other hand, finding the optimal solution with greedy search is clearly not guaranteed, so this algorithm has to be improved [17]. All the nodes should be searched even if their goal estimate is very low.

3.1.2.3 A* (A star) Search

A* search is very similar to greedy search. In fact, A* is a new generation of greedy. It not only evaluates individual costs of candidates but also combines a measure of the cost-so-far and the pre-evaluated score (cost-to-goal) to get a very efficient search strategy [19].

H (n) = estimated cost of the best path from node n to goal node

+ cost to reach the selected node [16].

A* search algorithm can be represented as in Figure 3-2:

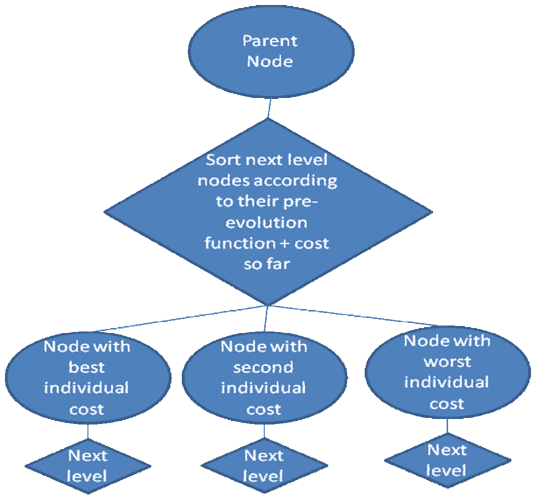


Figure 3-2: A Search*

A* search is also useful for WTP type problems since the decision of which candidate to start with is selected in a better way by this algorithm. On the other hand, A* may use lots of memory. For big search spaces, A* may cause run out of memory error [17]. So A* search can only be used with some stopping criteria and the search will then be not completed.

3.2 Branch & Bound (B&B)

Without any heuristics, any search algorithm may search all combinations for finding out the best combinational series of missions. This set of all combinations for t targets, w weapons and p positions are evaluated as follows.

For each mission / assignment, there should be one target, one weapon and one position (of mission) as expected. The search depends on the target assignment, because the aim is to assign every target with a mission and obtain the maximum score according to the values of targets and costs of the missions. Therefore, basis of the search is the assignment of targets first.

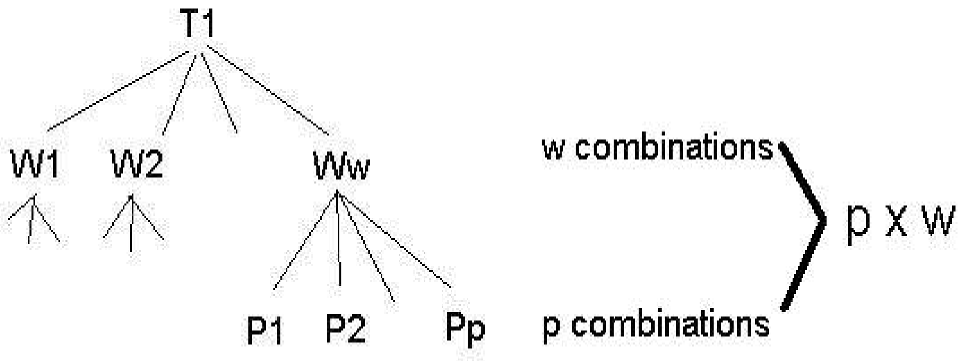


Figure 3-3: Assignment of Targets

So, for each target there are w numbers of possible weapons and for each weapon there are p numbers of positions. Combining two factors, (p x w) number of combinations are available. Since all the targets will be tried to be assigned, a tree of search will be obtained as shown in Figure 3-4. Starting from the top, at each one more level through deeper levels, one more target will be assigned after the one ahead. For the case study 1, for the path indicated in Figure 3-4 with thick line; target T2 will be assigned to one of the w x p weapon-position combinations and then target T1, then target T3, then target T4, then target T6 and so on. The weapon to be used to shoot the target T2 will be busy until the movement ends and all the shots finished, so that weapon will not be time-available for a while to be assigned to a new target and, therefore, weapon assigned to target T2 will have less chance to shoot target T1.

According to the tree below, there will be too many missions to be searched. To find out the number of assignments (nodes), possible number of leaf nodes should be decided first. The number of nodes in the following tree is equivalent to the number of paths available and that is equivalent to P (t, t) = t! (the permutation of number of targets “t” to “t”).

$$P(n,k) = n!/(n-k)! = n.(n-1)(n-2)\dots(n-k+1) \tag{35}$$

For one level above the leafs, there will be P (t, t-1) different nodes. So, total target assignment node number can be found by the sum of the all levels.

On the other hand, each node in the tree in fact, includes (p x w) number of combinations inside. For instance, for the path indicated in case study 1 with thick line in Figure 3-4, for target T3, if target T2 is assumed to be assigned, (w x p) number of combinations are available for T2. Then if T1 is assumed to be assigned, (w x p) number of combinations is available for T1. So, (w x p) x (w x p) number of combinations are available before target T3 is assigned.

Finally the total number of nodes to be search by the full search algorithms is given by equation 36:

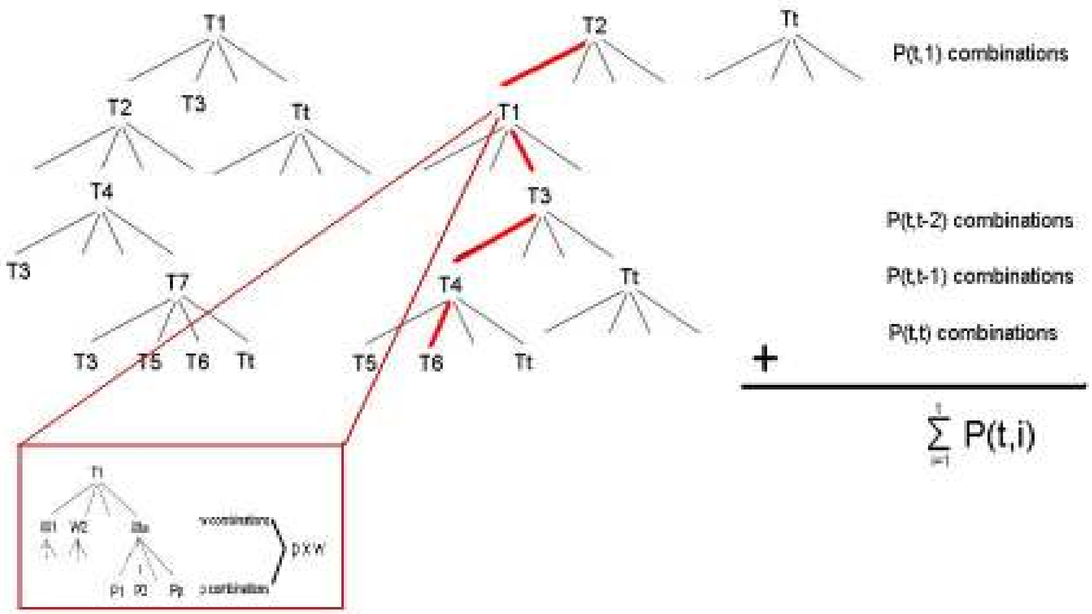


Figure 3-4: Case Study 1: Search Tree Example of WTA

$$\sum_{i=1}^{t}\left[P(t,\ i) \times (p \times w)^{i}\right] \qquad (36)$$

For example, for t=4, w=4, p=4 evaluation, total number of combinations is 1674304.

$$\sum_{i=1}^{4}\left[P(4,\ i) \times (4 \times 4)^{i}\right] = 1674304 \qquad (37)$$

Since for larger problems, the space needed is very large, a stopping criterion for the algorithms is needed.

The applied stopping criterion is the number of nodes evaluated. For example the search algorithm developed on the development environment evaluates one billion nodes (assignments) in one minute. It is needed to stop this algorithm every thirty seconds to decide either terminate or go on searching. If the user selects to stop and is satisfied with the best solution which is found till that second, the “stop” button is used and results are listed. If “go on” button is used, search goes on to find better solutions.

Stopping criteria (a constant number of searched nodes) are generally used for bigger number of targets, weapons and positions. The evaluation environment can run out of memory because the number of expected nodes to be searched gets exponentially large. Getting out of memory can make the problem unsolved since the results cannot be seen. On the other hand, if the algorithm is stopped before memory limit is reached, search cannot be complete.

So, for these types of problems, algorithms and heuristics are very important and critical to reach the best solution in optimum time.

In this thesis, B&B algorithm with different heuristics is used to solve WTPA problem. Parameters and requirements (constraints) of this specific problem are defined in the “Scope of Thesis” part. Solution alternatives are briefly described in the following sections.

3.2.1 B & B - General Information

The fundamental difficulty in solving WTA like assignment problems is that it takes too long to search for an optimal solution in the space of permutations.

A B&B algorithm generates a tree, where each node represents a sub problem to be solved. The activity at each node consists of computing a lower bound on the sub problem and then generating child sub problems (“branching”). Computing lower bounds allows us to prune (or “fathom”) the tree along paths that cannot lead to an optimal solution. In the case of QAP, the initial, root problem is to find a permutation (assignment) that minimizes an objective function. To branch the space of permutations, it is divided by either disallowing or forcing assignments to be made in each child sub problem [11].

For WTA problems, the algorithm starts from the first target and goes deeper and deeper as soon as it can assign weapons and positions to the remaining targets. From this point of view, B&B is a DFS. Once the algorithm hits a target that cannot be assigned, it goes to the next target and works like depth first algorithm. The difference occurs in the bound decisions which are results of different heuristics.

One basic heuristics is to use the feasibility functions of specific WTA problem for the selected target. Therefore, for the basic B&B algorithm (without other heuristics), only the feasibility or availability of the mission is used to bound in this thesis. So, basic B&B searches the feasible region only.

For example, for the case study 2, assume a specific problem with 3 targets, 3 weapons and 3 positions. For this case, the parameters are defined in Table 3-1, Table 3-2 and Table 3-3:

Table 3-1: Case Study 2: Target Information

TARGET ID	T0001	T0002	T0003
Latitude	330000	330000	330000
Longitude	331070	330100	330200
Elevation	330	330	330
Target Type Index	Bridge	Bridge	Air- Defense- Artillery
Protection Index	Prone- Overhead- Cover	Prone- Overhead- Cover	Prone- Overhead- Cover
Priority Index	Vital	Urgent	Vital
Number Of Targets	1	1	1
Target Size (m2)	100	100	100
Effects Desired (%)	90	90	90
Time To Shoot (sec.)	5000	5000	5000

Table 3-2: Case Study 2: Weapon Information

WEAPON URN	W0001	W0002	W0003
Latitude	330000	330000	330000
Longitude	320000	320000	320000
Elevation	300	300	300
Type	Panter	Fırtına	UT1
Status	Ready	Ready	Ready
# of Ammunition	100	100	100
Fuel of Weapon	3100	2100	1100

Table 3-3: Case Study 2: Position Information

POSITION ID	P0001	P0002	P0003
Latitude	330100	320200	330300
Longitude	328000	325000	325000
Elevation	310	310	310

For this case, a search starts with T0001, W0001 and P0001 and goes on like:

Table 3-4: Case Study 2: Search Tree Memory

Node ID	Target	Weapon	Position	Ammu. Avai.	Fuel Avai.	Time Avai.	Range Avai.	Score	Search Level
1	T0001	W0001	P0001	1	1	1	1	8754209	1
2	T0002	W0001	P0001	0	1	1	1	-963542	2
3	T0002	W0001	P0002	0	0	1	1	-1809699	2
4	T0002	W0001	P0003	0	0	1	1	-1204168	2
5	T0002	W0002	P0001	1	1	1	1	9505007	2
6	T0003	W0001	P0001	0	1	1	1	-428922	3
7	T0003	W0001	P0002	0	0	1	1	-1098066	3
8	T0003	W0001	P0003	0	0	1	1	-618988	3
9	T0003	W0002	P0001	1	1	1	1	9869527	3

Table 3-4 cont'd: Case Study 2: Search Tree Memory

Node ID	Target	Weapon	Position	Ammu. Avai.	Fuel Avai.	Time Avai.	Range Avai.	Score	Search Level
10	T0003	W0002	P0002	1	0	1	1	-397348	3
11	T0003	W0002	P0003	1	0	1	1	-204542	3
12	T0003	W0003	P0001	0	0	1	1	-1566689	3
13	T0003	W0003	P0002	1	0	1	0	-40722	3
14	T0003	W0003	P0003	0	1	1	1	-9132941	3
15	T0002	W0002	P0002	1	0	1	1	-739806	2
16	T0002	W0002	P0003	1	1	1	1	9503012	2
17	T0003	W0001	P0001	0	1	1	1	-428922	3
18	T0003	W0001	P0002	0	0	1	1	-109866	3
19	T0003	W0001	P0003	0	0	1	1	-618988	3
20	T0003	W0002	P0001	1	1	1	1	9821468	3
21	T0003	W0002	P0002	1	0	1	1	-393607	3
22	T0003	W0002	P0003	1	1	1	1	9843517	3
23	T0003	W0003	P0001	0	0	1	1	-1566689	3
24	T0003	W0003	P0002	1	0	1	0	-40722	3
25	T0003	W0003	P0003	0	1	1	1	-9132941	3
26	T0002	W0003	P0001	0	0	1	1	-3013211	2
...									
58	T0001	W0002	P0001	1	1	1	1	9600402	1
59	T0002	W0001	P0001	1	1	1	1	8672458	2
63	T0003	W0002	P0001	1	1	1	1	9869527	3
...									
181	T0002	W0001	P0001	1	1	1	1	8672458	1
182	T0001	W0001	P0001	0	1	1	1	-881791	2
...									
603	T0003	W0003	P0003	0	1	1	1	-9132941	1

The information according to the case study 2 presented in Table 3-4 is explained as

- Node 1: A feasible mission is found; therefore, the first element of the remaining targets (T0002, T0003) is selected as T0002 and it is expanded in node 2 as the level 2 of the search.

- Node 2: Since the mission assignment of T0002, W0001 and P0001 is not feasible because of ammunition (ammunition availability), this node is not branched as T0003. This node is bounded and the next candidate (T0002, W0001, P0002) is selected to be searched next.

- Node 9: Last mission assignment is obtained as T0003, W0002, P0001 and a total score of the mission list is evaluated: Score of node 1 (8.754.209) + score of node 5 (9.595.007) + score of node 9 (9.869.527) (=28.128.743). At this point, the score evaluated and the list members are saved as the current best mission list.

- Node 10: The total score is to be maximized in this problem. Since the solution found after node 9 is not known to be a best solution, search goes on. The last assigned mission is rejected. Any other assignment for T0003 can be found until node 14. Therefore, second assigned mission is also rejected. Search goes on to level 2.

- Nodes 16 and 20: Search algorithm finds another solution. The total score is evaluated; however, since it is not better (28.078.689 < 28.128.743) than the previous solution, this mission list is not replaced with the best mission list saved before.

- Nodes 9 and 20: These two nodes are seemed to be the same; because both of them include T0003, W002, and P0001. However, score of these two nodes are different because of the path difference. Path difference is that for node 20, W0002 travelled to P0003 (node 16) and then turned back to P0001, but for node 9, W0002 just stayed at P0001. Traveling affected the time, risk and fuel costs.

  Here it is shown that WTPA nodes are DEPENDENT to the previous nodes (path). This result is very important for the solution methods, heuristics and algorithms used.

- Nodes 58, 59 and 63: Search algorithm finds another solution. The total score is evaluated and since it is better (28.142.387 > 28.128.743) than the previous best solution, this mission list is replaced with the best mission list saved before.

- Node 181: After all the branch of mission list starting with T0001 finished, search algorithm goes on the second main branch with T0002 assigned first.

- Node 182: Here the ammunition availability is not satisfied and the constant of penalty (COP) is applied to the score. Therefore, the assignment cost (score) can be seen to be very small (also negative) because of the penalty of the constraints.

- Node 603: After all the branches are searched, B&B finishes the search and the stored best mission list is output to the user interface. The number of visited nodes is 603. In fact, for 3 targets, 3 weapons and 3 positions, there are 4.887 nodes. (P(3,1)x(3x3)^1)+ P(3,2)x(3x3)^2)+ P(3,3)x(3x3)^3)=4.887) However, since infeasible nodes like node 2 is bounded, the possible unnecessary nodes below node 2 are discarded. Therefore, the number of evaluated nodes is decreased from 4.887 to 603 (only basic bounding is used).

After B&B algorithm terminates, obtained optimum mission list is (node 58, 59 and 63) given in Table 3-5.

Table 3-5: Case Study 2: Optimum Mission List

ID	Tar get	Wea pon	Posi tion	Avai.	Miss. Start	Miss. End.	Amm. Start	NOR	Amm. End	Fuel Start	Fuel End	M. Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
# of Nodes	603											
Total. Cost	28.142.387											

Some other examples of problems with random inputs are searched and the numbers of nodes for these searches are:

Table 3-6: Advantage of (Basic) Bounding

Target	Weapon	Position	Possible Nodes	Evaluated Nodes
3	3	3	4887	603
3	10	10	167490	58900

3.2.2 B & B with Tabu Search

While searching through the nodes, for each level, the same number of alternatives is used as given in Figure 3-5.

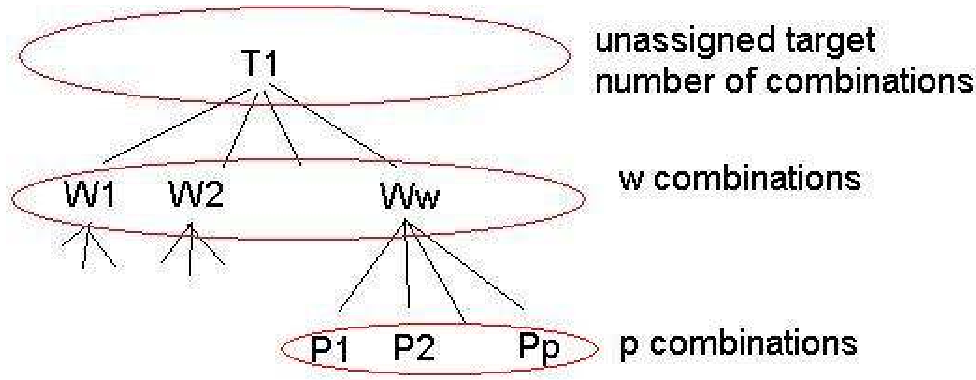


Figure 3-5: Target Assignment Combinations

Trying to assign a weapon to some of the targets may be a waste of time. For example a target which is 100 km far away may never be shot by any of the weapons from any of the positions, because the maximum range of the weapons is less than 100 km. Another possibility is that a weapon can never shoot any target since it is not ready or it is broken. Also, it may not have enough fuel to go any of the shooting positions. Another possibility is that one of the positions cannot be appropriate for any combination of assignment. Therefore, in order to avoid these kinds of useless assignments, unused targets, weapons and positions are investigated for feasibility before the search and eliminated from the available target, weapon and position lists when necessary. This procedure is called as tabu and the eliminated alternatives forms the tabu list.

To see the effect of tabu method, by inserting random inputs with random numbers of targets, weapons and positions, obtained results are shown in Table 3-7, with and without tabu columns shows the number of nodes searched for a full search.

Table 3-7: Advantage of Tabu List for B&B

Target	Weapon	Position	# of nodes searched without Tabu	# of nodes searched with Tabu
3	3	3	603	268
3	10	10	58900	17670

3.2.3 B & B with Tabu and Greedy

Obtaining the unused elements of the lists and eliminating them, number of nodes searched decreases and an advantage of time and space complexity is achieved by using tabu list. Another additional heuristic that can be added is the greedy search. By greedy search, while the tabu list is being formed, each combination of target, weapon and position is checked for feasibility. The additional part is that a matrix of these combinations is produced and for each feasible combination, the matrix is filled with one, infeasible combinations are denoted with zeros. Using this matrix, each target, weapon and position can be scored with the total number of feasible combinations of itself and this number gives the heuristic of that target, weapon or position. After obtaining the heuristic function of each list element, they can be re-sorted within the list to be searched first. Re-sorting each list will cause the better combinations to be searched first and when the algorithm reaches a stopping criterion, it will result in a better solution. Re-sorting is applied as follows:

- Targets with much less possibility to be shot are sorted first.
- Weapons with much more possibility to be used are sorted first.
- Positions with much more possibility to be used are sorted first.

Although greedy method is used with tabu discarding, it is also possible to use only greedy method in the user interface.

To see the effect of greedy method, by inserting random inputs with constant numbers of targets (50), weapons (50) and positions (50), following results are obtained.

Table 3-8: Advantage of Greedy for B&B

Total cost obtained with only Tabu after 500000 nodes visited	225982321
Total cost obtained with Greedy + Tabu after 500000 nodes visited	227870285

Here, since the better alternatives of targets, weapons and positions are evaluated previously (greedy), a better solution is obtained in the same time period according to only tabu method.

3.2.4 B & B with Tabu, Greedy and A*

A* is a new generation of greedy. It combines a measure of the cost-so-far and the pre-evaluated score (cost-to-goal) to get a very efficient search strategy. Before branching at node n, the estimated cost of the best path from node n to goal node is added to the cost to reach the selected node. Therefore, the possible total cost estimate is obtained. If this estimated total cost is better (bigger) then the current best solution, the algorithm branches for node n; if it is worse (smaller), the algorithm decides that branching is useless. When branching is useless, this method enables the algorithm to bind to other nodes and accelerates the search. When the stopping criteria are reached, it is possible to have a better solution since a bigger part of the search tree is searched.

For WTPA applications, a score estimate of all missions is needed. Therefore, all the combinations are searched first and the possible biggest score is found out. Then, at each branch or bound decision point, this estimate of cost is multiplied with the remaining level number and added to the current score. If the score estimate of that branch is not bigger than the best score found until that moment, bound procedure is applied. If a better solution is available, branch procedure is preferred.

B&B without A* searches all the feasible search tree, but B&B with A* searches a smaller search tree and saves time. The search is not complete, but A* enables the algorithm to produce a result faster.

3.3 Variable Neighborhood Search (VNS)

Search algorithms for WTA type problems are very complicated. Since they are very complex, repeating the same algorithm each time needs too much time. To apply search algorithms to WTA dynamically, a shortcut, VNS should be used. This search algorithm searches a sub-search tree in a shorter time. So, VNS can be thought to be the dynamic or specific implementation of B&B.

The basic idea of VNS is to explore different neighborhoods for the solution space whenever a local optimum is reached by using a local search method. VNS is told to be meta-heuristics for solving combinatorial optimization problems, such as traveling salesman problem, p-median problem, minimum sum of squares clustering problem, resource-constrained project scheduling problem, vehicle routing problems, scheduling problems, graph coloring, linear ordering problem, etc [26]. A set of neighborhood structures is predefined (N_k,k = 1, 2,...,kmax), an initial solution is found, and a stopping criterion is determined in the first phase of VNS. In the second phase, the search is started with the initial solution until a local optimum for k = 1 is found. Then, steps illustrated in Figure 3-7 are carried out repeatedly until a stopping criterion (a constant number of searched nodes) is met [26, 27].

VNS is based on a systematic change of neighborhood within a local search algorithm. In this thesis, knowing the previous solution, neighbors of the solution are searched one by one. Methodology of search is the same but the searched state set is in the neighborhood of the old local optimum solution. Three different kinds of branching rules (or neighborhood property) are used for this search:

- Target order change: Missions are replaced with each other (Figure 3-6); therefore the order of the mission is changed and the changes in the schedule are recovered by changing the timing.
- Weapon change: Weapons of individual missions are changed (Figure 3-6); therefore the changes in the weapon capabilities are recovered by changing the weapon.
- Position change: Positions of individual missions are changed (Figure 3-6); therefore the changes in the coordinates, range and movement capabilities are recovered by changing the mission position.

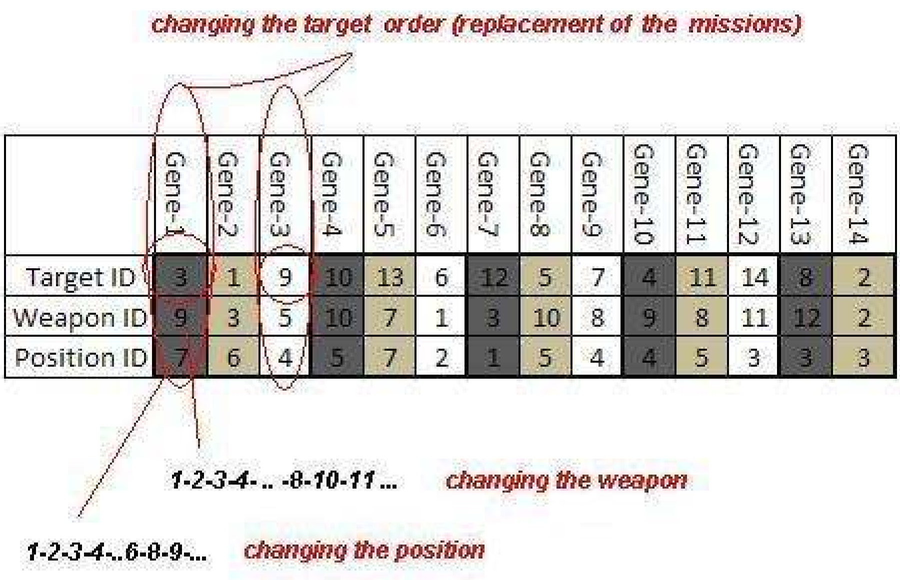


Figure 3-6: Neighborhood Search

Each branching operation causes one unit distance change from the local (old) optimum (LO) solution. Using a distance limit to search enables the algorithm to search a limited closeness of the local optimum. Changing this limit optimizes the time and the possibility of finding the global optimum.

For example, for case study 3 in Figure 3-7, for the given optimum (LO) solution

(T1, Ww, P6) + (T2, Ww, P2):

- First, weapon Ww goes to position P6 and shoots T1,
- Second, weapon Ww goes to position P2 and shoots T2.

One change in position assignment (P6→P2, LO→LO') causes one distance change and if the new solution LO' is better than LO, it is said to be the new local optimum. After searching all one unit distance changes (in target, weapon or position change direction), if necessary a second branching can be done to find a better solution. Here, the method of search can be depth first, breath first or any other. In this thesis, DLS is applied with a variable depth (distance) limit.

For the Figure 3-7, LO'' is a two unit distance alternative since one position and one weapon is different from LO. (P6→P2 + Ww→W2, LO→LO'→LO'')

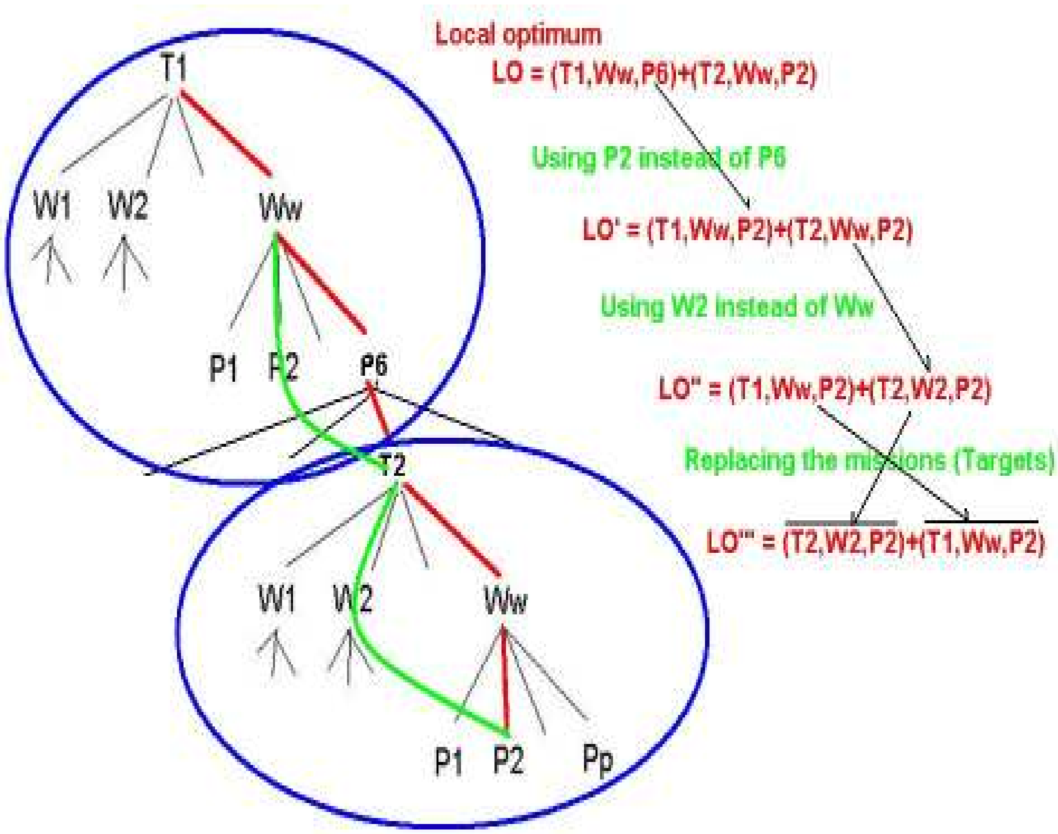


Figure 3-7: Case Study 3: Neighborhood Search Distances

By this search algorithm, the number of the nodes to be searched for one level depth can be evaluated as the sum of target order change combinations (C (t, 2)), weapon change combinations (t x w) and position change combination (t x p) where t is the number of assignment targets in the initial solution, w is the number of weapons and p is the number of positions.

So, for a d distance of search:

$$\sum_{i=1}^{d}[(t \times (t-1)/2) + w + p]^{i} \tag{38}$$

is the number of possible nodes to be searched for VNS algorithm.

Any change in the solution can affect the feasibility of the mission list. Therefore, each new generated neighbor should go through a feasibility check and reproduction process. Feasibility and reproduction processes control all the missions and if there exists an unfeasible mission, they make an appropriate change and reproduce the list. Therefore, after this process, a feasible mission list is guaranteed to be produced for that neighbor. If a target is deleted, it is recovered with this procedure also.

While these parameters are being changed, unassigned targets are also controlled. If some new targets are added, extra missions are also added to the end of the mission list to success for those targets / missions.

For WTA problems, VNS enables the algorithm to compensate small input changes and refresh the solution in a smaller time division. Therefore, after WTA is solved by any suitable algorithm statically, by using the solution of the first search, the new solutions can be produced with VNS dynamically.

CHAPTER 4

SOLUTION TECHNIQUES 2: GENETIC ALGORITHM

4.1 General Information

Genetic Algorithms (GAs) are one of the most popular heuristic algorithms to solve optimization problems and they are extremely useful. As regards to combinatorial optimization problems, GAs in their elementary form, is not competitive with other heuristic algorithms such as tabu search [23]. GAs were initially developed to mimic the process of evolution and natural selection. They simulate the process of evolution by taking a population of solutions and applying genetic operators in each iteration/generation. GAs evaluates each solution in the population according to some fitness measure and give highly fit solutions higher chance of survival. They generate new "offspring" solutions through recombination and replace less fit solutions. GAs repeat this evaluation-selection-recombination cycle until a satisfactory solution is found or some termination criterion is met.

GAs are different from traditional optimization methods in the following four ways;

1. GAs work with a coding of the parameter set, not the parameters themselves. The parameters and control variables are usually coded in a finite-length binary string (chromosome).

2. GAs search from a population of strings, not a single point. This means that GAs work from a population, and therefore, is less likely to get stuck in a local optimum compared to other methods.

3. Unlike traditional methods, GAs do not require much auxiliary information about the problem to work properly; they only require pay-off (objective function) information, not derivatives or other knowledge. GAs attempt to optimize the objective function using a strategy essentially independent of the problem at hand. There are plenty of these "black-box" problems in the real world and GAs have become a promising approach for these difficult problems.

4. GAs use probabilistic transition rules, not deterministic ones. The best solution is not guaranteed to be found. The differences between GAs and other traditional methods contribute to GAs' robustness. GAs are able to quickly find good or acceptable solutions for those difficult combinatorial optimization problems where traditional methods usually fail or give poor results. They have been proven effective on those difficult problems, and usually they are methods of the last resort [15].

GAs contain different kinds of terminologies and operations as in the nature. These operations are described next.

4.1.1 Encoding and Chromosome Representation

Since the genetic theory depends on the chromosomes, GA is built on encoded chromosomes of the problem. A chromosome represents a solution to the problem and is encoded as a vector of random keys. In a direct representation, a chromosome represents a solution of the original problem, and is usually called a genotype [22]. So, an encoding scheme maps feasible solutions of the problem to chromosomes [15]. These chromosomes include the genes. These genes can be visualized as replaceable elements of the problem.

$$\text{Chromosome} = \left(\underbrace{\text{gene}_1, \ldots, \text{gene}_n}_{\textit{attribute 1}}, \underbrace{\text{gene}_{n+1}, \ldots, \text{gene}_{2n}}_{\textit{attribute 2}}, \underbrace{\text{gene}_{2n+1}, \ldots, \text{gene}_{2n+m}}_{\textit{attribute 3}} \right)$$

Figure 4-1: Chromosome Representation

4.1.2 Initial Population Generation

The performance of a GA is often sensitive to the quality of its initial population. The quality of the initial population depends both on the average fitness (that is, the objective function value) of individuals in the population and the diversity in the population. Losing on either count tends to produce a poor GA. By having an initial population with better fitness values, we typically get better final individuals. Further, high diversity in the population inhibits early convergence to a locally optimal solution [21].

4.1.3 Crossover

The crossover scheme is widely acknowledged as critical to the success of GA. Crossover is done between two different chromosomes which are called as parents. The operation is basically the replacement of genes or gene blocks of the parents. The crossover scheme should be capable of producing a new feasible solution (i.e., a child) by combining good characteristics of both parents. Preferably, the child should be considerably different from each parent [21]. It is observed that when a good initial population is generated, then crossover results in better generations.

4.1.4 Mutation

Any change in a chromosome sequence is called a mutation. Most often case is the point mutation. In addition, there are two frame shift mutations: deletion, in which one or more base-pairs are lost, and insertion, in which one or more base-pairs are inserted into the sequence. Mutation provides a mechanism to maintain the population diversity [23]. Mutation and crossover are fundamentally important processes. Without mutation, after several generations are produced, all the chromosomes will be similar, and without crossover, there is no use of good chromosomes. Both are needed for evolution.

4.1.5 Feasibility Check

A chromosome after any operation should be checked for feasibility (i.e., with respect to problem constraints). If the chromosome is not feasible, it should not be used as a member in the population since it will cause other chromosomes to be infeasible. This type of chromosomes are filtered or repaired by a reproduction process before a new population is constructed.

4.1.6 Reproduction

Infeasible chromosomes are changed by suitable heuristics (operators or mutations). The heuristics that can be used can be defined according to the problem specifications. After reproduction, new chromosomes are inserted into the population.

4.1.7 Elimination

Initial population is used to generate new generations by the methods above. After each generation is produced the elitist chromosomes are selected as the best members to be inserted to the next populations. Main advantage of this method is that the available best solution is monotonically improving from one generation to the next. However, it can lead to a rapid population convergence to a local minimum. Nevertheless, this can be overcome by using high mutation rates [22].

4.2 Application for WTPA

For the application GA for WTPA, the GA starts with a set of randomly selected chromosomes as the initial population that encodes a set of possible solutions. Variables of the problem are represented as genes in a chromosome. Each operation of GA is applied on these chromosomes as described in the following sections:

4.2.1 Encoding and Chromosome Representation

The chromosome for WTPA problem is a combination / list of missions / assignments:

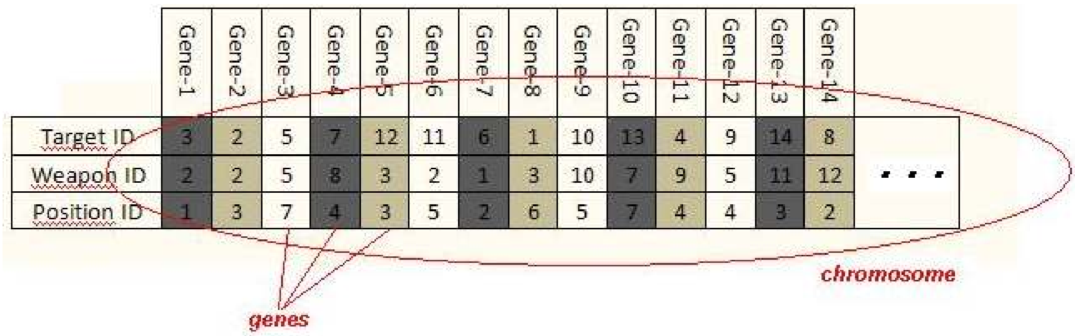


Figure 4-2: Chromosome Representation of WTPA

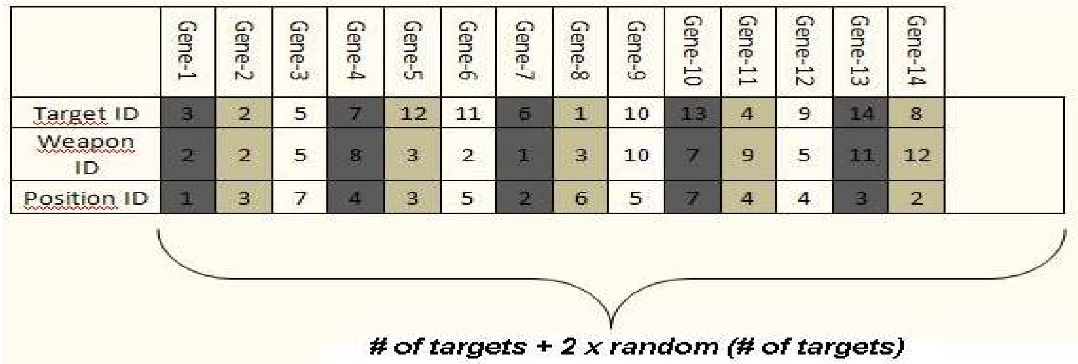
WTPA problem is briefly introduced by the chromosome structure in Figure 4-2. Each gene contains a target, a weapon and a position ID. WTPA chromosomes are encoded as a vector of targets. The value of the i'th gene indicates which target, which weapon is assigned to shoot from which position. Consider the chromosome above (Figure 4-2). It represents an assignment list where target 3 is assigned to weapon 2 and position 1 (gene-1); target 2 is assigned to weapon 2 and position3 (gene-2) and so on.

Noted that:

- There is only one assignment for each target. Therefore, target X is used in only one gene.
- On the other hand, there may be many assignments for each weapon because one weapon can shoot too many targets.
- Also there may be many assignments for each position, because positions are available mission areas and too many weapons can use one position at the same time.
- Length of the chromosome during the GA operations is variable, but for a feasible chromosome, length should not exceed the number of targets. If it exceeds, it means that one target is assigned twice and the second mission is meaningless since the target is already shot.
- Chromosomes are evaluated according to their feasibilities in WTA problem (ammunition, fuel, time and range), which are obtained by evaluating the considered feasibility functions.

4.2.2 Initial Population Generation

The initial population generator of the applied algorithm produces 100 chromosomes. Each gene of each chromosome is randomly generated by using a random function of the target, weapon and position numbers. A chromosome is obtained by combining all the generated genes together. The length of this chromosome is randomly selected between the target number and 3 times target number.



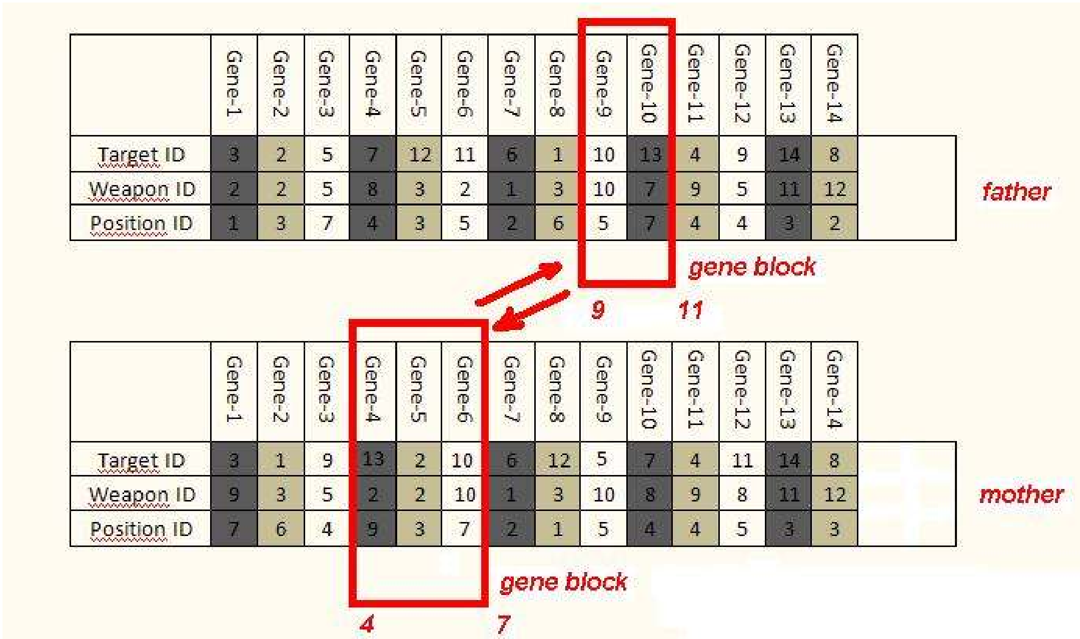
	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	Gene-12	Gene-13	Gene-14
Target ID	3	2	5	7	12	11	6	1	10	13	4	9	14	8
Weapon ID	2	2	5	8	3	2	1	3	10	7	9	5	11	12
Position ID	1	3	7	4	3	5	2	6	5	7	4	4	3	2

Figure 4-3: Initial Chromosome

Variable length chromosomes increase the ability of obtaining feasible genes. Since the randomly generated chromosomes that are newly generated may not be feasible, they go through a feasibility check. Then if they are infeasible, they also go through a reproduction process. Feasibility check and reproduction is described in the following sections in detail.

4.2.3 Crossover

Crossover of the WTPA chromosomes are done by an interchange of variable length gene blocks between father and mother chromosomes. Two random numbers between “one” and “father chromosome length” are selected. Two random numbers between “one” and “mother chromosome length” are also selected. These four numbers defines the “start” and “end” of gene blocks of the corresponding parents. After selecting the gene blocks (Figure 4-4), these blocks are replaced with each other.

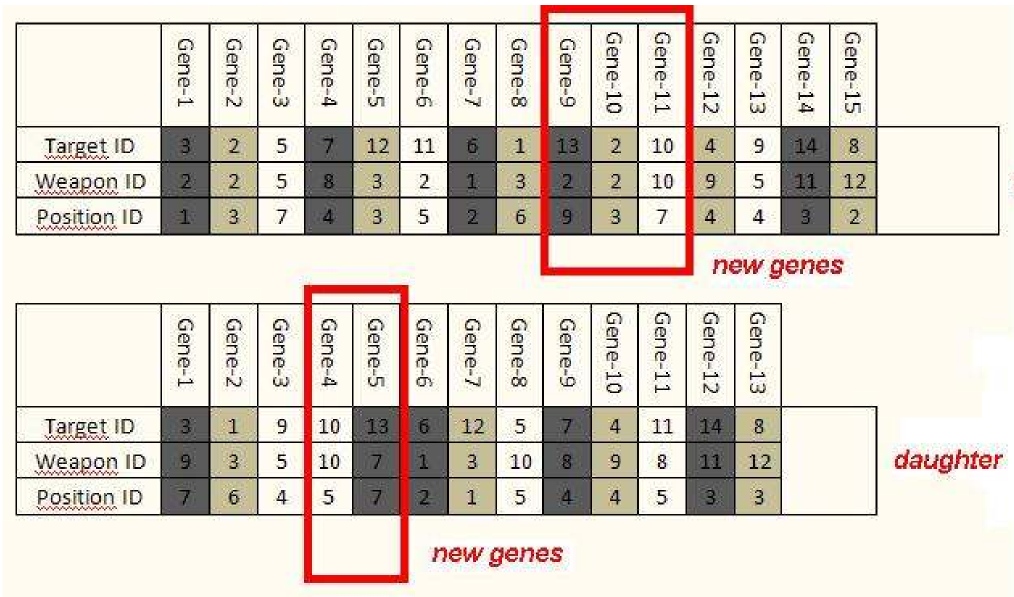


	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	Gene-12	Gene-13	Gene-14
Target ID	3	2	5	7	12	11	6	1	10	13	4	9	14	8
Weapon ID	2	2	5	8	3	2	1	3	10	7	9	5	11	12
Position ID	1	3	7	4	3	5	2	6	5	7	4	4	3	2

	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	Gene-12	Gene-13	Gene-14
Target ID	3	1	9	13	2	10	6	12	5	7	4	11	14	8
Weapon ID	9	3	5	2	2	10	1	3	10	8	9	8	11	12
Position ID	7	6	4	9	3	7	2	1	5	4	4	5	3	3

Figure 4-4: Crossover Operation of WTPA

So, new chromosomes (Figure 4-5) which are denoted as son and daughter are obtained.



	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	Gene-12	Gene-13	Gene-14	Gene-15
Target ID	3	2	5	7	12	11	6	1	13	2	10	4	9	14	8
Weapon ID	2	2	5	8	3	2	1	3	2	2	10	9	5	11	12
Position ID	1	3	7	4	3	5	2	6	9	3	7	4	4	3	2

	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	Gene-12	Gene-13
Target ID	3	1	9	10	13	6	12	5	7	4	11	14	8
Weapon ID	9	3	5	10	7	1	3	10	8	9	8	11	12
Position ID	7	6	4	5	7	2	1	5	4	4	5	3	3

Figure 4-5: Chromosomes after Crossover

Because of the dependency in-between the genes, replacement of the genes results in infeasibility in the new chromosomes. So they go through a feasibility check. Then, if they are feasible, they also go through a reproduction process. For instance, the son chromosome shown in Figure 4-5 will fail in feasibility check because target 2 is assigned twice. Feasibility check and reproduction is described in the following sections in detail.

4.2.4 Mutation

Mutation of the chromosomes for WTPA is done in a sequence of operations. First, the length of the chromosome is measured. The required length to reach the target number (maximum feasible chromosome length) is found by subtracting the length of the gene from the total number of targets. Then, the unassigned targets are determined. Unassigned targets are added to the end of the chromosome by a random order and with random weapons and targets. After this operation, the length of the chromosome becomes equal to the number of targets (Figure 4-6).

	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	missing targets
Target ID	3	1	9	10	13	6	12	5	7	4	11	
Weapon ID	9	3	5	10	7	1	3	10	8	9	8	
Position ID	7	6	4	5	7	2	1	5	4	4	5	

	Gene-1	Gene-2	Gene-3	Gene-4	Gene-5	Gene-6	Gene-7	Gene-8	Gene-9	Gene-10	Gene-11	Gene-12	Gene-13	Gene-14
Target ID	3	1	9	10	13	6	12	5	7	4	11	14	8	2
Weapon ID	9	3	5	10	7	1	3	10	8	9	8	11	12	2
Position ID	7	6	4	5	7	2	1	5	4	4	5	3	3	3

added genes

Figure 4-6: Mutation Operation - 1 of WTPA

Then, one of the old genes is replaced with one of the new genes to increase the randomness (Figure 4-7).

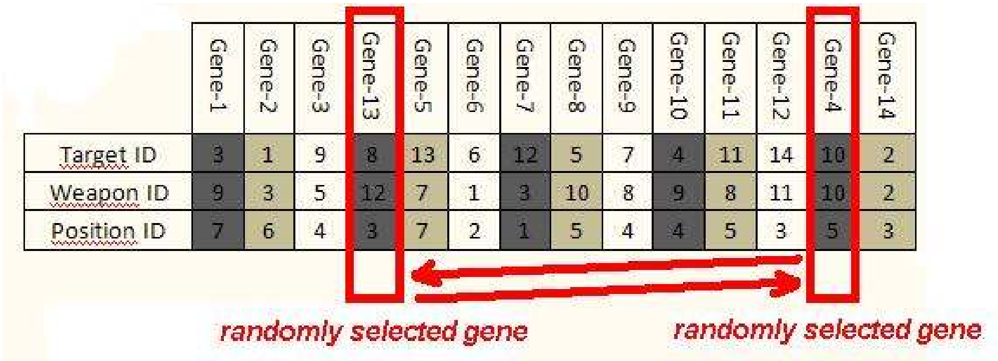


Figure 4-7: Mutation Operation - 2 of WTPA

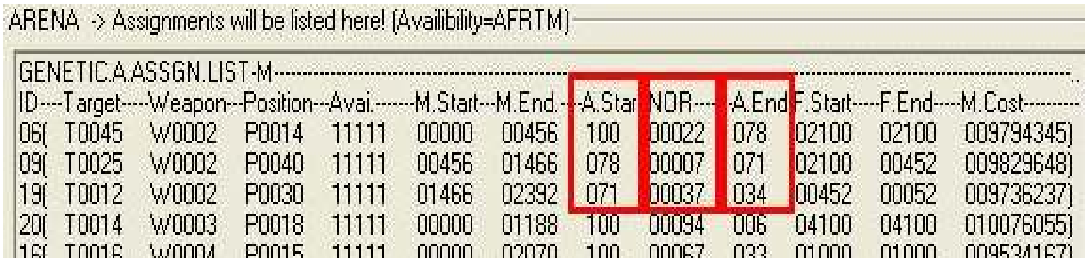
As expected these operations can also affect feasibility; therefore, the new genes go through a feasibility check. Then, if they are infeasible, they also go through a reproduction process. Feasibility check and reproduction is described in the following sections in detail.

4.2.5 Feasibility Check

Feasibility rules of the WTPA problem are described in the following sections in detail. For each mission all these checks are done and according to the results, a constant of penalty is decided to be applied or not.

4.2.5.1 Ammunition Availability

Each weapon in the weapon list has a capability of carrying a limited amount of ammunition. This limit can also be the limit of the army for that weapon. This amount of ammunition decreases in each mission done by the weapon and after enough of ammunition spent, the weapon cannot accept new missions. The amount of ammunition that is required by the mission depends on different reasons. One of these reasons is the type of the weapon. For instance, one weapon requires 4 shots to a particular target while another weapon has to shoot 40 to produce the same effect. If the algorithm selects the weapon that requires 40 shots, this decision shows its effect in ammunition feasibility check after a while. The weapons which require too much ammunition for the target cannot accept the mission. Here, there is an assumption that each target can only be shot by one weapon. Two weapons should not shoot a target together. Therefore, after each gene produced or changed, ammunition availability is checked. If it is failed, a penalty is also applied to the cost of the gene.


ARENA -> Assignments will be listed here! (Availibility=AFRTM)

GENETIC.A.ASSGN.LIST-M

ID	Target	Weapon	Position	Avai.	M.Start	M.End.	A.Star	NOR	A.End	F.Start	F.End	M.Cost
06(	T0045	W0002	P0014	11111	00000	00456	100	00022	078	02100	02100	009794345)
09(	T0025	W0002	P0040	11111	00456	01466	078	00007	071	02100	00452	009829648)
19(	T0012	W0002	P0030	11111	01466	02392	071	00037	034	00452	00052	009736237)
20(	T0014	W0003	P0018	11111	00000	01188	100	00094	006	04100	04100	010076055)
16(	T0016	W0004	P0015	11111	00000	02070	100	00067	033	01000	01000	009534167)

Figure 4-8: Ammunition Usage

In this example (Figure 4-8), missions are listed according to the ascending weapon IDs. It can be seen that after 3 missions the amount of ammunition for weapon W0002 shows 34 and it could not be enough for any other mission with more than 34 ammunition requirement because of the infeasibility.

4.2.5.2 Fuel Availability

Each weapon in the weapon list has a capability of carrying a limited amount of fuel. This limit can also be the limit of the army for that weapon. This fuel is decreased if the weapon moves. Each weapon spends different amounts of fuels for a unit distance according to its weapon type. After a while, if the weapon moves too much, it goes out of fuel and cannot accept any more missions.

GENETIC.A.ASSGN.LIST-M

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
17(	T0020	W0010	P0011	11111	00759	00979	069	00011	058	00653	00503	010291860)
18(	T0023	W0010	P0011	11111	00979	01559	058	00052	006	00503	00503	009517558)
03(	T0026	W0043	P0044	11111	00000	04627	100	00039	061	04100	00001	009655313)
											Total.Cost	0207366455

Figure 4-9: Fuel Usage

In this example (Figure 4-9), where missions are listed according to the ascending weapon IDs, it can be seen that after mission 03 weapon W0043 has spent all its fuel, and so it could not move for any other mission. However, on the other hand, it can accept all the missions without moving, by staying at the same position (P0044). Here the infeasibility of a possible mission with another position can be said to be because of the position.

4.2.5.3 Time Availability

Each target in the target list has a limited time to be shot. This means that it is meaningless to shoot this target outside of the particular time duration. For instance, target may move to another place or some other operation can be effected by that target, because targets are generally weapons of enemies. In fact, every target should be shot as soon as possible; however, here the limit gives an availability to try different assignments with different times.

Time of a mission is effected by different reasons. For instance, if a slow weapon is to be used and if the position of the mission is far away from the weapon, it cannot catch the required time and shoot even if it is the best weapon for that target. Here, the infeasibility of the combination can be said to be because of the time.

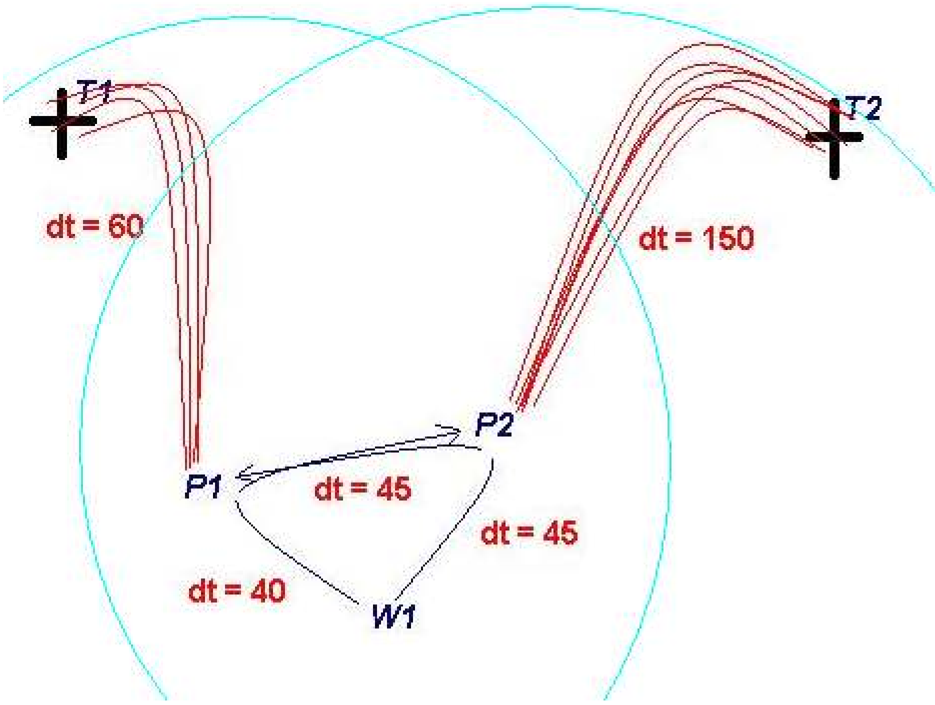


Figure 4-10: Time Schedule of Missions

In this example (Figure 4-10), there exists two targets, T1 and T2, two appropriate position, P1 and P2 and only one weapon, W1. Assume the time limit of target T1 is 120 and time limit of target T2 is 300. For one weapon condition, there are two possible combinations of mission plan. First one is to go to position P1 first to shoot target T1 and then go to position P2 to shoot target T2. Second is vice versa. If time feasibility was not a constraint, there would be no problem. However with this criterion, these mission plans are different from each other.

Table 4-1: Mission List Alternative -1

Mission ID	Target	Weapon	Position	First Shot	Last Shot	Time Availability
0	T1	W1	P1	40	100	100<120 → OK
1	T2	W1	P2	145	295	295<300 → OK

Table 4-2: Mission List Alternative - 2

Mission ID	Target	Weapon	Position	First Shot	Last Shot	Time Availability
2	T2	W1	P2	45	195	195<300 → OK
3	T1	W1	P1	240	300	300>120 → FAIL

As can be seen in the tables, mission 3 is not time available, therefore the second mission list alternative (plan) cannot be selected as a solution.

Here we also see that the background (previous missions/ genes) of each mission affects the feasibility of that mission and each mission is said to be dependent on its previous missions in the mission list.

4.2.5.4 Range Availability

Each weapon in the weapon list has a limited range which differs according to the type of the weapon. This limit can also be the range limit of the ammunition; however for this thesis, maximum ranges of weapons are assumed to be smaller than the limits of the ranges of ammunitions of those weapons. For feasibility check of range, the range of the weapon is compared with the distance for the mission. Here, the distance between the coordinates of the position of the mission and the coordinates of the target of the mission is used.

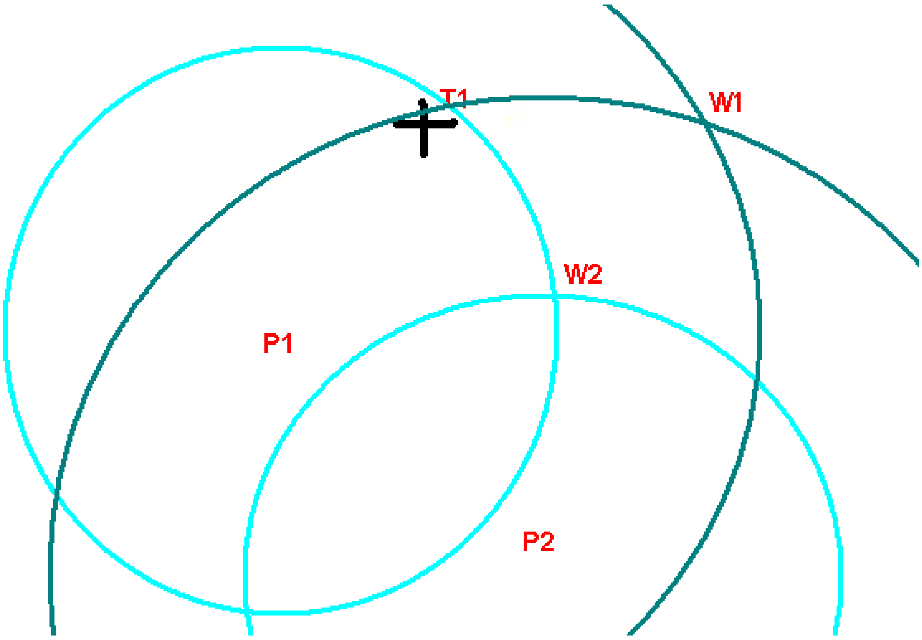


Figure 4-11: Range Availability

In this example (Figure 4-11) the ranges of weapon W1 and weapon W2 from position P1 and P2 are shown and labeled with their names. These ranges are drawn as circles with centers at position P1 and position P2. Therefore, an area of available region is shown for both weapons from both positions. As seen in Figure 4-11, to shoot target T1, weapon W1 can utilize both positions P1 and P2, where weapon W2 can only utilize position P1. An assignment of T1, W2 and P2 is not feasible with respect to the range.

4.2.6 Reproduction

Reproduction (making infeasible chromosomes feasible) of chromosomes for WTPA:

- after a change in the chromosome or,
- for a generated new chromosome

is required to obtain a feasible solution. So chromosomes that fail in feasibility check are reproduced after some operations.

First of all, the reason for infeasibility should be determined before the reproduction. According to the reason, assignment of weapon or position is changed randomly by the algorithm. However, the target of the assignment stays constant since it is aimed that this target to be shot somehow. On the other hand, if the reason is ammunition availability or time availability, weapon is changed. If the reason is fuel availability or range availability, position is changed. After each change a recursive call of feasibility check is done and the new gene is checked for problem constraints. If it is feasible, available gene it is accepted. But if it is not, it is processed again. This loop is allowed to be repeated at most one hundred times for each gene. At the end, if the gene is still infeasible, it is deleted from the chromosome. This process is shown in Figure 4-12.

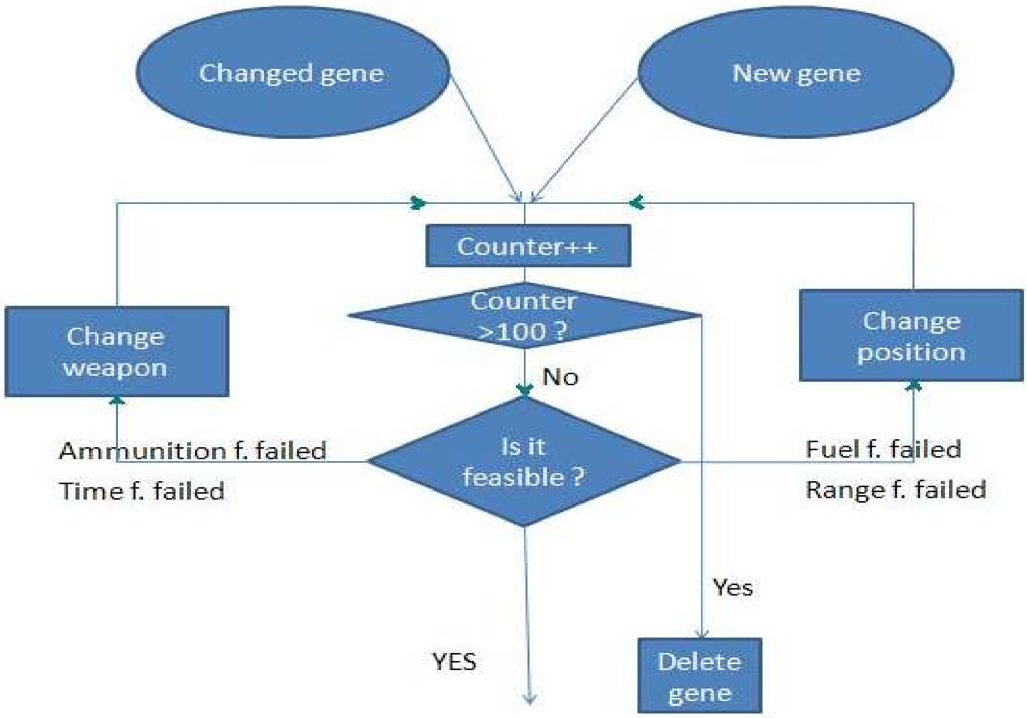


Figure 4-12: Reproduction Operation of WTPA

4.2.7 Elimination

New generations for WTPA are produced by applying the methods mentioned above repeatedly, on the population of the problem. After each generation, the elitist chromosome of the previous population is selected and inserted into the next population to carry the old best solution to the new generation. By this method, the best member of the new generation can never be worse than the previous generation.

4.2.8 Procedure of GA for WTPA

WTPA problem is solved by the GA by combining all the operations above as in Figure 4-13. The algorithms are used in the following order:

1. An initial population is created with one hundred randomly generated chromosomes. Chromosomes may have variable lengths.

2. Every chromosome is checked for feasibility. For this operation every gene of each chromosome is checked and changed if necessary, according to the reproduction algorithm.

3. After reproduction, one hundred feasible chromosomes are obtained and the best one is selected as the elitist member. The elitist chromosome is carried to the new generation to keep track of the best assignment obtained so far.

4. Inserting the elitist member to the new population, each pair of the old population is combined with each other for crossover operation and 100 new chromosomes are generated.

5. These new chromosomes are checked for feasibility and changed if necessary according to the reproduction algorithm.

6. After reproduction, best one of the available chromosomes are selected and the best one is chosen as the elitist member. The elitist chromosome is carried over to the new generation to keep to keep track of the best assignment obtained so far.

7. After inserting the elitist member to the new population, each chromosome of the old population is mutated and another 100 new generated chromosomes are obtained.

8. These new chromosomes are checked for feasibility and changed if necessary according to the reproduction algorithm.

The operations can be performed as shown Figure 4-13 until a stopping criterion is met.

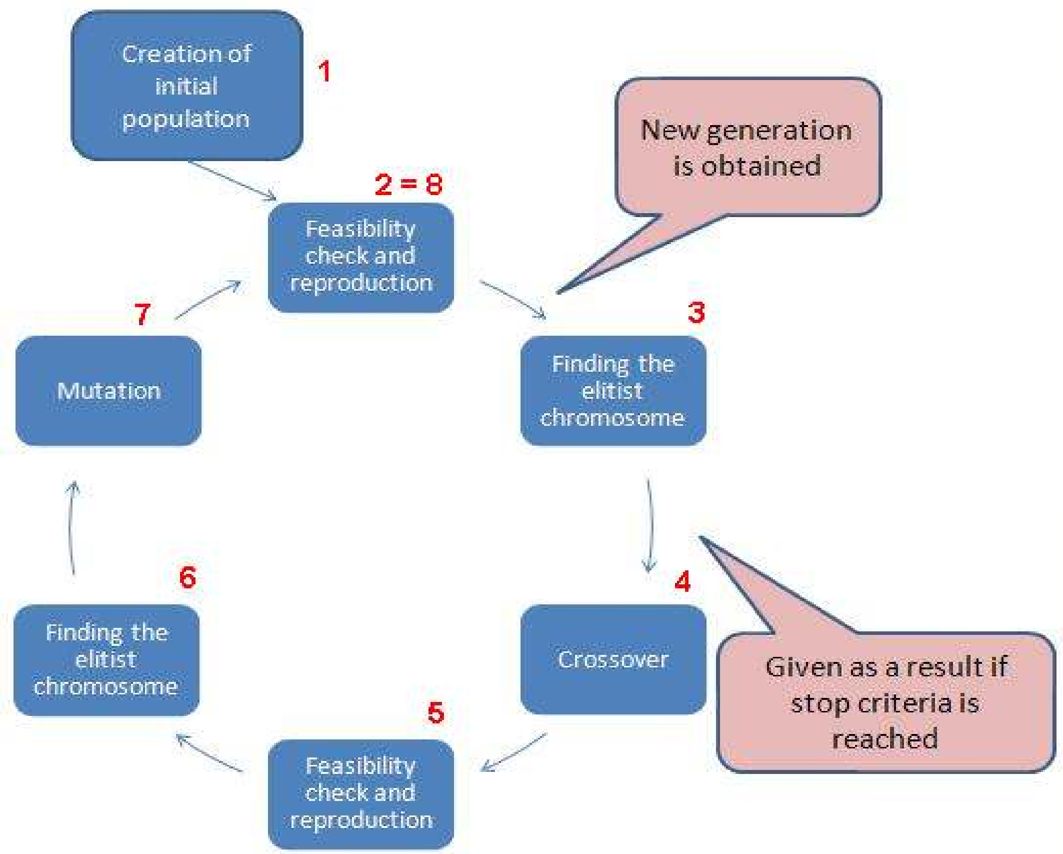


Figure 4-13: GA Cycle of WTPA

On the other hand, the change in the elitist chromosome is scanned in each loop (Figure 4-14). If the elitist member does not improve for some number of loops, a shock is given to the population. Therefore, in Figures 4-13 and 4-14, at step 2, improvement sensor checks the changes in the elitists and affects the algorithm with a new set of initial population injection as a shock. This avoids the algorithm to stack in a bad population.

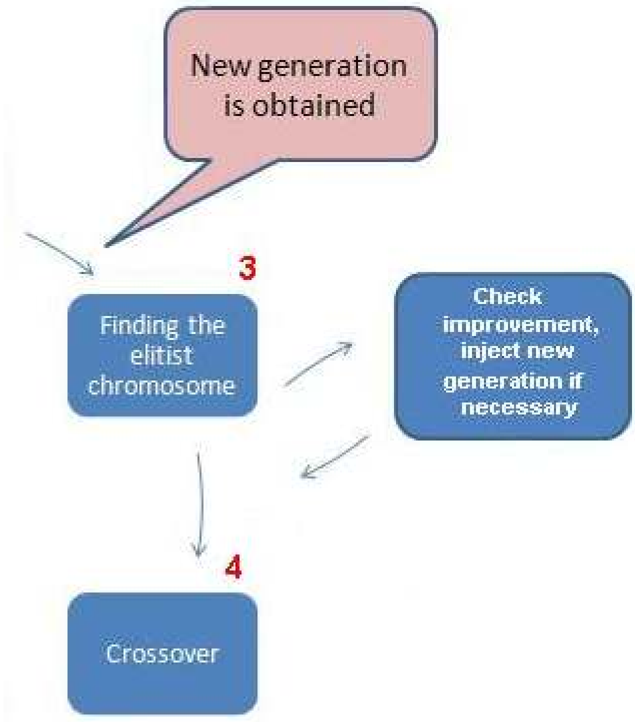


Figure 4-14: Shock Given in GA Cycle of WTPA

4.3 GA with Tabu Search

As explained in “B&B with Tabu Search section”, while searching through the nodes, some of the targets, weapons or positions can be a waste of time. GA selects targets, weapons and positions randomly from a set of all. Therefore, to avoid these kinds of wastes, unused targets, weapons and positions are determined before the search and eliminated from the available target, weapon and position lists. This procedure is called as tabu and the eliminated alternatives forms the tabu list.

To see the effect of tabu method, by inserting random inputs with random numbers of targets (50), weapons (50) and positions (50), following results are obtained (Table 4-3). The columns show the obtained results for the same number of evaluations. It can be seen that without tabu, GA lost time on useless targets, weapons and positions and with tabu, GA used this time to improve its results and obtain a better solution.

Table 4-3: Advantage of Tabu List for GA

GA	Without Tabu	With Tabu
Cost evaluated	414315428	434438275
Missions assigned	41	43

4.4 GA with Greedy or A*

As explained in “B&B with Tabu and Greedy” and “B&B with Tabu, Greedy and A*” sections, while searching through the nodes, the order of target, weapon and position lists are very important for B&B. However, for a random selecting algorithm such as GA, the order of these elements are not important and since sorting procedure will spend evaluation time, greedy and A* methods are useless for GA. In this thesis, although user interface enables user to use these methods together with GA, this combination will not be analyzed.

CHAPTER 5

SIMULATION PROGRAM

The algorithms explained in the previous chapters are implemented in the form of a user friendly simulation program. This program is a software environment (Graphical User Interface (GUI) / database interaction) that is developed with Microsoft Visual C++ 6.0. C++ is selected to improve the run time. All the code for this program is developed as part of this thesis. The developed simulation program and GUI is used to:

- Input and simulate a tactical theatre of operations,
- Apply different WTPA solution algorithms and heuristics,
- Give / output an optimum WTPA list in a reasonable time,
- Give / output detailed information to compare and contrast the algorithms and heuristics.

Using this interface, simulations are run with constant inputs and constant environments representing some particular scenarios. The detailed information about the simulation program GUI and simulation environment is described in the following sections.

5.1 Graphical User Interface (GUI)

The main GUI is given in Figure 5-1.

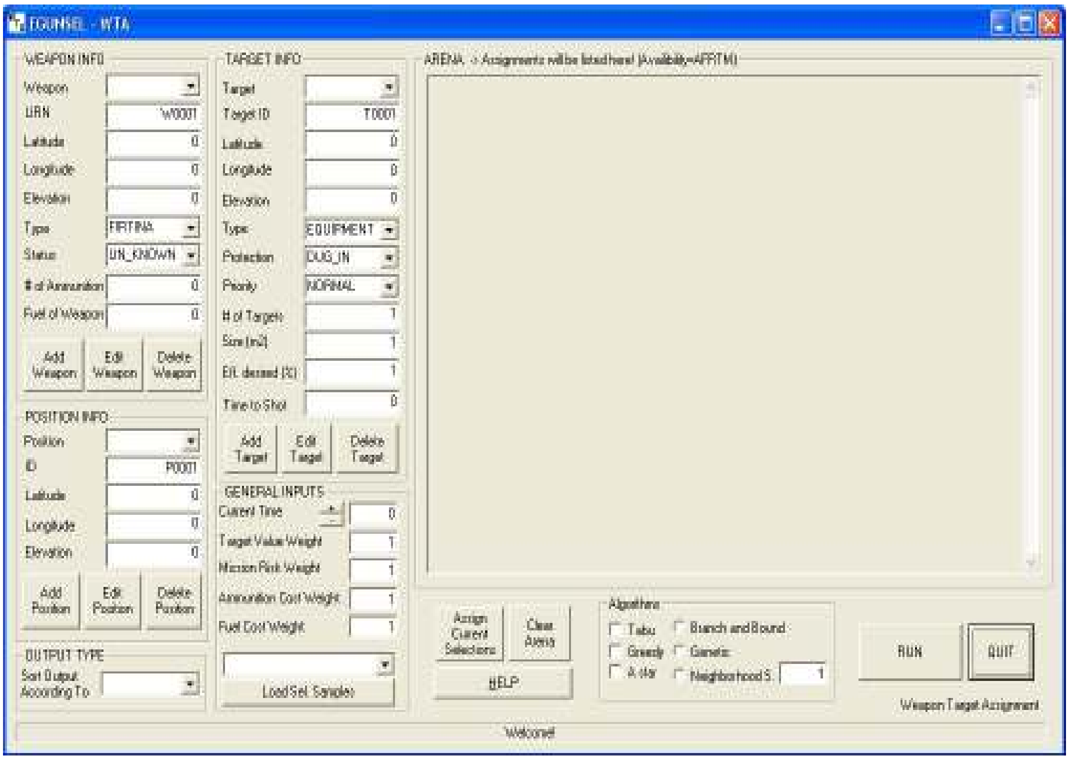


Figure 5-1: Graphical User Interface

In this GUI, the corresponding parts are described in the following sections in detail:

- Weapon Info: Weapons with their corresponding attributes (URN / ID, latitude, longitude, elevation, type, status, number of ammunition, fuel) are chosen in this part. Also, they can be edited or deleted using this part with the related buttons.

- Target Info: Targets with their corresponding attributes (ID, latitude, longitude, elevation, type, protection, and priority, number of targets, size, effects desired, and time to shoot) are chosen in this part. Also, they can be edited or deleted using this part with the related buttons.

- Position Info: Positions with their corresponding attributes (ID, latitude, longitude, elevation) are chosen in this part. Also, they can be edited or deleted using this part with the related buttons.

- General Inputs:
  - Current time: Current time for the problem solution can be entered manually to be able to repeat the problems. It could also be taken from the system clock. This value will be used in the time scheduling / feasibility.
  - Target value weight: The weight of the target value (W_{TV}) in the cost function (33) is entered here.
  - Mission risk weight: The weight of the mission risk (W_{MR}) in the cost function (33) is entered here.
  - Ammunition cost weight: The weight of the ammunition cost (W_{WAC}) in the cost function (33) is entered here.
  - Fuel cost weight: The weight of the fuel cost (W_{FC}) in the cost function (33) is entered here.
  - Load Selected Samples: Enters corresponding pre-defined inputs. This function is used with the combo box together. Some pre-defined inputs are defined in the software. These samples can be used to input many targets, weapons and positions in a very short time and can be used for controlled simulations.

- Arena: Assignments (found after each run) are listed in this part (Figure 5-2).

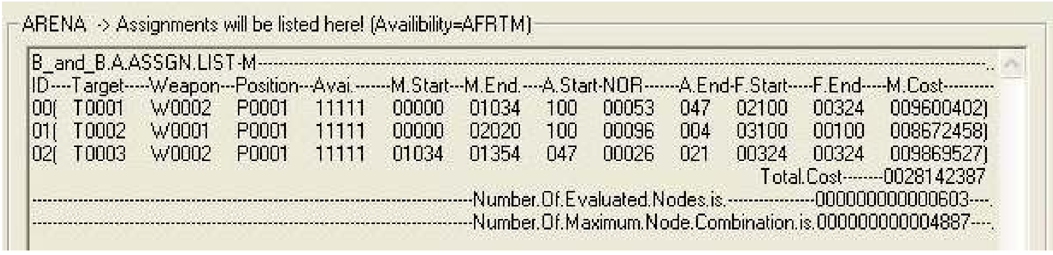


Figure 5-2: Arena (Output of the GUI)

Each column gives a detailed value for the related mission and each row gives the missions.

For each row the following attributes are listed in this area:

- ID : Mission ID,
- Target : Target ID,
- Weapon : Weapon ID,
- Position : Position ID,
- Avai. : Availabilities (ammunition a., fuel a., range a., time a., mission a.) of mission (1=available, 0=not available),
- M.Start :Mission start time,
- M.End : Mission end time,
- A.Start : Ammunition quantity at mission start time,
- NOR : Number of rounds for the mission,
- A.End : Ammunition quantity at mission end time,
- F.Start : Fuel amount at the mission start time,
- F.End : Fuel amount at the mission end time,
- M.Cost : Mission cost.

At the end of the list, Total Cost and other processed information are listed as shown in Figure 5-2.

- Output type: The output of each run is listed in this area. Sorting source is defined by this selection and it can be mission ID, target ID or weapon

ID. Sorting with mission ID shows the exact output. Sorting with target ID shows the assigned targets and can be used to see the situation of each target. Sorting with weapon ID shows the mission lists of each weapon. This list can be used to observe the state of affairs of each weapon.

- Algorithms: The selection of different algorithms (B&B, GA and VNS) can be done in this part. The depth limit to be used in the search can also be entered. Also the heuristics (tabu, greedy and/or A*) to be used can be selected in this part. Each heuristic method can be used with each algorithm practically; however some combinations are meaningless as expected in "GA with Greedy or A*" section.

- Assign current selections button: This button is used to see the situation of an individual mission that is the combination of the target, weapon and position selected from the user interface. By this button, the feasibility and other functional controls can be seen on the area for one individual mission / assignment.

- Clear arena button: This button is used to clear the arena to avoid old and useless data.

- Run button: This button is used to run the selected algorithm and heuristics with the inputs entered. This button will result the algorithm to run and output the best mission list to the arena.

- Help button: This button is used to show some helpful information to the user about the user interface.

- Quit button: This button is used to exit the program.

- Message box: This bar at the bottom of the GUI screen shows

- The last operation done by the user,
- The operation being done at the current time,
- Warnings,
- Errors occurred.

Users should follow the instructions shown in message box.

Some stopping criteria are determined and integrated into the algorithms to take the responses of the users. The aim of this criterion is to ask the user (Figure 5-3) whether to stop or continue on searching after some number of cycles is completed.

Figure 5-3: Stop Criterion Reached

If the user stops the algorithm, the current solution is printed on the screen. If the user wants to go on by pressing the “Go On” button, algorithm goes on until another stopping criterion is reached. The stopping criteria of each algorithm are different from each other.

5.2 Simulation Environment

The simulation environment is a personal computer (Toshiba Satellite A200-1BP) with the following specifications:

- Intel ® Core™2 CPU,
- 997 MHz, 2GB RAM.

CHAPTER 6

SIMULATION STUDIES

In this chapter, the algorithms and methods outlined in the previous chapters, B&B, GA, VNS and additional heuristics are applied to representative combat arenas and results are discussed. Outputs of these algorithms are compared according to the comparison criteria: completeness, optimality, time complexity and space complexity.

The main parameter used for comparison in this study is the optimality and time complexity. However, for each test, all of the following criteria are examined:

- Completeness: Is the algorithm guaranteed to find a solution when there is one?
- Optimality: Does the strategy find the optimal solution?
- Time complexity: How long does it take to find the solution?
- Space Complexity: How much memory is needed [16] to perform the search?

In this chapter:

- In tests 1, 2 and 3 (parts 1, 2 and 3), static WTPA problems with different numbers of elements (target, weapon, position) are simulated and the test results are presented in the form of tables. Comparing these results, GA algorithm is selected to be used.

- In tests 4, 5 and 6 (parts 4, 5 and 6), WTPA problems with different numbers of elements are solved and graphical representations of the results are presented to see the assignments on a map.

- In test 7 and 8 (part 7), dynamic WTPA problems with different numbers of elements are simulated and the test results are presented in the form of tables.

All the tests are run on the same environment and for each test; same inputs are applied to each algorithm to observe the results objectively.

6.1 Test - 1 (3T, 3W, 3P)

Scenario: 3 targets, 3 weapons, 3 positions.

6.1.1 Inputs

Table 6-1: Test-1 Target Inputs

ID	Lat.	Lon.	El.	ED	NOT	TPYE	TPE	TS	TTE	TTS
T0001	330000	331070	330	90	1	2	4	100	14	5000
T0002	330000	330100	330	90	1	1	4	100	14	5000
T0003	330000	330200	330	90	1	2	4	100	13	5000

Table 6-2: Test-1 Weapon Inputs

ID	Lat.	Lon.	Ele.	WT	WS	NA	FOW
W0001	330000	320000	300	9	1	100	3100
W0002	330000	320000	300	10	1	100	2100
W0003	330000	320000	300	2	1	100	1100

Table 6-3: Test-1 Position Inputs

ID	Lat.	Lon.	Ele.
P0001	330100	328000	310
P0002	320200	325000	310
P0003	330300	325000	310

6.1.2 Outputs

Table 6-4: Test-1 B&B Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
Number.Of.Evaluated.Nodes.is.603												
Number.Of.Maximum.Node.Combination.is.4887												
Total.Cost.28142387												

Table 6-5: Test-1 B&B with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
Number.Of.Evaluated.Nodes.is.268												
Number.Of.Maximum.Node.Combination.is.4887												
Total.Cost.28142387												

Table 6-6: Test-1 B&B with Tabu + Greedy Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
Number.Of.Evaluated.Nodes.is.268												
Number.Of.Maximum.Node.Combination.is.4887												
Total.Cost.28142387												

Table 6-7: Test-1 B&B with Tabu + Greedy + A* Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
Number.Of.Evaluated.Nodes.is.240												
Number.Of.Maximum.Node.Combination.is.4887												
Total.Cost.28142387												

Table 6-8: Test-1 GA Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
Number.Of.Evaluated.Cycles.is.10												
Total.Cost.28142387												

Table 6-9: Test-1 GA with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0001	11111	0	1034	100	53	47	2100	324	9600402
1	T0002	W0001	P0001	11111	0	2020	100	96	4	3100	100	8672458
2	T0003	W0002	P0001	11111	1034	1354	47	26	21	324	324	9869527
Number.Of.Evaluated.Cycles.is.10												
Total.Cost.28142387												

Results show that for small number of assignment elements (target, weapon and position), it can be said that all algorithms found the global optimum. For other comparison criteria, each algorithm set is run for 30 generations / iterations and the following results are obtained on the average:

Table 6-10: Test-1 Comparison for 3 Targets, 3 Weapons, 3 Positions

Algorithm	Tabu	Greedy	A*	Completeness	Optimality	Time complexity (CPU User Time)	Space complexity (Virtual Memory Private Bytes)(Kb)	Best Total Cost
B & B	-	-	-	complete	global	18	13	28142387
B & B	used	-	-	complete	global	8	5	28142387
B & B	used	used	-	complete	global	15	5	28142387
B & B	used	used	used	local	global	22	5	28142387
GA	-	-	-	10 cycles	local	345	8391	28142387
GA	used	-	-	10 cycles	local	341	8062	28142387

According to the results, best method is B & B with tabu. This can be explained as:

- Using greedy or greedy + A* method requires extra evaluations and for small number of elements, it is unnecessary.
- Using GA requires a constant number (100) of chromosomes to be generated, crossed over, and mutated, and so on. However this procedure is useless for small number of elements.
- On the other hand, it can be seen that using tabu method decreased the time and space complexity even for GA.

Also, we can say that tabu method has advantages for both B&B and GA.

6.2 Test - 2 (5T, 5W, 5P)

Scenario: 5 targets, 5 weapons, 5 positions.

6.2.1 Inputs

Table 6-11: Test-2 Target Inputs

ID	Lat.	Long.	El.	ED	NOT	TPYE	TPE	TS	TTE	TTS
T0001	325000	330000	330	90	1	2	4	100	14	5000
T0002	325000	325000	330	90	1	2	4	100	14	5000
T0003	325000	320000	330	90	1	2	4	100	13	5000
T0004	325000	315000	330	90	1	2	4	100	12	5000
T0005	325000	320000	330	90	1	2	4	100	11	5000

Table 6-12: Test-2 Weapon Inputs

ID	Lat	Lon.	Ele.	WT	WS	NA	FOW
W0001	319000	320000	300	9	1	300	3100
W0002	319000	320000	300	9	1	300	2100
W0003	319000	320000	300	9	1	300	4100
W0004	319000	320000	300	9	1	300	1000
W0005	319000	320000	300	9	1	300	3200

Table 6-13: Test-2 Weapon Inputs

ID	Lat.	Lon.	Ele.
P0001	320000	320000	310
P0002	320000	325000	310
P0003	315000	325000	310
P0004	320000	315000	310
P0005	315000	320000	310

6.2.2 Outputs

Table 6-14: Test-2 B&B Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0001	P0001	11111	0	1445	300	126	174	3100	2725	8710680
1	T0002	W0001	P0001	11111	1445	2605	174	110	64	2725	2725	8962529
2	T0003	W0001	P0001	11111	2605	3185	64	52	12	2725	2725	9517720
3	T0004	W0002	P0001	11111	0	315	300	13	287	2100	1725	9828109
4	T0005	W0002	P0001	11111	315	505	287	13	274	1725	1725	9875680
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.1219700125												
Total.Cost.46894718												

Table 6-15: Test-2 B&B with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0001	P0001	11111	0	1445	300	126	174	3100	2725	8710680
1	T0002	W0001	P0001	11111	1445	2605	174	110	64	2725	2725	8962529
2	T0003	W0001	P0001	11111	2605	3185	64	52	12	2725	2725	9517720
3	T0004	W0002	P0001	11111	0	315	300	13	287	2100	1725	9828109
4	T0005	W0002	P0001	11111	315	505	287	13	274	1725	1725	9875680
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.1219700125												
Total.Cost.46894718												

Table 6-16: Test-2 B&B with Tabu + Greedy Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0001	P0001	11111	0	1445	300	126	174	3100	2725	8710680
1	T0002	W0001	P0001	11111	1445	2605	174	110	64	2725	2725	8962529
2	T0003	W0001	P0001	11111	2605	3185	64	52	12	2725	2725	9517720
3	T0004	W0003	P0001	11111	0	315	300	13	287	4100	3725	9828109
4	T0005	W0003	P0001	11111	315	505	287	13	274	3725	3725	9875680
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.1219700125												
Total.Cost.46894718												

Table 6-17: Test-2 B&B with Tabu + Greedy + A* Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0001	P0002	11111	0	1797	300	110	190	3100	1189	8730661
1	T0002	W0001	P0002	11111	1797	2897	190	104	86	1189	1189	9040440
2	T0004	W0003	P0001	11111	0	315	300	13	287	4100	3725	9828109
3	T0005	W0003	P0001	11111	315	505	287	13	274	3725	3725	9875680
4	T0003	W0003	P0001	11111	505	1085	274	52	222	3725	3725	9517720
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.1219700125												
Total.Cost.46992610												

Table 6-18: Test-2 GA Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0003	P0002	11111	0	1797	300	110	190	4100	2189	8730661
1	T0003	W0004	P0001	11111	0	705	300	52	248	1000	625	9472220
2	T0005	W0004	P0001	11111	705	895	248	13	235	625	625	9875680
3	T0002	W0003	P0002	11111	1797	2897	190	104	86	2189	2189	9040440
4	T0004	W0003	P0002	11111	2897	3107	86	15	71	2189	2189	9844220
Number.Of.Evaluated.Cycles.is.100 (20000 nodes)												
Number.Of.Maximum.Node.Combination.is.1219700125												
Total.Cost.46963221												

Table 6-19: Test-2 GA with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0003	P0002	11111	0	1797	300	110	190	4100	2189	8730661
1	T0003	W0004	P0001	11111	0	705	300	52	248	1000	625	9472220
2	T0005	W0004	P0001	11111	705	895	248	13	235	625	625	9875680
3	T0002	W0003	P0002	11111	1797	2897	190	104	86	2189	2189	9040440
4	T0004	W0003	P0002	11111	2897	3107	86	15	71	2189	2189	9844220
Number.Of.Evaluated.Cycles.is.100 (20000 nodes)												
Number.Of.Maximum.Node.Combination.is.1219700125												
Total.Cost.46963221												

Results show that for bigger number of assignment elements (target, weapon and position), it can be said that almost every algorithms find different local optimums. For other comparison criteria, each algorithm set is run for 10 times and the following results are obtained on the average:

Table 6-20: Test-2 Comparison for 5 Targets, 5 Weapons, 5 Positions

Algorithm	Tabu	Greedy	A*	Completeness	Optimality	Time complexity (CPU User Time)	Space complexity (Virtual Memory Private Bytes)(Kb)	Best Total Cost
B & B	-	-	-	3000000 nodes	local	7576	18	46894718
B & B	used	-	-	3000000 nodes	local	7603	7	46894718
B & B	used	used	-	3000000 nodes	local	7604	11	46894718
B & B	used	used	used	3000000 nodes	local	7584	11	46992610
B&B	-	-	used	3000000 nodes	local	8778	14	46992610
GA	-	-	-	100 cycles	local	4641	12195	46963221
GA	used	-	-	100 cycles	local	4731	12201	46963221

According to the results, best method is GA without tabu. This can be explained as:

- Using tabu or tabu + greedy method requires extra evaluations and does not deal with this complexity, because the result does not change.
- On the other hand using tabu + greedy + A* or only A* improves the solution, however it causes time and space complexity to increase. Since its time complexity is too high, GA is assumed to be better under these conditions.
- Using GA requires a constant number (100) of chromosomes to be generated, crossed over, mutated and so on, and this procedure is very helpful for bigger number of elements. Therefore, although GA's space complexity is very high compared to B&B, it has a very small time complexity towards a better solution.

- On the other hand, it can be seen that using tabu method did not decreased the time and space complexity even for GA.

We can say that tabu method does not bring advantages for both B&B and GA for this example. To obtain more advantage of the tabu method, more elements (targets, weapons, positions) should be used and many of them should be useless.

6.3 Test - 3 (50T, 50W, 50P)

Scenario: 50 targets, 50 weapons, 50 positions.

6.3.1 Inputs

Table 6-21: Test-3 Target Inputs

ID	Lat.	Long.	El.	ED	NOT	TPYE	TPE	TS	TTE	TTS
T0001	330000	331070	330	90	1	2	4	1	14	9000
T0002	330000	330100	330	90	1	2	4	1	14	9000
T0003	330000	330200	330	90	1	2	4	1	13	5000
T0004	330000	330300	330	90	1	2	4	10	12	9000
T0005	330000	330400	330	90	1	2	4	10	11	5000
T0006	330000	330500	330	90	1	2	4	10	10	5000
T0007	330000	330600	330	90	1	2	4	50	9	5000
T0008	330000	330700	330	90	1	2	4	1	8	9000
T0009	330000	330800	330	90	1	2	4	1	7	9000
T0010	330000	330900	330	90	1	2	4	2	6	9000
T0011	330000	331000	330	90	1	2	4	10	5	5000
T0012	330000	332000	330	90	1	2	4	10	4	5000
T0013	330000	333000	330	90	1	2	4	10	3	5000
T0014	330000	334000	330	90	1	2	4	10	2	5000
T0015	330000	335000	330	90	1	2	4	10	1	5000
T0016	331000	331000	330	90	1	2	4	10	5	5000
T0017	331000	332000	330	90	1	2	4	10	4	5000
T0018	331000	333000	330	90	1	2	4	10	3	5000
T0019	331000	334000	330	90	1	2	4	10	2	5000
T0020	331000	335000	330	90	1	2	4	10	1	5000
T0021	332000	331070	330	90	1	2	4	1	14	9000

Table 6-21 cont'd: Test-3 Target Inputs

ID	Lat.	Long.	El.	ED	NOT	TPYE	TPE	TS	TTE	TTS
T0022	332000	330100	330	90	1	2	4	1	14	9000
T0023	332000	330200	330	90	1	2	4	1	13	9000
T0024	332000	330300	330	90	1	2	4	10	12	9000
T0025	332000	330400	330	90	1	2	4	10	11	9000
T0026	332000	330500	330	90	1	2	4	10	10	5000
T0027	332000	330600	330	90	1	2	4	50	9	5000
T0028	332000	330700	330	90	1	2	4	1	8	5000
T0029	332000	330800	330	90	1	2	4	1	7	5000
T0030	332000	330900	330	90	1	2	4	2	6	5000
T0031	333000	331070	330	90	1	2	4	1	14	5000
T0032	333000	330100	330	90	1	2	4	1	14	5000
T0033	333000	330200	330	90	1	2	4	100	13	5000
T0034	333000	330300	330	90	1	2	4	500	12	5000
T0035	333000	330400	330	90	1	2	4	10	11	5000
T0036	333000	330500	330	90	1	2	4	10	10	5000
T0037	333000	330600	330	90	1	2	4	50	9	5000
T0038	333000	330700	330	90	1	2	4	1	8	5000
T0039	333000	330800	330	90	1	2	4	1	7	5000
T0040	333000	330900	330	90	1	2	4	2	6	5000
T0041	334000	331070	330	90	1	2	4	1	14	5000
T0042	334000	330100	330	90	1	2	4	1	14	5000
T0043	334000	330200	330	90	1	2	4	1	13	5000
T0044	334000	330300	330	90	1	2	4	10	12	5000
T0045	334000	330400	330	90	1	2	4	10	11	5000
T0046	334000	330500	330	90	1	2	4	10	10	5000
T0047	334000	330600	330	90	1	2	4	50	9	5000
T0048	334000	330700	330	90	1	2	4	100	8	5000
T0049	334000	330800	330	90	1	2	4	100	7	5000
T0050	334000	330900	330	90	1	2	4	200	6	5000

Table 6-22: Test-3 Weapon Inputs

ID	Lat	Lon.	Ele.	WT	WS	NA	FOW
W0001	300000	300000	300	0	1	100	3100
W0002	300000	300000	300	1	1	100	2100
W0003	300000	300000	300	2	1	100	4100
W0004	300000	300000	300	3	1	100	1000
W0005	300000	300000	300	4	1	100	3200
W0006	300000	300000	300	5	1	100	4100
W0007	300000	300000	300	6	1	100	3100
W0008	300000	300000	300	7	1	100	2100

Table 6-22 cont'd: Test-3 Weapon Inputs

ID	Lat	Lon.	Ele.	WT	WS	NA	FOW
W0009	300000	300000	300	8	1	100	1100
W0010	300000	300000	300	9	1	100	1100
W0011	300000	300000	300	0	1	100	3100
W0012	300000	310000	300	1	1	100	2100
W0013	300000	310000	300	2	1	100	4100
W0014	300000	310000	300	3	1	100	1000
W0015	300000	310000	300	4	1	100	3200
W0016	300000	310000	300	5	1	100	4100
W0017	300000	310000	300	6	1	100	3100
W0018	300000	310000	300	7	1	100	2100
W0019	300000	310000	300	8	1	100	1100
W0020	300000	310000	300	9	1	100	1100
W0021	300000	320000	300	0	1	100	3100
W0022	300000	320000	300	1	1	100	2100
W0023	300000	320000	300	2	1	100	4100
W0024	300000	320000	300	3	1	100	1000
W0025	300000	320000	300	4	1	100	3200
W0026	300000	320000	300	5	1	100	4100
W0027	300000	320000	300	6	1	100	3100
W0028	300000	320000	300	7	1	100	2100
W0029	300000	320000	300	8	1	100	1100
W0030	300000	320000	300	9	1	100	1100
W0031	300000	330000	300	0	1	100	3100
W0032	300000	330000	300	1	1	100	2100
W0033	300000	330000	300	2	1	100	4100
W0034	300000	330000	300	3	1	100	1000
W0035	300000	330000	300	4	1	100	3200
W0036	300000	330000	300	5	1	100	4100
W0037	300000	330000	300	6	1	100	3100
W0038	300000	330000	300	7	1	100	2100
W0039	300000	330000	300	8	1	100	1100
W0040	300000	330000	300	9	1	100	1100
W0041	310000	330000	300	0	1	100	3100
W0042	310000	330000	300	1	1	100	2100
W0043	310000	330000	300	2	1	100	4100
W0044	310000	330000	300	3	1	100	1000
W0045	310000	330000	300	4	1	100	3200
W0046	310000	330000	300	5	1	100	4100
W0047	310000	330000	300	6	1	100	3100
W0048	310000	330000	300	7	1	100	2100
W0049	310000	330000	300	8	1	100	1100
W0050	310000	330000	300	9	1	100	1100

Table 6-23: Test-3 Position Inputs

ID	Lat.	Lon.	Ele.
P0001	330100	325000	310
P0002	320200	325000	310
P0003	330300	325000	310
P0004	330400	325000	310
P0005	330500	325000	310
P0006	330600	325000	310
P0007	330700	325000	310
P0008	330800	325000	310
P0009	330900	325000	310
P0010	331400	325000	310
P0011	330100	335000	310
P0012	320200	335000	310
P0013	330300	335000	310
P0014	330400	335000	310
P0015	330500	335000	310
P0016	330600	335000	310
P0017	330700	335000	310
P0018	330800	335000	310
P0019	330900	335000	310
P0020	331400	335000	310
P0021	330100	330000	310
P0022	320200	330000	310
P0023	330300	330000	310
P0024	330400	330000	310
P0025	330500	330000	310

ID	Lat.	Lon.	Ele.
P0026	330600	330000	310
P0027	330700	330000	310
P0028	330800	330000	310
P0029	330900	330000	310
P0030	331400	330000	310
P0031	330100	331000	310
P0032	320200	331000	310
P0033	330300	331000	310
P0034	330400	331000	310
P0035	330500	331000	310
P0036	330600	331000	310
P0037	330700	331000	310
P0038	330800	331000	310
P0039	330900	331000	310
P0040	331400	331000	310
P0041	330100	332000	310
P0042	320200	332000	310
P0043	330300	332000	310
P0044	330400	332000	310
P0045	330500	332000	310
P0046	330600	332000	310
P0047	330700	332000	310
P0048	330800	332000	310
P0049	330900	332000	310
P0050	331400	332000	310

6.3.2 Outputs

Table 6-24: Test-3 B&B Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0014	11111	0	240	100	10	90	2100	2100	10244123
1	T0002	W0002	P0011	11111	240	576	90	12	78	2100	1980	10210723
2	T0003	W0002	P0011	11111	576	744	78	6	72	1980	1980	10106828
3	T0004	W0002	P0011	11111	744	1074	72	15	57	1980	1980	9873021
4	T0005	W0002	P0011	11111	1074	1386	57	14	43	1980	1980	9884167
5	T0006	W0002	P0011	11111	1386	1968	43	29	14	1980	1980	9764389
6	T0007	W0005	P0014	11111	0	7260	100	60	40	3200	3200	9243207
7	T0008	W0002	P0031	11111	1968	3044	14	12	2	1980	380	10544464
8	T0009	W0003	P0014	11111	0	576	100	43	57	4100	4100	10350380
9	T0010	W0003	P0021	11111	576	2045	57	34	23	4100	3099	10383239
10	T0011	W0004	P0014	11111	0	2070	100	67	33	1000	1000	9534178
11	T0012	W0006	P0014	11111	0	3360	100	55	45	4100	4100	9667331
12	T0013	W0008	P0014	11111	0	3420	100	56	44	2100	2100	9979121
13	T0014	W0007	P0014	11111	0	3660	100	60	40	3100	3100	9753943
14	T0015	W0009	P0014	11111	0	3960	100	65	35	1100	1100	9868438
15	T0016	W0006	P0011	11111	3360	6157	45	45	0	4100	3952	9712919
16	T0017	W0007	P0033	11111	3660	6620	40	40	0	3100	1100	9814023
17	T0018	W0010	P0014	11111	0	180	100	12	88	1100	1100	10296232
18	T0019	W0008	P0011	11111	3420	5418	44	32	12	2100	2028	10000078
19	T0020	W0010	P0011	11111	180	387	88	11	77	1100	989	10296592
20	T0021	W0003	P0021	11111	2045	2285	23	15	8	3099	3099	10323124
21	T0022	W0004	P0011	11111	2070	2797	33	21	12	1000	889	10166569
22	T0023	W0002	P0039	11111	3044	3300	2	2	0	380	60	10129346
23	T0024	W0005	P0011	11111	7260	8790	40	12	28	3200	3080	9880426
24	T0025	W0005	P0011	11111	8790	10290	28	12	16	3080	3080	9881572
25	T0026	W0009	P0011	11111	3960	5058	35	17	18	1100	1028	9933732
26	T0027	W0010	P0011	11111	387	507	77	6	71	989	989	9964368
27	T0028	W0010	P0011	11111	507	587	71	2	69	989	989	10780019
28	T0029	W0010	P0011	11111	587	657	69	1	68	989	989	10432751
29	T0030	W0010	P0011	11111	657	747	68	3	65	989	989	10497562
30	T0031	W0010	P0011	11111	747	817	65	1	64	989	989	10342476
31	T0032	W0010	P0011	11111	817	887	64	1	63	989	989	10341667

Table 6-24 cont'd: Test-3 B&B Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
32	T0033	W0010	P0011	11111	887	1467	63	52	11	989	989	9517112
33	T0035	W0010	P0011	11111	1467	1537	11	1	10	989	989	10026923
34	T0036	W0010	P0011	11111	1537	1617	10	2	8	989	989	10059367
35	T0037	W0010	P0011	11111	1617	1737	8	6	2	989	989	9963891
36	T0038	W0010	P0011	11111	1737	1817	2	2	0	989	989	10779534
37	T0039	W0043	P0021	11111	0	4380	100	25	75	4100	80	10047912
38	T0041	W0043	P0021	11111	4380	4812	75	31	44	80	80	10285174
39	T0042	W0043	P0025	11111	4812	5240	44	24	20	80	0	10294144
40	T0043	W0003	P0030	11111	2285	2701	8	8	0	3099	2839	10133848
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.out of integer bounds												
Total.Cost.413308913												

Table 6-25: Test-3 B&B with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0014	11111	0	240	100	10	90	2100	2100	10244123
1	T0002	W0002	P0014	11111	240	516	90	12	78	2100	2100	10220684
2	T0003	W0002	P0014	11111	516	684	78	6	72	2100	2100	10106805
3	T0004	W0002	P0014	11111	684	1014	72	15	57	2100	2100	9872997
4	T0005	W0002	P0014	11111	1014	1326	57	14	43	2100	2100	9884143
5	T0006	W0002	P0014	11111	1326	1908	43	29	14	2100	2100	9764365
6	T0007	W0005	P0014	11111	0	7260	100	60	40	3200	3200	9243207
7	T0008	W0002	P0021	11111	1908	3185	14	12	2	2100	98	10510432
8	T0009	W0003	P0014	11111	0	576	100	43	57	4100	4100	10350380
9	T0010	W0003	P0021	11111	576	2045	57	34	23	4100	3099	10383239
10	T0011	W0004	P0014	11111	0	2070	100	67	33	1000	1000	9534178
11	T0012	W0006	P0014	11111	0	3360	100	55	45	4100	4100	9667331
12	T0013	W0008	P0014	11111	0	3420	100	56	44	2100	2100	9979121
13	T0014	W0007	P0014	11111	0	3660	100	60	40	3100	3100	9753943
14	T0015	W0009	P0014	11111	0	3960	100	65	35	1100	1100	9868438
15	T0016	W0006	P0014	11111	3360	6120	45	45	0	4100	4100	9714367

6-25 cont'd: Test-3 B&B with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
16	T0017	W0007	P0033	11111	3660	6620	40	40	0	3100	1100	9814023
17	T0018	W0010	P0014	11111	0	180	100	12	88	1100	1100	10296232
18	T0019	W0008	P0014	11111	3420	5400	44	32	12	2100	2100	10000926
19	T0020	W0010	P0014	11111	180	350	88	11	77	1100	1100	10310360
20	T0021	W0003	P0021	11111	2045	2285	23	15	8	3099	3099	10323124
21	T0022	W0004	P0014	11111	2070	2760	33	21	12	1000	1000	10167789
22	T0023	W0003	P0021	11111	2285	2417	8	6	2	3099	3099	10162136
23	T0024	W0005	P0014	11111	7260	8760	40	12	28	3200	3200	9881581
24	T0025	W0005	P0014	11111	8760	10260	28	12	16	3200	3200	9881652
25	T0026	W0009	P0014	11111	3960	5040	35	17	18	1100	1100	9934396
26	T0027	W0010	P0014	11111	350	470	77	6	71	1100	1100	9964479
27	T0028	W0010	P0014	11111	470	550	71	2	69	1100	1100	10780132
28	T0029	W0010	P0014	11111	550	620	69	1	68	1100	1100	10432866
29	T0030	W0010	P0014	11111	620	710	68	3	65	1100	1100	10497679
30	T0031	W0010	P0014	11111	710	780	65	1	64	1100	1100	10342648
31	T0032	W0010	P0014	11111	780	850	64	1	63	1100	1100	10341813
32	T0033	W0010	P0014	11111	850	1430	63	52	11	1100	1100	9517262
33	T0035	W0010	P0014	11111	1430	1500	11	1	10	1100	1100	10027077
34	T0036	W0010	P0014	11111	1500	1580	10	2	8	1100	1100	10059523
35	T0037	W0010	P0014	11111	1580	1700	8	6	2	1100	1100	9964050
36	T0038	W0010	P0014	11111	1700	1780	2	2	0	1100	1100	10779696
37	T0039	W0043	P0021	11111	0	4380	100	25	75	4100	80	10047912
38	T0042	W0043	P0025	11111	4380	4808	75	24	51	80	0	10294144
39	T0043	W0043	P0025	11111	4808	5000	51	11	40	0	0	10147239
40	T0041	W0043	P0025	11111	5000	5372	40	26	14	0	0	10296441
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.out of integer bounds												
Total.Cost.413362933												

Table 6-26: Test-3 B&B with Tabu + Greedy Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0033	W0010	P0015	11111	0	580	100	52	48	1100	1100	9517308
1	T0013	W0010	P0015	11111	580	760	48	12	36	1100	1100	10296259
2	T0018	W0010	P0015	11111	760	940	36	12	24	1100	1100	10296259
3	T0020	W0010	P0015	11111	940	1110	24	11	13	1100	1100	10310460
4	T0015	W0010	P0015	11111	1110	1280	13	11	2	1100	1100	10310460
5	T0047	W0008	P0015	11111	0	1980	100	32	68	2100	2100	9772584
6	T0037	W0008	P0015	11111	1980	3900	68	31	37	2100	2100	9781387
7	T0027	W0043	P0023	11111	0	5188	100	89	11	4100	40	9513689
8	T0007	W0008	P0015	11111	3900	5760	37	30	7	2100	2100	9790242
9	T0014	W0009	P0015	11111	0	2520	100	41	59	1100	1100	9902520
10	T0019	W0009	P0015	11111	2520	5040	59	41	18	1100	1100	9902520
11	T0011	W0002	P0015	11111	0	744	100	38	62	2100	2100	9711137
12	T0012	W0002	P0015	11111	744	1542	62	41	21	2100	2100	9724636
13	T0006	W0010	P0015	11111	1280	1360	2	2	0	1100	1100	10060193
14	T0016	W0002	P0033	11111	1542	2780	21	21	0	2100	500	9769931
15	T0017	W0007	P0015	11111	0	2760	100	45	55	3100	3100	9791283
16	T0026	W0007	P0015	11111	2760	4320	55	25	30	3100	3100	9871756
17	T0036	W0007	P0015	11111	4320	5880	30	25	5	3100	3100	9871453
18	T0038	W0005	P0015	11111	0	2700	100	22	78	3200	3200	10513495
19	T0040	W0005	P0015	11111	2700	6240	78	29	49	3200	3200	10150467
20	T0042	W0006	P0015	11111	0	900	100	14	86	4100	4100	10225565
21	T0044	W0006	P0015	11111	900	1980	86	17	69	4100	4100	9882845
22	T0045	W0006	P0015	11111	1980	3060	69	17	52	4100	4100	9882905
23	T0046	W0006	P0015	11111	3060	5220	52	35	17	4100	4100	9760925
24	T0030	W0004	P0015	11111	0	1740	100	56	44	1000	1000	10036815
25	T0032	W0004	P0015	11111	1740	2460	44	22	22	1000	1000	10158535
26	T0043	W0004	P0015	11111	2460	2850	22	11	11	1000	1000	10076078
27	T0010	W0009	P0033	11111	5040	6250	18	15	3	1100	100	10397532
28	T0035	W0003	P0018	11111	0	1188	100	94	6	4100	4100	9839598
29	T0001	W0008	P0015	11111	5760	6060	7	4	3	2100	2100	10323505
30	T0002	W0043	P0023	11111	5188	5368	11	10	1	40	40	10338828
31	T0003	W0004	P0015	11111	2850	3180	11	9	2	1000	1000	10095007
32	T0004	W0005	P0015	11111	6240	7740	49	12	37	3200	3200	9881760
33	T0008	W0005	P0015	11111	7740	10320	37	21	16	3200	3200	10527286
34	T0009	W0006	P0031	11111	5220	6382	17	10	7	4100	2092	10337956
35	T0023	W0009	P0033	11111	6250	6430	3	2	1	100	100	10160707

Table 6-26 cont'd: Test-3 B&B with Tabu + Greedy Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.out of integer bounds												
Total.Cost.360783886												

Table 6-27: Test-3 B&B with Tabu + Greedy + A* Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0033	W0010	P0015	11111	0	580	100	52	48	1100	1100	9517308
1	T0013	W0010	P0015	11111	580	760	48	12	36	1100	1100	10296259
2	T0018	W0010	P0015	11111	760	940	36	12	24	1100	1100	10296259
3	T0020	W0010	P0015	11111	940	1110	24	11	13	1100	1100	10310460
4	T0015	W0010	P0015	11111	1110	1280	13	11	2	1100	1100	10310460
5	T0047	W0008	P0015	11111	0	1980	100	32	68	2100	2100	9772584
6	T0037	W0008	P0015	11111	1980	3900	68	31	37	2100	2100	9781387
7	T0027	W0043	P0023	11111	0	5188	100	89	11	4100	40	9513689
8	T0007	W0008	P0015	11111	3900	5760	37	30	7	2100	2100	9790242
9	T0014	W0009	P0015	11111	0	2520	100	41	59	1100	1100	9902520
10	T0019	W0009	P0015	11111	2520	5040	59	41	18	1100	1100	9902520
11	T0011	W0002	P0015	11111	0	744	100	38	62	2100	2100	9711137
12	T0012	W0002	P0015	11111	744	1542	62	41	21	2100	2100	9724636
13	T0006	W0010	P0015	11111	1280	1360	2	2	0	1100	1100	10060193
14	T0016	W0002	P0033	11111	1542	2780	21	21	0	2100	500	9769931
15	T0017	W0007	P0015	11111	0	2760	100	45	55	3100	3100	9791283
16	T0026	W0007	P0015	11111	2760	4320	55	25	30	3100	3100	9871756
17	T0036	W0007	P0015	11111	4320	5880	30	25	5	3100	3100	9871453
18	T0038	W0005	P0015	11111	0	2700	100	22	78	3200	3200	10513495
19	T0040	W0005	P0015	11111	2700	6240	78	29	49	3200	3200	10150467
20	T0042	W0006	P0015	11111	0	900	100	14	86	4100	4100	10225565
21	T0044	W0006	P0015	11111	900	1980	86	17	69	4100	4100	9882845
22	T0045	W0006	P0015	11111	1980	3060	69	17	52	4100	4100	9882905
23	T0046	W0006	P0015	11111	3060	5220	52	35	17	4100	4100	9760925
24	T0030	W0004	P0015	11111	0	1740	100	56	44	1000	1000	10036815

Table 6-27 cont'd: Test-3 B&B with Tabu + Greedy + A* Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
25	T0032	W0004	P0015	11111	1740	2460	44	22	22	1000	1000	10158535
26	T0043	W0004	P0015	11111	2460	2850	22	11	11	1000	1000	10076078
27	T0010	W0009	P0033	11111	5040	6250	18	15	3	1100	100	10397532
28	T0035	W0003	P0018	11111	0	1188	100	94	6	4100	4100	9839598
29	T0001	W0008	P0015	11111	5760	6060	7	4	3	2100	2100	10323505
30	T0002	W0043	P0023	11111	5188	5368	11	10	1	40	40	10338828
31	T0003	W0004	P0015	11111	2850	3180	11	9	2	1000	1000	10095007
32	T0004	W0005	P0015	11111	6240	7740	49	12	37	3200	3200	9881760
33	T0008	W0005	P0015	11111	7740	10320	37	21	16	3200	3200	10527286
34	T0009	W0006	P0031	11111	5220	6382	17	10	7	4100	2092	10337956
35	T0023	W0009	P0033	11111	6250	6430	3	2	1	100	100	10160707
Number.Of.Evaluated.Nodes.is.3000000												
Number.Of.Maximum.Node.Combination.is.out of integer bounds												
Total.Cost.360783886												

Table 6-28: Test-3 GA Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0019	W0008	P0019	11111	0	1980	100	32	68	2100	2100	10001047
1	T0012	W0010	P0016	11111	0	100	100	4	96	1100	1100	10126381
2	T0009	W0004	P0016	11111	0	720	100	22	78	1000	1000	10249793
3	T0017	W0010	P0011	11111	100	262	96	4	92	1100	914	10103740
4	T0044	W0010	P0014	11111	262	369	92	1	91	914	803	10012972
5	T0025	W0004	P0020	11111	720	1600	78	24	54	1000	700	9823556
6	T0001	W0009	P0014	11111	0	420	100	6	94	1100	1100	10303383
7	T0046	W0009	P0035	11111	420	1630	94	15	79	1100	100	9945118
8	T0022	W0010	P0017	11111	369	476	91	1	90	803	692	10328823
9	T0007	W0006	P0016	11111	0	4740	100	78	22	4100	4100	9317830
10	T0011	W0009	P0033	11111	1630	2842	79	19	60	100	52	9964479
11	T0015	W0010	P0020	11111	476	733	90	11	79	692	431	10277892
12	T0039	W0008	P0018	11111	1980	2406	68	6	62	2100	2076	10395920
13	T0004	W0006	P0027	11111	4740	6085	22	11	11	4100	1600	9918600

Table 6-28 cont'd: Test-3 GA Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
14	T0023	W0003	P0014	11111	0	480	100	35	65	4100	4100	10094039
15	T0045	W0008	P0020	11111	2406	2863	62	6	56	2076	1928	9989283
16	T0037	W0010	P0020	11111	733	853	79	6	73	431	431	9964479
17	T0028	W0007	P0020	11111	0	1320	100	21	79	3100	3100	10626408
18	T0016	W0007	P0018	11111	1320	3555	79	35	44	3100	2800	9828201
19	T0042	W0002	P0019	11111	0	366	100	17	83	2100	2100	10164385
20	T0026	W0002	P0035	11111	366	1517	83	16	67	2100	494	9778087
21	T0036	W0010	P0018	11111	853	1008	73	2	71	431	206	10032411
22	T0043	W0004	P0018	11111	1600	2065	54	11	43	700	475	10073981
23	T0021	W0005	P0015	11111	0	1140	100	9	91	3200	3200	10237074
24	T0003	W0002	P0035	11111	1517	1613	67	2	65	494	494	10156596
25	T0029	W0010	P0016	11111	1008	1103	71	1	70	206	131	10423833
26	T0006	W0008	P0018	11111	2863	3680	56	12	44	1928	1780	9984530
27	T0040	W0008	P0046	11111	3680	4707	44	13	31	1780	1032	10421643
28	T0035	W0043	P0024	11111	0	4404	100	22	78	4100	20	9645002
29	T0008	W0002	P0038	11111	1613	1949	65	12	53	494	374	10666793
30	T0031	W0008	P0013	11111	4707	5195	31	4	27	1032	280	10315438
31	T0041	W0004	P0016	11111	2065	2780	43	21	22	475	400	10166991
32	T0018	W0010	P0018	11111	1103	1298	70	11	59	131	56	10299850
33	T0032	W0010	P0018	11111	1298	1368	59	1	58	56	56	10341989
34	T0002	W0004	P0020	11111	2780	3540	22	20	2	400	100	10173729
35	T0013	W0010	P0018	11111	1368	1548	58	12	46	56	56	10296166
36	T0030	W0003	P0021	11111	480	2081	65	45	20	4100	3099	10357401
37	T0024	W0007	P0016	11111	3555	4360	44	12	32	2800	2700	9937194
38	T0047	W0005	P0036	11111	1140	8440	91	57	34	3200	1600	9269517
39	T0005	W0003	P0021	11111	2081	2297	20	13	7	3099	3099	10017473
40	T0010	W0007	P0020	11111	4360	6260	32	29	3	2700	2300	10283401
Number.Of.Evaluated.Cycles.is.100 (20000 nodes)												
Number.Of.Maximum.Node.Combination.is.out of integer bounds												
Total.Cost.414315428												

Table 6-29: Test-3 GA with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0002	P0020	11111	0	240	100	10	90	2100	2100	10243792
1	T0029	W0006	P0020	11111	0	960	100	15	85	4100	4100	10307619
2	T0004	W0010	P0019	11111	0	70	100	1	99	1100	1100	10027575
3	T0035	W0008	P0020	11111	0	420	100	6	94	2100	2100	9991061
4	T0019	W0009	P0018	11111	0	2460	100	40	60	1100	1100	9911535
5	T0043	W0010	P0014	11111	70	202	99	1	98	1100	914	10138792
6	T0040	W0010	P0018	11111	202	342	98	3	95	914	764	10479228
7	T0018	W0010	P0019	11111	342	524	95	11	84	764	728	10304590
8	T0042	W0008	P0018	11111	420	817	94	5	89	2100	1952	10312237
9	T0014	W0010	P0017	11111	524	679	84	7	77	728	653	10171200
10	T0032	W0043	P0034	11111	0	4360	100	18	82	4100	16	9967387
11	T0009	W0009	P0031	11111	2460	3133	60	6	54	1100	88	10387987
12	T0025	W0004	P0014	11111	0	810	100	25	75	1000	1000	9817443
13	T0021	W0010	P0016	11111	679	761	77	1	76	653	617	10338821
14	T0028	W0010	P0017	11111	761	853	76	2	74	617	581	10775860
15	T0005	W0003	P0014	11111	0	756	100	58	42	4100	4100	9913850
16	T0006	W0006	P0016	11111	960	2980	85	31	54	4100	3700	9795116
17	T0046	W0008	P0033	11111	817	1788	89	11	78	1952	948	9985046
18	T0012	W0010	P0019	11111	853	978	74	4	70	581	506	10117208
19	T0026	W0006	P0037	11111	2980	4980	54	24	30	3700	1700	9847236
20	T0016	W0010	P0015	11111	978	1118	70	3	67	506	356	10074849
21	T0031	W0010	P0019	11111	1118	1238	67	1	66	356	206	10324705
22	T0002	W0005	P0017	11111	0	1140	100	9	91	3200	3200	10236517
23	T0045	W0007	P0015	11111	0	840	100	13	87	3100	3100	9929028
24	T0022	W0007	P0048	11111	840	1756	87	8	79	3100	1596	10273400
25	T0038	W0007	P0030	11111	1756	3157	79	18	61	1596	552	10642861
26	T0047	W0008	P0037	11111	1788	3613	78	29	49	948	848	9798459
27	T0033	W0010	P0016	11111	1238	1855	66	52	14	206	95	9503886
28	T0041	W0043	P0034	11111	4360	4720	82	25	57	16	16	10298640
29	T0015	W0010	P0017	11111	1855	2037	14	11	3	95	59	10305892
30	T0003	W0005	P0031	11111	1140	1964	91	3	88	3200	1584	10124969
31	T0039	W0010	P0016	11111	2037	2119	3	1	2	59	23	10428155
32	T0023	W0006	P0027	11111	4980	5405	30	4	26	1700	1200	10137144
33	T0020	W0002	P0015	11111	240	1722	90	69	21	2100	1740	9660630
34	T0008	W0004	P0020	11111	810	2255	75	42	33	1000	625	10427582
35	T0007	W0009	P0034	11111	3133	5131	54	32	22	88	16	9757834

Table 6-29 cont'd: Test-3 GA with Tabu Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
36	T0024	W0006	P0023	11111	5405	6235	26	12	14	1200	1000	9930446
37	T0030	W0003	P0048	11111	756	1901	42	40	2	4100	3495	10402674
38	T0044	W0002	P0017	11111	1722	2200	21	21	0	1740	1660	9798832
39	T0036	W0005	P0040	11111	1964	4674	88	21	67	1584	1064	9804608
40	T0011	W0005	P0050	11111	4674	8554	67	31	36	1064	664	9717572
41	T0037	W0008	P0037	11111	3613	5353	49	28	21	848	848	9808575
42	T0010	W0005	P0048	11111	8554	11554	36	24	12	664	424	10217434
Number.Of.Evaluated.Cycles.is.100 (20000 nodes)												
Number.Of.Maximum.Node.Combination.is.out of integer bounds												
Total.Cost.434438275												

Results show that for a more realistic number of assignment elements (target, weapon and position), it can be said that almost every algorithm find different local optimums. For other comparison criteria, each algorithm set is run for 10 times and the following results are obtained on the average:

Table 6-30: Test-3(a) Comp. for 50 Targets, 50 Weapons, 50 Positions

Algorithm	Tabu	Greedy	A*	Completeness	Optimality	Time complexity (CPU User Time)(msec)	Space complexity (Virtual Memory Private Bytes)(Kb)	Best Total Cost	# of assigned targets
B & B	-	-	-	3000000 nodes	local	9197	511	413308913	41
B & B	used	-	-	3000000 nodes	local	9248	796	413362933	41
B & B	used	used	-	3000000 nodes	local	9242	845	360783886	36
B & B	used	used	used	3000000 nodes	local	9212	863	360783886	36
GA	-	-	-	100 cycles	local	7525	83464	414315428	41
GA	used	-	-	100 cycles	local	7530	83502	434438275	43

According to the results, best method is GA with tabu. This can be explained as:

- Using tabu / greedy or A* requires extra evaluations and does not deal with this complexity, because they do not provide sufficient improvement, as expected. So, extra tests are done with different input sets. Results are listed in Table 6-31 and Table 6-32:

Table 6-31: Test-3(b) Comp. for 50 Targets, 50 Weapons, 50 Positions

Algorithm	Tabu	Greedy	A*	Best Total Cost	# of assigned targets
B & B	-	-	-	477045437	47
B & B	used	-	-	475392216	47
B & B	used	used	-	479403412	47
B & B	used	used	used	479403412	47
GA	-	-	-	475385367	47
GA	used	-	-	475385367	47

Table 6-32: Test-3(c) Comp. for 50 Targets, 50 Weapons, 50 Positions

Algorithm	Tabu	Greedy	A*	Best Total Cost	# of assigned targets
B & B	-	-	-	226005214	23
B & B	used	-	-	225982321	23
B & B	used	used	-	227913804	23
B & B	used	used	used	227913804	23
GA	-	-	-	227181628	23
GA	used	-	-	227275047	23

According to the results (Table 6-30, Table 6-31, Table 6-32), it can be said that greedy and A* results are not reliable and their performance is dependent on the input data set. So these methods are not robust to be used for uncompleted searches.

- Using GA requires a constant number (100) of chromosomes to be generated, crossed over, mutated and so on, and this procedure is very helpful for bigger number of elements. Therefore, although GA's space complexity is very high compared to B&B, it has a smaller time complexity towards a better solution.
- Additionally, it can be seen that using tabu method also increases the time, space complexity and the total gain for GA. This caused two more targets to be shot. Since increasing the gain is the most important criterion, greedy with tabu is selected to be the best method.

6.4 Test - 4 (7T, 7W, 7P, Graphical)

Scenario: 7 targets, 7 weapons, 7 positions are assigned using B&B with tabu search and graphical results are given below:

6.4.1 Inputs

Table 6-33: Test-4 Target Inputs

	Latitude	Longitude
T1	300500	301000
T2	300500	303000
T3	301750	302000
T4	302500	304500
T5	301000	306000
T6	300000	308000
T7	304000	307500

Table 6-34: Test-4 Weapon Inputs

	Latitude	Longitude
W1	317500	305000
W2	317500	305500
W3	317000	310000
W4	317000	310500
W5	316500	301500
W6	316500	302000
W7	316500	302500

Table 6-35: Test-4 Position Inputs

	Latitude	Longitude
P1	306000	308000
P2	304500	305000
P3	312000	306000
P4	312500	310000
P5	312500	312000
P6	305500	302000
P7	316000	306000

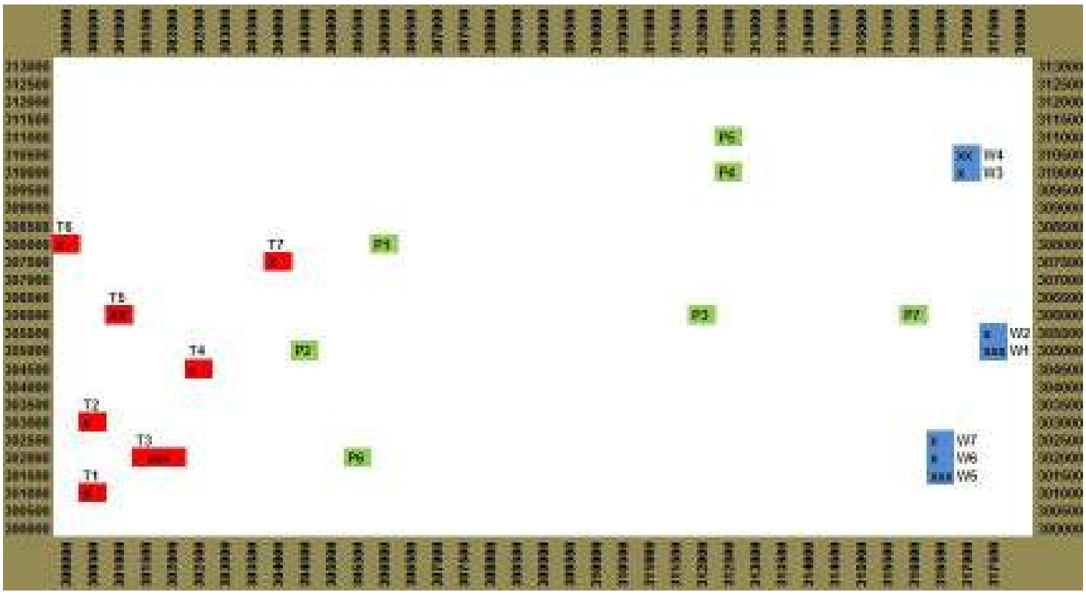


Figure 6-1: Test 4 – Input Arena

In Figure 6-1, a typical input is simulated to visualize the problem inputs. Weapons are blue squares, possible positions are green ones and the targets are red squares in the Figure 6-1.

6.4.2 Outputs

Table 6-36: Test-4 Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	T0001	W0007	P0001	11111	0	4340	100	59	41	5100	2140	9577462
1	T0002	W0002	P0002	11111	0	3657	100	83	17	9100	6499	9705120
2	T0003	W0003	P0002	11111	0	3724	100	81	19	9100	6408	9774621
3	T0004	W0003	P0002	11111	3724	4000	19	18	1	6408	6408	9970914
4	T0005	W0007	P0001	11111	4340	4760	41	6	35	2140	2140	9959175
5	T0006	W0004	P0001	11111	0	3090	100	27	73	9000	3360	9765832
6	T0007	W0007	P0001	11111	4760	6440	35	27	8	2140	2140	9785662

In Figure 6-2, a typical output is simulated to visualize the problem outputs. In this problem, the assignments are as follows:

- Weapon W0002 will move to position P0002 and shoot target T0002 until time is 3657 and hide in an appropriate position then.
- Weapon W0003 will move to position P0002 and shoot target T0003 until time is 3724 and go on shooting target T0004 with 18 ammunitions until time is 4000.
- Weapon W0004 will move to position P0001 and shoot target T0006 until time is 3090.
- Weapon W0007 will move to position P0001 and shoot target T0001 until time is 4340 and go on shooting target T0005 with 6 ammunitions until time is 4760. Then, weapon W0007 will go on shooting target T0007 until time is 6440.

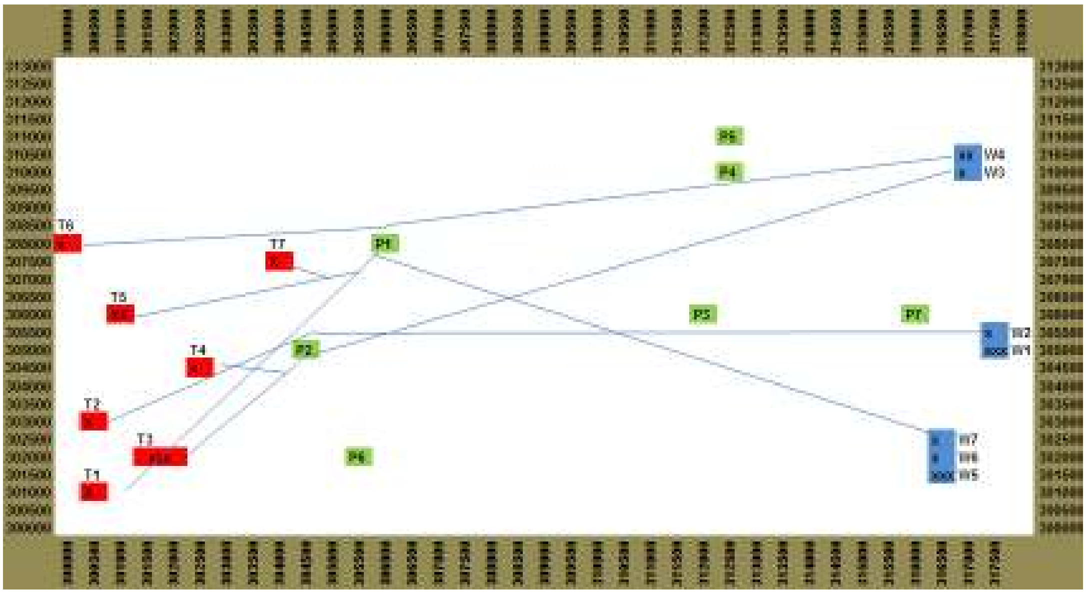


Figure 6-2: Test 4 – Output Arena

6.5 Test – 5 (10T, 20W, 50P, Graphical)

Scenario: 10 targets, 20 weapons, 50 positions are assigned using B&B and GA with tabu search and graphical results are given below:

6.5.1 Inputs

Table 6-37: Test-5 Target Inputs

	Latitude	Longitude
T1	300500	301000
T2	300500	303000
T3	301750	302000
T4	302500	304500
T5	301000	306000
T6	300000	308000
T7	304000	307500
T8	302250	309500
T9	300500	311000
T10	301500	312000

Table 6-38: Test-5 Weapon Inputs

	Latitude	Longitude
W1	317500	305000
W2	317500	305500
W3	317000	310000
W4	317000	310500
W5	316500	301500
W6	316500	302000
W7	316500	302500
W8	314000	311500
W9	314500	310000
W10	314000	309000

	Latitude	Longitude
W11	315500	308500
W12	315000	307000
W13	314500	305500
W14	313500	304500
W15	314500	303000
W16	315000	302000
W17	313000	301000
W18	311500	305500
W19	311500	302500
W20	311500	301000

Table 6-39: Test-5 Position Inputs

	Latitude	Longitude
P1	306500	311500
P2	307000	310000
P3	306500	308500
P4	307500	308000
P5	306500	307000
P6	307500	306000
P7	306000	305000
P8	307500	304000
P9	306500	303000
P10	307500	301500
P11	310000	311000
P12	311000	310000
P13	310000	308000
P14	311000	307000
P15	310000	306000
P16	311500	304000
P17	310500	302500
P18	311000	301000
P19	312500	312500
P20	313000	312000
P21	312500	311500
P22	313500	311000
P23	312500	310500
P24	313000	309500
P25	312500	308500

	Latitude	Longitude
P26	313500	308000
P27	312500	307000
P28	313500	306500
P29	313000	305000
P30	313500	304000
P31	313000	303000
P32	314000	301500
P33	315500	312000
P34	314500	311000
P35	315500	310000
P36	315000	308500
P37	316500	307500
P38	315500	306000
P39	316500	305500
P40	315500	304500
P41	316500	304000
P42	315000	303500
P43	316000	302500
P44	316000	301000
P45	316500	311500
P46	317500	309500
P47	317500	308000
P48	318000	307500
P49	317500	302500
P50	318000	301500

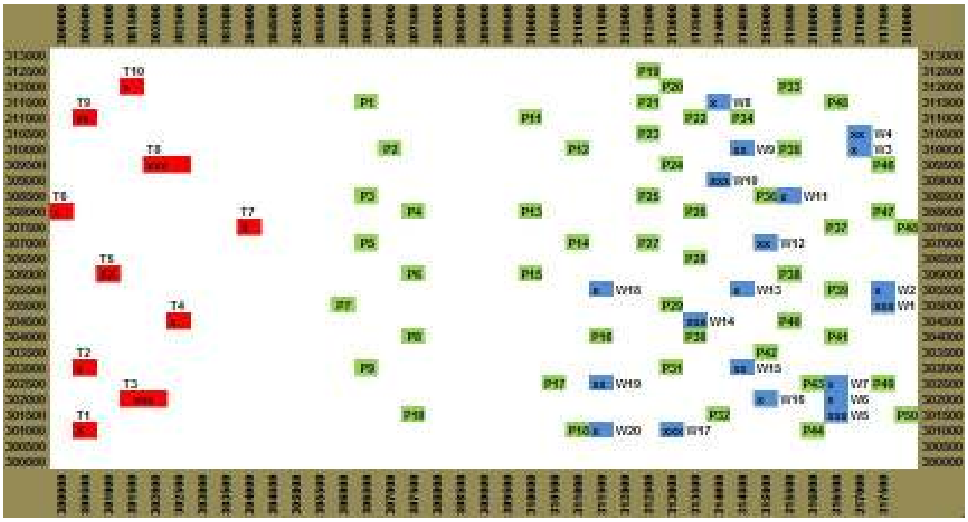


Figure 6-3: Test 5 – Input Arena

In Figure 6-3, a typical input is simulated to imagine the problem inputs. Weapons are blue squares, possible positions are green ones and the targets are red squares in the Figure 6-3.

6.5.2 Outputs

In Figure 6-4, a random output of GA is simulated to imagine the problem outputs. In this problem, the assignments obtained by GA with tabu search are as follows:

Table 6-40: Test-5 GA Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
7	T0007	W0008	P0022	11111	0	2444	100	39	61	7100	6924	9677292
8	T0008	W0008	P0008	11111	2444	6380	61	55	6	6924	4620	9690470
4	T0005	W0012	P0021	11111	0	2263	100	26	74	7100	4528	9752936
0	T0010	W0014	P0011	11111	0	840	100	37	63	7000	5360	9976116
1	T0002	W0014	P0016	11111	840	1307	63	1	62	5360	3772	9973592
9	T0009	W0014	P0016	11111	1307	1637	62	27	35	3772	3772	10119522
5	T0006	W0018	P0008	11111	0	1167	100	14	86	2100	1032	9928691

Table 6-40 cont'd: Test-5 GA Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
2	T0004	W0019	P0022	11111	0	2711	100	26	74	7100	2736	9733594
3	T0001	W0020	P0016	11111	0	555	100	12	88	9100	7975	9808419
6	T0003	W0020	P0009	11111	555	1272	88	2	86	7975	6064	9917998

- Weapon W0008 will move to position P0022 and shoot target T0007 until time is 2444 and without waiting, move to position P0008 and shoot T0008, then hide in an appropriate position.
- Weapon W0012 will move to position P0021 and shoot target T0005 until time is 2263 and hide in an appropriate position then.
- Weapon W0014 will move to position P0011 and shoot T0010 first. W0014 will then move to P0016 to shoot T0002 and without waiting, shoot target T0009 until time is 1637. At last, it will hide in an appropriate position.
- Weapon W0018 will move to position P0008 and shoot target T0006 with 14 ammunitions until time is 1167.
- Weapon W0019 will move to position P0022 and shoot target T0004 with 26 ammunitions until time is 2711.
- Weapon W0020 will move to position P0016 and shoot target T0001 until time is 555 and without waiting move to position P0009 and shoot T0003, then hide in an appropriate position.

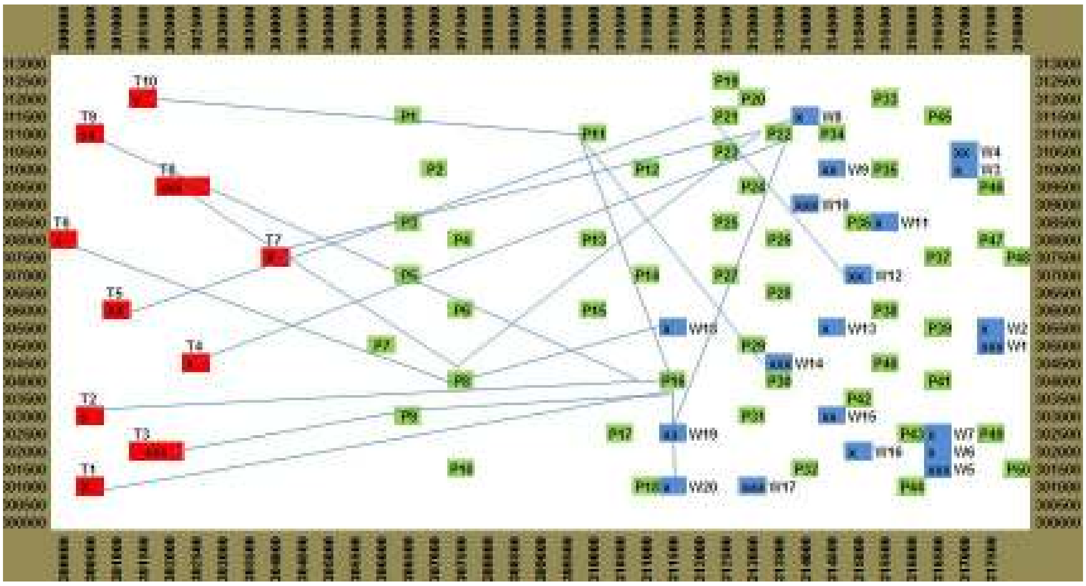


Figure 6-4: Test 5 – GA Output Arena

In Figure 6-5, a typical output of B&B is simulated to imagine the problem outputs. In this problem, the assignments obtained by B&B with tabu search are as follows:

Table 6-41: Test-5 B&B Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
3	T0004	W0002	P0005	11111	0	3036	100	63	37	9100	6880	9682794
4	T0005	W0003	P0007	11111	0	3604	100	94	6	9100	6684	9602255
1	T0002	W0004	P0001	11111	0	3238	100	31	69	9000	3728	9777146
2	T0003	W0004	P0001	11111	3238	5698	69	40	29	3728	3728	9828535
6	T0007	W0006	P0006	11111	0	3109	100	60	40	4100	162	9020437
0	T0001	W0007	P0001	11111	0	5040	100	69	31	5100	1740	9487288
5	T0006	W0008	P0001	11111	0	1308	100	13	87	7100	5228	9929934
7	T0008	W0008	P0003	11111	1308	4435	87	48	39	5228	4480	9767005
8	T0010	W0014	P0002	11111	0	873	100	34	66	7000	5108	9995648
9	T0009	W0014	P0002	11111	873	1153	66	22	44	5108	5108	10183757

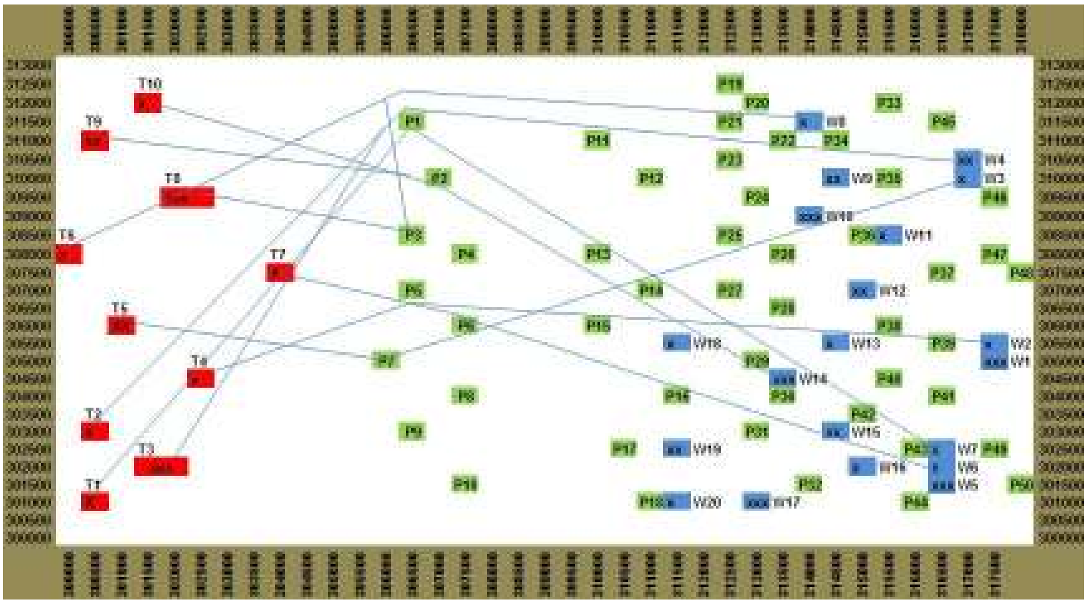


Figure 6-5: Test 5 – B&B Output Arena

6.6 Test – 6 (30T, 20W, 10P, Graphical)

Scenario: 30 targets, 20 weapons, 10 positions are assigned using GA with tabu search and graphical results are given below:

6.6.1 Inputs

Table 6-42: Test-6 Target Inputs

	Latitude	Longitude
T1	300500	301000
T2	300500	303000
T3	301750	302000
T4	302500	304500
T5	301000	306000
T6	300000	308000
T7	304000	307500
T8	302250	309500
T9	300500	311000
T10	301500	312000
T11	303000	312500

	Latitude	Longitude
T16	304500	309500
T17	305000	306500
T18	303000	308500
T19	301500	308000
T20	302000	307000
T21	303000	307000
T22	304000	305500
T23	301500	304500
T24	303500	304500
T25	303000	303000
T26	304500	303000

Table 6-42 Cont'd: Test-6 Target Inputs

	Latitude	Longitude
T12	304500	312000
T13	303500	311000
T14	304500	310500
T15	303500	310000

	Latitude	Longitude
T27	302500	301000
T28	303500	301500
T29	304500	300500
T30	306000	301500

Table 6-43: Test-6 Weapon Inputs

	Latitude	Longitude
W1	317500	305000
W2	317500	305500
W3	317000	310000
W4	317000	310500
W5	316500	301500
W6	316500	302000
W7	316500	302500
W8	314000	311500
W9	314500	310000
W10	314000	309000

	Latitude	Longitude
W11	315500	308500
W12	315000	307000
W13	314500	305500
W14	313500	304500
W15	314500	303000
W16	315000	302000
W17	313000	301000
W18	311500	305500
W19	311500	302500
W20	311500	301000

Table 6-44: Test-6 Position Inputs

	Latitude	Longitude
P1	306500	311500
P2	307000	310000
P3	306500	308500
P4	307500	308000
P5	306500	307000
P6	307500	306000
P7	306000	305000
P8	307500	304000
P9	306500	303000
P10	307500	301500

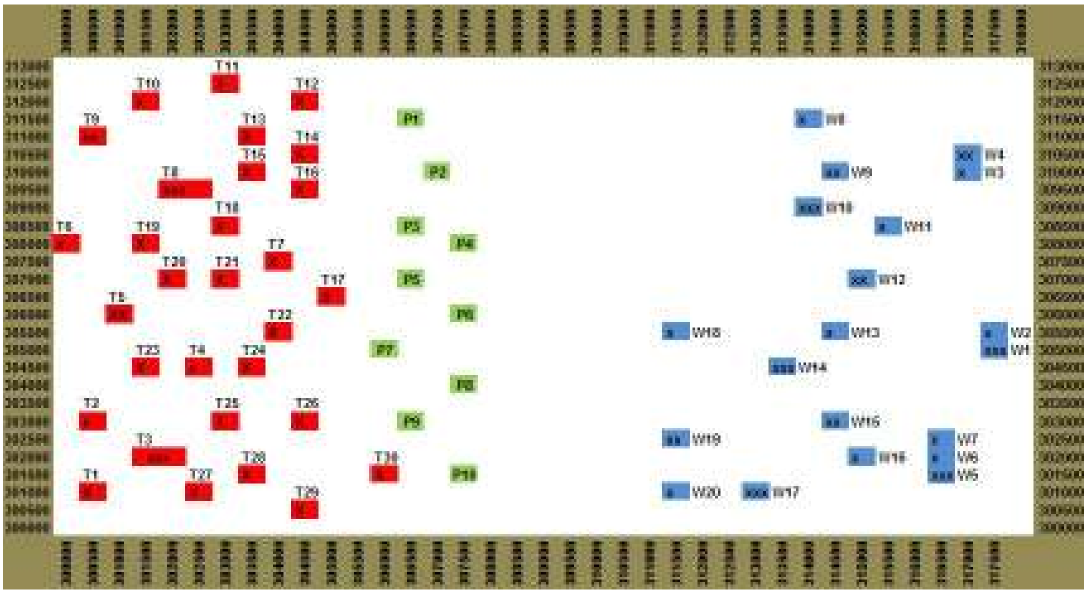


Figure 6-6: Test 6 – Input Arena

In Figure 6-6, a typical input is simulated to imagine the problem inputs. Weapons are blue squares, possible positions are green ones and the targets are red squares in the Figure 6-6.

6.6.2 Outputs

Table 6-45: Test-6 Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
1	**T0016**	**W0007**	**P0010**	**11111**	**0**	**1885**	**100**	**21**	**79**	5100	2840	9930163
10	**T0024**	**W0007**	**P0010**	**11111**	**1885**	**4945**	**79**	**50**	**29**	2840	2840	9622188
19	**T0002**	**W0007**	**P0010**	**11111**	**4945**	**5605**	**29**	**10**	**19**	2840	2840	10018820
6	**T0011**	**W0008**	**P0010**	**11111**	**0**	**2305**	**100**	**25**	**75**	7100	4120	9886989
9	**T0027**	**W0008**	**P0010**	**11111**	**2305**	**3085**	**75**	**12**	**63**	4120	4120	9940657
18	**T0017**	**W0008**	**P0010**	**11111**	**3085**	**4645**	**63**	**25**	**38**	4120	4120	9966277
20	**T0019**	**W0008**	**P0007**	**11111**	**4645**	**7222**	**38**	**38**	**0**	4120	3172	9938051
13	**T0005**	**W0012**	**P0010**	**11111**	**0**	**2122**	**100**	**15**	**85**	7100	2452	9828649
8	**T0012**	**W0013**	**P0001**	**11111**	**0**	**2972**	**100**	**76**	**24**	4100	2100	9849715

Table 6-45 Cont'd: Test-6 Results

ID	Target	Weapon	Position	Avai.	M.Start	M.End	A.Start	NOR	A.End	F.Start	F.End	M.Cost
0	**T0003**	**W0014**	**P0010**	**11111**	**0**	**442**	**100**	**1**	**99**	7000	5512	10056741
2	**T0022**	**W0014**	**P0010**	**11111**	**442**	**612**	**99**	**11**	**88**	5512	5512	10225359
4	**T0021**	**W0014**	**P0010**	**11111**	**612**	**852**	**88**	**18**	**70**	5512	5512	10140871
7	**T0023**	**W0014**	**P0010**	**11111**	**852**	**1082**	**70**	**17**	**53**	5512	5512	9973047
11	**T0030**	**W0014**	**P0010**	**11111**	**1082**	**1292**	**53**	**15**	**38**	5512	5512	10360709
17	**T0025**	**W0014**	**P0010**	**11111**	**1292**	**1362**	**38**	**1**	**37**	5512	5512	10029566
21	**T0026**	**W0014**	**P0010**	**11111**	**1362**	**1432**	**37**	**1**	**36**	5512	5512	10075608
23	**T0014**	**W0014**	**P0002**	**11111**	**1432**	**1995**	**36**	**3**	**33**	5512	3620	10096337
25	**T0015**	**W0014**	**P0001**	**11111**	**1995**	**2202**	**33**	**6**	**27**	3620	3272	10351182
26	**T0006**	**W0014**	**P0009**	**11111**	**2202**	**2744**	**27**	**1**	**26**	3272	1384	9909142
27	**T0029**	**W0014**	**P0009**	**11111**	**2744**	**3064**	**26**	**26**	**0**	1384	1384	10138778
5	**T0007**	**W0018**	**P0002**	**11111**	**0**	**2197**	**100**	**29**	**71**	2100	512	9751760
24	**T0020**	**W0018**	**P0003**	**11111**	**2197**	**6075**	**71**	**62**	**9**	512	120	9922917
3	**T0018**	**W0020**	**P0010**	**11111**	**0**	**703**	**100**	**14**	**86**	9100	7591	10081627
12	**T0009**	**W0020**	**P0010**	**11111**	**703**	**1243**	**86**	**48**	**38**	7591	7591	9831480
14	**T0004**	**W0020**	**P0010**	**11111**	**1243**	**1313**	**38**	**1**	**37**	7591	7591	9995030
15	**T0008**	**W0020**	**P0010**	**11111**	**1313**	**1483**	**37**	**11**	**26**	7591	7591	10031392
16	**T0001**	**W0020**	**P0010**	**11111**	**1483**	**1653**	**26**	**11**	**15**	7591	7591	9961943
22	**T0013**	**W0020**	**P0010**	**11111**	**1653**	**1863**	**15**	**15**	**0**	7591	7591	10250093

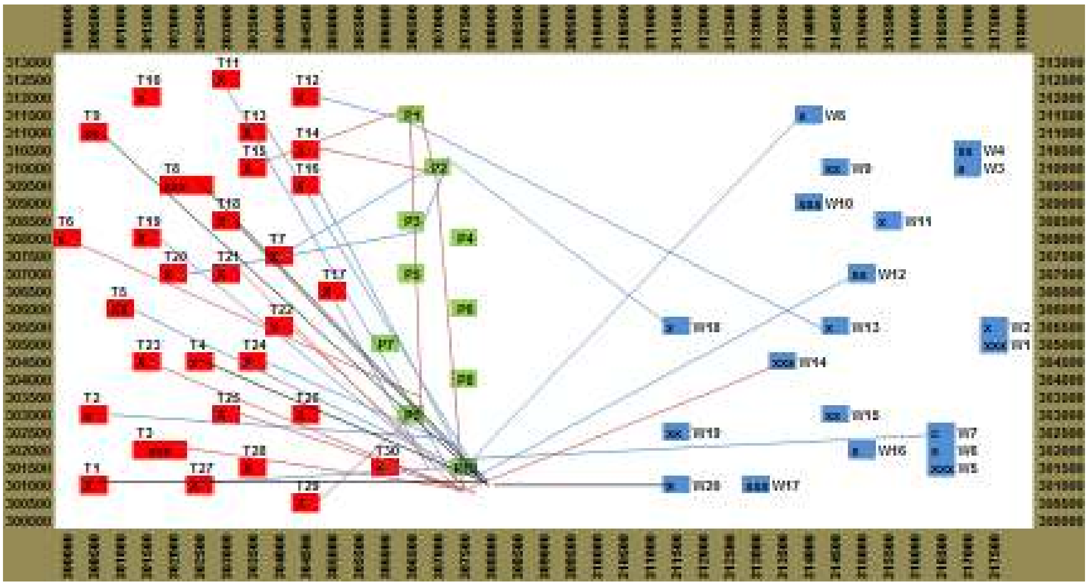


Figure 6-7: Test 6 – Output Arena

6.7 Tests on Dynamic WTPA

In this study, different kinds of heuristic methods and optimization algorithms are used to solve WTPA for two different versions of the WTPA problem: static and dynamic.

In the static version, all the inputs to the problem are fixed; that is, all targets are known, all weapons are known, and all weapons engage targets according to the information available at the instant problem is started to be solved. In the previous tests, the static WTPA problem is studied. After the tests, GA with tabu search is accepted to be the best option for static weapon target position assignment problem.

The dynamic version of the problem is a multi-stage problem where some weapons are engaged at the targets at the static state. The outcome of this engagement is assessed and strategy for the next stages is decided. The static WTPA solution can be assumed to be the initial point for dynamic WTPA. In this thesis, to solve the dynamic WTPA problem, VNS and GA are used, because VNS and GA can start with the previous solutions / populations as an initial starting point, easily. However B&B cannot start with an initial point in a straightforward manner. For B&B, the change in inputs can affect the current searched space and this results in a need to search them all over again and this causes a static search again.

To simulate dynamic WTPA, static WTPA problems are solved by GA and the results are given to the competitor algorithms with some changes in input. These changes are increase in time ($t=t_o+100$) and changes in weapon statuses (W0002 broken).

In test-7, results of VNS and GA will be compared.

Table 6-46: Test-7 Dynamic WTPA Comparisons (3T, 3W, 3P)

Prob (T,W,P)	Algorithm	Tabu	Greedy	A*	Completeness	Optimality	Time complexity (CPU User Time) (msec)	Space complexity (Virtual Memory Private Bytes) (Kb)	Best Total Cost (28142387 for initial)	# of assigned targets (3 for initial)
3,3,3	NS (dist. 1)	-	-	-	Local (21 nodes)	local	6	32	28142387	3
3,3,3	NS (dist. 2)	-	-	-	Local (454 nodes)	local	81	645	28142387	3
3,3,3	NS (dist. 3)	-	-	-	Local (9395 nodes)	local	863	13398	28142387	3
3,3,3	GA	-	-	-	100 cycles	local	3184	76069	28142387	3
3,3,3	GA	used	-	-	100 cycles	local	3255	76099	28142387	3

According to results listed in Table 6-46, for a 3 targets, 3 weapons and 3 positions problem, no improvement is obtained. This is expected, because the initial results were the global optimum since it is obtained by a complete search. However comparing the time complexities, NS (distance 1) can be assumed to be the best with a 6 m.sec average evaluation time. Using NS (distance 1) iteratively (by repeating) can produce the best result.

To simulate dynamic WTPA, static WTPA problems are solved by GA as quickly as possible (decreased number of generations, decreased population size, etc.) and the results are given to the competitor algorithms with some changes in input. These changes are increase in time ($t=t_o+100$) and changes in target inputs (T0051 is added).

Table 6-47: Test-8 Dynamic WTPA Comparisons (50T, 50W, 50P)

Prob (T,W,P)	Algorithm	Tabu	Greedy	A*	Completeness	Optimality	Time complexity (CPU User Time) (msec)	Space complexity (Virtual Memory Private Bytes) (Kb)	Best Total Cost (403142373 for initial)	# of assigned targets (39 for initial)
50,50,50	NS (dist. 1)	used	-	-	local	local	8172	5148	414769040	41
50,50,50	GA	-	-	-	100 cycles	local	7525	83464	424737354	41
50,50,50	GA	used	-	-	100 cycles	local	7530	83502	424757104	41

NS with depth 2 is not applicable because large number of nodes to be searched makes the evaluation too long for dynamic solutions. However comparing the time complexities for other methods, GA without tabu can be assumed to be the best with a 7525 m.sec average evaluation time. On the other hand, NS (distance 1) has a close time complexity and improvements in the gain with a lesser space complexity and with minimum change in the assignment list (changes in the assignment list means extra communication and coordination complexity in fire support). According to results listed in Table 6-47 and discussions, best improvement is obtained by NS (distance 1).

CHAPTER 7

SUMMARY AND CONCLUSION

In this thesis, the mathematical formulation of the problem of WTPA for fire support automation systems is made first. According to this formulation, the problem is converted to an optimization problem and is solved by several techniques. Different search algorithms and heuristics (mainly, B&B method, GA and VNS methods) are discussed among the possible solution techniques. Results of B&B method, GA and VNS algorithms are compared. Different heuristics such as tabu search, greedy search and A* search are tested and simulated. The results are classified according to the complexity of the problems.

In this thesis, it is also observed that weapon target assignment problem of fire support automation systems is a very complex problem, increasing the number of assignment elements increases the solution time exponentially and causes the complete search to become impossible (in the required time).

Smaller problems such as WTPA with 3 positions, 3 targets and 3 weapons are to be solved best with B&B. On the other hand, larger problems such as WTA with 50 positions, 50 targets and 50 weapons should be solved with GA with tabu search, because of the time complexity advantage. Since the larger problems are more realistic in real application areas, GA with tabu search is decided to be the best method to solve static WTPA in this thesis.

For the dynamic assignment processes, B&B is not preferred because of the time constraint of the dynamic solution. Although B&B is not preferred, a subset of its search space is searched by VNS and it is compared with GA. The results obtained show that VNS (with a depth limit of 1) and GA find new solutions with improved gain and do not increase the time complexity too much. Increasing the distance of VNS increases the time complexity more than improving the gain. Therefore, using VNS (depth 1) iteratively (by repeating) is decided to be the best method in general. On the other hand, GA can also be used; however it has some extra time requirement in real applications.

To conclude, after the tests and comparisons, the thesis concludes that for a real fire support application, before the combat starts, the static WTPA problem should be solved by GA with tabu search and after the combat starts, the algorithm should go on with continuous (recursive) VNS with depth limit 1 for dynamic WTPA.

As a future work, the present study can be improved by considering the following additional jobs of fire support automation systems to be taken care of:

- Registration missions,
- Fire support scheduling for target groups,
- Coordinated illumination missions,
- Mass fire missions,
- Different practical constraints like terrain features,
- Moving targets and estimation theory solutions.

Additionally, WTPA algorithms can be coupled with estimation methods and be used as basic units of decision making systems.

REFERENCES

[1] Zne-Jung Lee, Shun-Feng Su, and Chou-Yuan Lee, “Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms with Greedy Eugenics”, IEEE Transactions On Systems, Man, And Cybernetics, 2003.

[2] Karla Hoffman, “Combinatorial and Integer Optimization”, George Mason University, 1999, [online]. Available: http://iris.gmu.edu/~khoffman/papers/newcomb1.html [Accessed: August 25, 2008].

[3] S. P. Lloyd and H. S. Witsenhausen, “Weapon allocation is NP-complete”, in IEEE Summer Simulation Conf., Reno, NV, 1986.

[4] A. William Meter, and F. L. Preston, “A Suite of Weapon Assignment Algorithms for a SDI Mid-Course Battle Manager”, AT&T Bell Laboratories, 1990.

[5] Michael R. Garey and David S. Jonson, W.H. Freeman and Company, “Computers and Intractability: A Guide to the Theory of NP-Completeness”, New York, ISBN: 0-7167-1045-5, 1979.

[6] D.M. Tate and A. E. Smith, “A genetic approach to the quadratic assignment problem”, Comput. Oper . Res., vol. 22, no. 1, pp. 73–83, 1995.

[7] A.M.H. Bjorndal et al., “Some thoughts on combinatorial optimization”, Eur. J. Oper. Res., pp. 253–270, 1995.

[8] I. Nicholas, M. Gould and Sven Leyffer, “An introduction to algorithms for nonlinear optimization”, RAL-TR-2002-03 (revised), 2002.

[9] D.A. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Reading, MA. Addision-Wesley, 1989.

[10] Y. Sun and Z. Wang, “The genetic algorithm for 0-1 programming with linear constraints”, in Proc. 1st. IEEE Conf. Evolutionary Computation, 1994.

[11] Nathan William Brixius, “Solving large scale quadratic assignment problems”, The University of Iowa, 2000. UMI Number: 9996075.

[12] Ravindra K. Ahuja, Arvind Kumar, Krishna C. Jha, James B. Orlin, “Exact and Heuristic Algorithms for the Weapon Target Assignment Problem”, 2003.

[13] Adem Çakır, H. Ender Başdaş, Kemal Leblebicioğlu, “Silah Etkinlik Katsayılarının Bulanık Mantık Yöntemi ile Tespit Edilmesi ”, Kara Harp Okulu Savunma Bilimleri Dergisi, Cilt 5, Sayı 1, Mayıs 2006.

[14] Zhong Ma, “The generalized quadratic assignment problem”, University of Toronto, 2003.

[15] Mei Wu, “Genetic Algorithms for Quadratic Assignment Problems”, University of Nevada, Reno, 1999. UMI Number: 1398123.

[16] Stuart J. Russell, Peter Norvig, Artificial Intelligence, “A Modern Approach”, 2003.

[17] Afşar Saranlı, “An Introduction to Artificial Intelligence”, Lecture N 2006.

[18] Kent University, http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/breadthSearch.htm, last accessed: August 25, 2008. “http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/depthSearch.htm”, last accessed: August 25, 2008.

[19] Tchler's web site, "http://wiki.tcl.tk/4176", last accessed: August 22, 2008.

[20] Geoffrey Vilcot, Jean-Charles Billaut, "A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem", European Journal of Operational Research 190 (2008) 398–411, 2008.

[21] Ravindra K. Ahuja, James B. Orlin, Ashish Tiwari, "A greedy genetic algorithm for the quadratic assignment problem", Computers & Operations Research 27 (2000) 917-934, 2000.

[22] J.F. Gonçalves, J.J.M. Mendes, M.G.C. Resende, "A genetic algorithm for the resource constrained multi-project scheduling problem", European Journal of Operational Research 189 (2008) 1171–1190, 2008.

[23] Lihong Ren, Yongsheng Ding and Shihuang Shao, "DNA Genetic Algorithms for Design of Fuzzy Systems", Department of Automation, College of Information Science and Technology Donghua University (China Textile University), Shanghai 200051, China, 2005.

[24] M. Grötschel, L. Lovasz and A. Schrijver (1988). "Geometric Algorithms and Combinatorial Optimization", Springer, Berlin, 1988.

[25] Edmund K. Burke, Timothy Curtois, Gerhard Post, Rong Qu and Bart Veltman (2007), "A hybrid heuristic ordering and variable neighborhood search for the nurse rostering problem", European Journal of Operational Research 188 (2008) 330–341, 2008.

[26] M. Duran Toksarı, Ertan Güner (2006), "Solving the unconstrained optimization problem by a variable neighborhood search", Science Direct J. Math. Anal. Appl. 328 (2007) 1178–1187, 2007.

[27] S. Consoli, K. Darby-Dowman, N. Mladenovic, J.A. Moreno Perez, "Greedy Randomized Adaptive Search and Variable Neighborhood Search

for the minimum labeling spanning tree problem", European Journal of Operational Research, 2008.

[28] Jean-JacquesGarroi, PeterGoos,Kenneth Sörensen, "A variable-neighborhood search algorithm for finding optimal run orders in the presence of serial correlation", Journal of Statistical Planning and Inference, 2008.