Optimization of NPC Assignment for Attack Strategy in the Game Using Ant Colony System

Herti Miawarni

Department of Electrical Engineering, Bhayangkara University, Surabaya INDONESIA.

herti_mia@ubhara.ac.id

ABSTRACT

In a fighting game, especially Real Time Strategy (RTS), generally there were two or more teams fighting each other. Each team in the game certainly played by a lot of NPCs (Non Player Character) that moved autonomous. Thus, it taken a special intelligence for NPCs in terms of the assignment which was a guideline for each NPC to behave. Assignment would affected almost all the NPCs behavior, such as where NPC will move, when the NPC will carry out attacks, the nearest enemy who should be attacked first, who the team member will be assisted comrades nearby, and others. Besides optimal assignment can make the game became more realistic, it also serves as a strategy of NPC in an attempt to defeat the opponent effectively. Assignment was calculated based on the distance parameter between the NPC unit team with each unit of NPC opponent.

In this paper was used the optimization algorithm Ant Colony System (ACS) which is one optimization method of shortest path finding. The goal of optimization was to generate a list of assignments for each unit of NPC to attack each opponent, then the total mileage in battle to be shorter, effective and efficient. Moreover, in this paper also discussed about the effect of using the optimization algorithms to the assignment of opponent. There were three different results on the total mileage of NPC in battle in some cases. The three different results were, first, if both of team used the optimization algorithm, the second when only one team used optimization algorithms, and third if there was no team that used optimization algorithm.

The results of this paper were if there was no team used optimization algorithm (Non Optimization vs Non Optimization), then the total mileage NPC achieved 444.39. Then if just one team that used optimization algorithm (ACS vs Non Optimization), the total mileage NPC achieved 433.65. And if both of team used optimization algorithm (ACS vs ACS), total mileage NPC achieved 402.95. These results indicated that the differences optimization NPC on a team assignment would affected the assignment of NPC opponents. Moreover, the differences in these results proved that using optimization Ant Colony System (ACS) was suitable solution in the assignment problem of NPC.

Keywords: Autonomous NPC, assignment, distance parameter, ant colony system, total Mileage.

INTRODUCTION

RTS games are the types of games that require players to be able to immediately make a decision quickly, because at the same time the opponents also act to execute its strategy, thus simultaneously can cause a series of events in real time. This type of game involves a lot of character, played by a number of NPCs, some examples of games that include the RTS game types are Age of Empire, Command, Rise of Nations, Army Men RTS, etc.

NPC or which may be called by the agent, according to Webster's New World Dictionary (Guralnik, 1983), the agent has been defined as a person or thing that acts or is capable of

ISSN: 2223-9944, e ISSN: 2223-9553
www.journals.savap.org.pk

acting or is empowered to act, for another. From these definitions it can be concluded into two points, namely:

- Agent has the ability to perform a task / job.
- Agent performs a task / job in the capacity for something, or to anyone else.

The intelligence of NPC obtained by adding algorithms of Artificial Intelligent (AI) inside. The Intelligence of this behavior will be applied to the thinking process of NPCs, for more detail can be seen in Table 1.1. Meanwhile, one of the characteristic of agent and should be owned by the NPC in the RTS game (Real Time Strategy), is to be autonomy.

Table 1.1 Stages of NPC Autonomous Behavior (Fanani, 2012).

Stages	Explanation
Sense	The ability to determine the state of the environment and themselves.
Think	The ability to think and choose the action from some available actions based on the results of the previous sensing process.
Act	The chosen action is the result of the thinking process of agent.

Autonomy is one of the many characters of agent. According to (Fanani, 2012), an agent can be said to have the properties of autonomy, if the agents can perform tasks independently and is not affected directly by the user, other agents or by the environment. To achieve the objective in performing their duties independently, the agent must have the ability to control every action that is done, either to the action outside and inside. And one more important thing that supports autonomy is a matter of intelligence of the agent.

Overall in the fighting game, the movement of an autonomous NPC group was not only influenced by its intelligence, but also the intelligence possessed by opponent NPC. When the NPC grouped into two camps, namely NPC Player and Enemy NPC fight each other, then the movement was performed by each unit of NPC player will affect the Enemy NPC units. This is because the battle always is preceded by the movement toward the target. If the NPC Player with optimization in the process of finding the shortest path, then the movement of the Enemy NPC will indirectly Optimized. In this research conducted trials ACS optimization algorithm to optimize the total mileage NPC especially during the battle and its effect on NPC opponents.

METHOD

In this paper, the test was done by dividing the NPC in two groups (Player and Enemy) which was designed to fight each other. In addition each NPC placed in a coordinate and moved according to intelligence possessed. The design of the system to be tested was as in Figure 1.

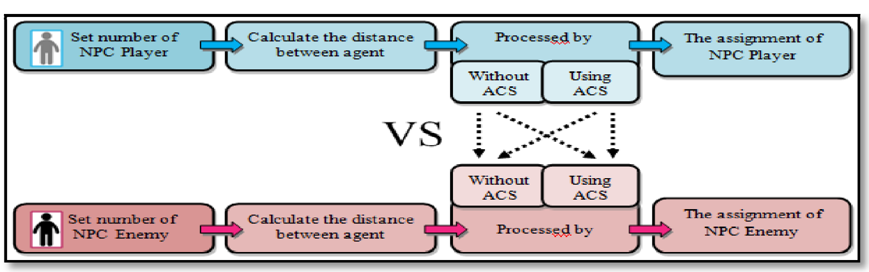

Figure 1. NPC Trial Block Diagram

Set Number of NPC

The first step was to set the number of NPC who were involved in the optimization process. In this paper, the number of NPC Enemy and NPC Player, bounded amounted to 30 agents. Thus, each of the NPCs player will perform the attack on NPCs enemy which was the target should be attacked. But it was not rule out the possibility of two or more NPCs player will attack the same target.

Calculate The Distance Between Agent For Initialization Distance Parameter

Each of NPC Enemy and NPC Player placed on a coordinate *(x,y)* in the gaming arena. The calculation of distance between them resolved by using the theory of Euclidean distance. If the coordinates of the NPC Player described as (x_1, y_1) and the coordinates of the NPC Enemy described as (x_2, y_2), then the distance between the player and enemy NPCs can be written as follows:

$$Z = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (2.1)$$

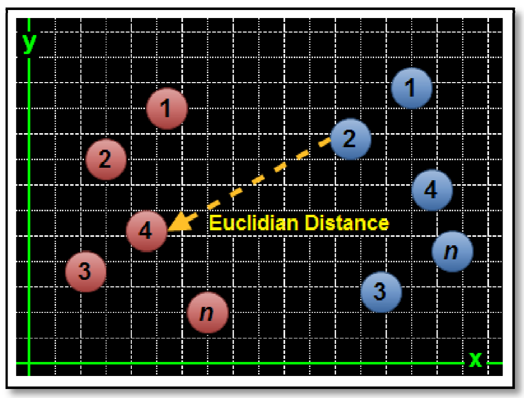

Figure 2. Distance Calculation of NPC Using Euclidian Distance

Example :

- The coordinate of NPC Player

  (x,y) ➔ (452,59)

- The coordinate of NPC Enemy

  (x,y) ➔ (609,38)

- The distance of NPC Player to NPC Enemy is :

  $$Z = \sqrt{(452 - 609)^2 + (59 - 38)^2} = 158$$

So the distance of NPC Player to NPC Enemy is 158 units.

Ant Colony Algorithm Design System (ACS)

Flowchart which shown in Figure 3. was the plot how the ACS algorithm was used to solve the optimization on assignment attack of NPC Player to NPC Enemy.

 ISSN: 2223-9944, e ISSN: 2223-9553

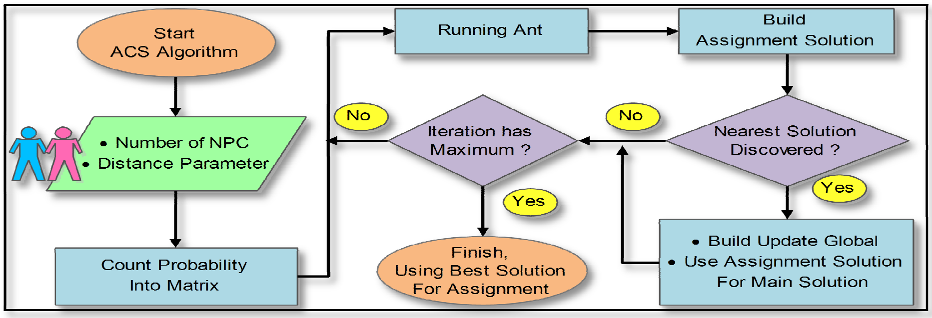

Figure 3. ACS Algorithm Flowchart

Figure 3. the optimization process flow diagram using the ACS algorithm can be described into a number of important points, among others :

Initialization in ACS

ACS initialization includes initialization all parameters into the matrix assignment. Matrix assignment is a set of figures which visualize the probability of assignment of each NPC. In this paper, there are two the assignment matrix, each belonging to player and enemy.

$$\begin{array}{c} Player \\ Assignment \end{array} = \begin{pmatrix} p_1e_1 & p_1e_2 & p_1e_3 & \cdots & p_1e_{(n)} \\ p_2e_1 & p_2e_2 & p_2e_3 & \cdots & p_2e_{(n)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{(n)}e_1 & p_{(n)}e_2 & p_{(n)}e_3 & \cdots & p_{(n)}e_{(n)} \end{pmatrix}$$

$$\begin{array}{c} Enemy \\ Assignment \end{array} = \begin{pmatrix} e_1p_1 & e_1p_2 & e_1p_3 & \cdots & e_1p_{(n)} \\ e_2p_1 & e_2p_2 & e_2p_3 & \cdots & e_2p_{(n)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{(n)}p_1 & e_{(n)}p_2 & e_{(n)}p_3 & \cdots & e_{(n)}p_{(n)} \end{pmatrix}$$

Where:

p = NPC player, e = NPC enemy, n = number of NPC

If the number of NPC Player and Enemy amounted to 30, then the assignment matrix has a size of 30 x 30. Then, all parameters are incorporated into the matrix. Matrix assignment is assumed as ants travel route, and an assignment is assumed to be a node, as seen in Figure 4.

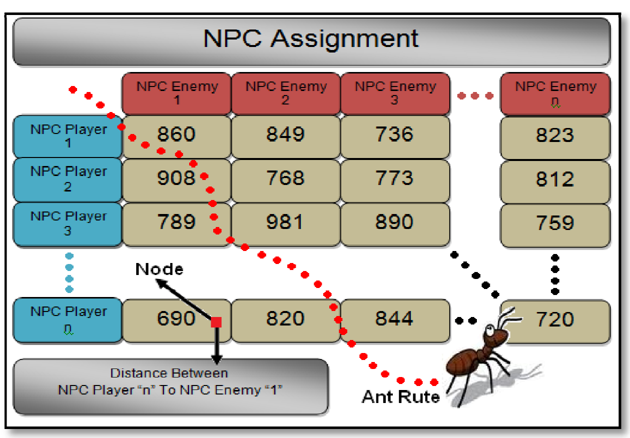

Figure 4. Ants Travel Route

ISSN: 2223-9944, e ISSN: 2223-9553

Running Ants

In this process, a number of ants run to take a trip on the nodes and leave pheromone. It can be written with the following formula :

$$\tau_{r,s} \leftarrow \tau_{r,s} + \Delta\tau^{k} \qquad (2.2)$$

Where :

r,s = name of node r,s, τ = pheromone viscosity, k = ant (k)

The formula can be described that each node which through the ants will be given a constant value. The size of the constant value depending on the setting of pheromones. This constant value will continue to grow along with the number of ants that passed and the number of iterations, as shown in Figure 5.

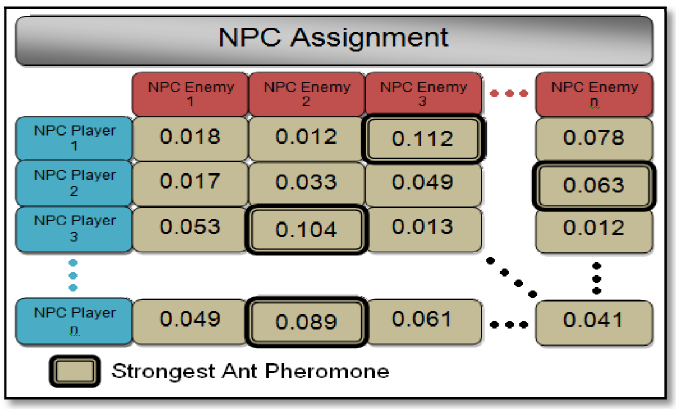

Figure 5. Strongest Pheromone in Matrix of Assignment

By the increasing value of pheromone on the node, then the node will be selected allowing the ants to be passed back to the next iteration. This will lead to the assignment of the NPC becomes stagnant. To avoid this, the pheromone evaporation imposed by the rule as follows:

$$\tau_{r,s} \leftarrow (1-\rho)_{\tau_{r,s}}, \mathrm{j}\,; \forall(r,s) \in A \qquad (2.3)$$

Where:

- ρ = evaporation rate expressed pheromone evaporation rate.
- A = node that has been passed by the ants.

The reduction in the amount of pheromone allows ants to explore other nodes in order to obtain a better solution of assignment.

Build The Assignment Solution

Setting pheromone and an evaporation rate at the end of the iteration produces multiple nodes with the highest pheromone value which is used as the initial solution assignment. However, searching for solutions is not just to be here, but it will be done continuously during the battles between NPCs in the game takes place. In building solution, to avoid stagnant assignment, apply transition rules that affect the likelihood of an ant exploring a new node to find a better solution. This is the flow of ants in the process of constructing a solution (Dorigo, 1997).

ISSN: 2223-9944, e ISSN: 2223-9553

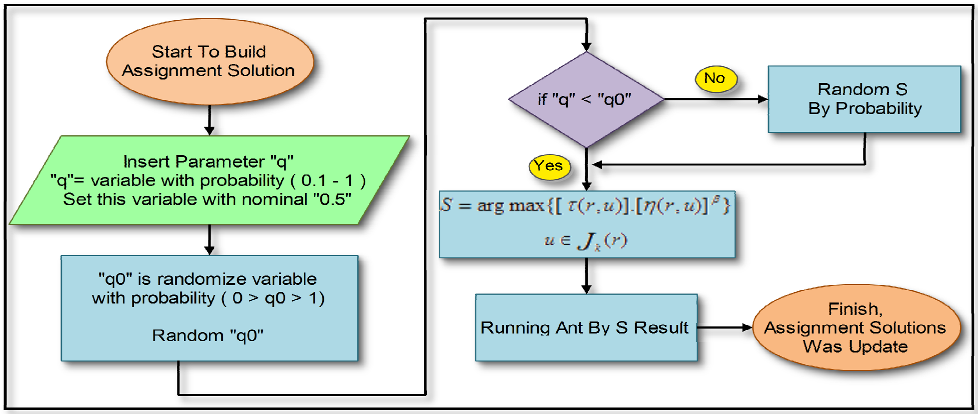

Figure 6. Flowchart of Process To Build Solution of Assignment

As seen in Figure 6, q and q0 are an auxiliary variable to determine the ant decision in choosing to exploit existing assignment or explore a new assignment. q given a value of 0.5 that the two possibilities have equally probable.

At first, q randomly generated value.

- If $q < q_0$ then it will be conducted the exploitation using the rules :

$$S = arg\ max\ \{\ [\tau\ (\ r,u\ )]\ .\ [\eta\ (\ r,u\ )]^{\ \beta}\ \} \qquad (2.4)$$

$$u \epsilon J_k\ (\ r\ )$$

- If otherwise, it will be conducted the exploration.

The results of this process then will be compared with existing assignment solution. If it is better, it will be conducted a global update.

Update Global For Assignment Solution

Update Global is a solution in which the pheromone is updated globally, so resulting in the assignment solution. The assignment taken from a collection of nodes that has the highest pheromone levels. As seen in Figure 6, where the figure with orange color is a node that has higher levels of pheromone than other nodes.

Update Global on the ACS only done if the ants get a better solution than the solution which has been built previously. Update Global has the following rules:

B. $$\tau(r,s) \leftarrow (1 - \alpha).\ \tau(r,s) + \alpha\ \Delta\ \tau(r,s) \qquad (2.5)$$

Where :

- $\Delta\ \tau\ (r,s)\ = \begin{cases} (L_{gb})^{-1} & if\ (r,s)\ global\ best\ tour \\ 0 & \end{cases}$
- $0 < \alpha < 1$ = pheromone decay parameter
- L_{gb} = distance of the best tour since the beginning

Discontinuation Of Optimization Process

Optimization process to obtain the assignment solution will stop if it reaches the iteration limit. In this case, the solution of assignment follows the greatest pheromone value in the table pheromones. But the optimization process will still continue as long as there is fighting between the NPC Player with NPC Enemy.

ISSN: 2223-9944, e ISSN: 2223-9553

RESULT AND DISCUSSION

The Results of the distance optimization can be seen from the total distance of all NPCs is involved in the battle. The average total mileage across the NPC is calculated by the rule:

The Average of NPC total mileage =

$$\frac{\text{NPC Player Total Mileage} + \text{NPC Enemy Total Mileage}}{2}$$

As for setting some attributes on ACS algorithm is as follows:

a. Iteration = 10
b. Pheromone Trail (□) = 0.1
c. Evaporation Rate (□) = 0.01
d. Ants = 10
e. Exploration (α) = 1
f. Exploitation (β) = 1

After the entire NPC placed in each coordinate, then the test is done in 3 stages. The first step is a Non Optimization mode vs Non Optimization. In this case, both of NPC Player and NPC Enemy didn't used the optimization algorithm ACS. From the 10 times trials, the results obtained in Figure 7.

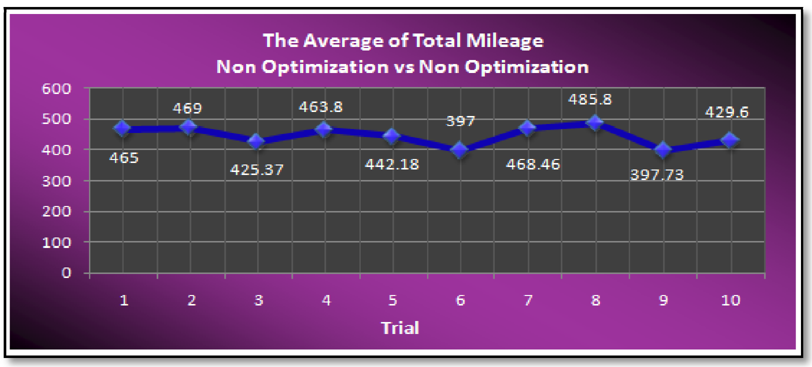

Figure 7. Graph 10 Times Trials of Non Opt vs Non Optimization

Then the second step is ACS Optimization vs Non Optimization. In this case, NPC Player using ACS optimization algorithm, while NPC Enemy didn't used optimization algorithms. From the 10 trials, the results obtained in Figure 8.

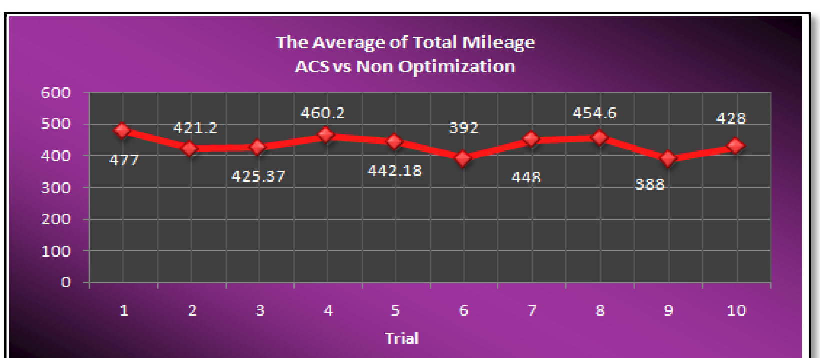

Figure 8. Graph 10 times Trials of ACS vs Non Optimization

 ISSN: 2223-9944, e ISSN: 2223-9553

And the last is ACS Vs ACS, in this case, NPC Player and Enemy at the ACS optimization algorithm. From the 10 times trials, the results obtained as in Figure 9.

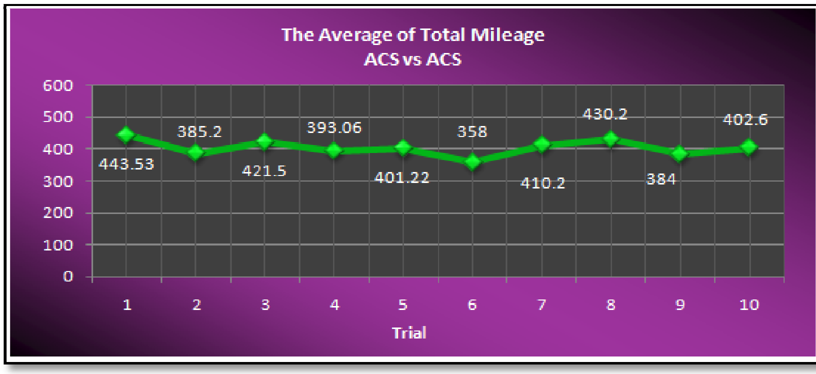

Figure 9. Graph 10 Times Trials of ACS vs ACS

If the three results compared simultaneously, it will be shown as in the Table 3.1.

Table 3.1. Average Total NPC Mileage

Trial	Non Optimization vs Non Optimization	ACS vs Non Optimization	ACS vs ACS
1	465	477	443.53
2	469	421.2	385.2
3	425.37	425.37	421.5
4	463.8	460.2	393.06
5	442.18	442.18	401.22
6	397	392	358
7	468.46	448	410.2
8	485.8	454.6	430.2
9	397.73	388	384
10	429.6	428	402.6
Average	444.39	433.65	402.95

Data of average total NPC mileage in the table 3.1, if it is visualized into the graph will be shown as in the Figure 10.

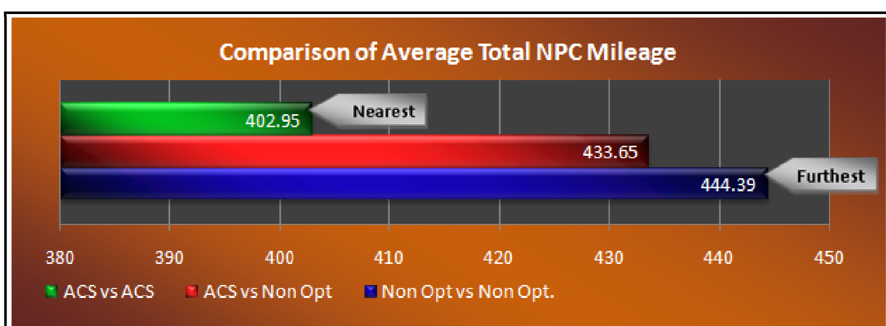

Figure 10. Average Total NPC Mileage

CONCLUSION

From the overall trial results can be concluded that if there is one or a whole group of NPCs move based optimization process, it will affect the movement of NPC opponents. So it will affect the overall battle in a fighting game. The average total mileage of NPC with ACS vs ACS mode is shorter than the other modes. And the average total mileage ACS vs Non optimization mode is shorter than if both of them used non optimization mode. Here are the results of average total mileage of intelligence variation modes used in battle of NPC:

- *ACS vs ACS* = 402.95
- *ACS* vs Non Opt = 433.65
- Non Opt vs Non Opt = 444.39

REFERENCES

[1] Dorigo et. al., (1996). *"The Ant System : Optimization by a Colony of Cooperating Agents". IEEE Transactions on Systems,* Man, and Cybernetics - Part B, *26*(1):29-41.

[2] Dorigo et. al., (1997), "Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem ". *IEEE Transactions on Evolutionary Computation, 1*(1).

[3] Dorigo et. al., (November 2006). "Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique". *IEEE Computational Intelligence Magazine.*

[4] Endah, D. dkk., (2010). "*Pencarian Jalur Terpendek Pada Pemodelan Pergerakan Agen Cerdas dengan Algoritma Ant Colony System".* Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.

[5] Guralnik, D. B. (1983). "Webster's New World Dictionary".

[6] John Von Neumann, Oskar Morgenstern. (1953). "Theory Of Games And Economic Behaviour". Princeton University Press.

[7] L.M. Gambardella et. al., (Feb 1999). "Ant colonies For The Quadratic Assignment Problem*". The Journal of the Operational Research Society, 50*(2), pp. 167-176.

[8] Nurul, Z. F. (2012). "*Optimisasi Strategi Menyerang Kelompok NPC Pada Game Pertarungan Jarak Dekat Menggunakan Algoritma Harmony Search".* Tesis. Teknik Elektro, Fakultas Teknologi Industri. ITS, Surabaya.

ISSN: 2223-9944, e ISSN: 2223-9553
www.journals.savap.org.pk