Journal of Quality Measurement and Analysis *JQMA* 11(2) 2015, 17-29
Jurnal Pengukuran Kualiti dan Analisis

A STUDY ON AN EXTENDED PREY-PREDATOR ALGORITHM

(Kajian ke atas Al-Khwarizmi Lanjutan Mangsa-Pemangsa)

HONG CHOON ONG & CHIA JIUN NG

ABSTRACT

Metaheuristic algorithms are approximate solution methods for optimisation problems which try to improve the quality of solution at hand iteratively in a random way. In recent years, various studies have been conducted in forming new metaheuristic algorithms and modifying or improving existing algorithms to enhance the performance in optimal solution search. In this study, we focus on extending an existing algorithm Prey-Predator algorithm proposed by Tilahun and Ong. Prey-Predator algorithm is a metaheuristic algorithm inspired by interaction between prey and predator among animals. The algorithm imitates the way a predator runs after and hunts its preys where each prey tries to stay with the pack trying to search for hiding place and run away from the predator. In extension of Prey-Predator algorithm, the number of both best preys and predators are increased resulting in a more reasonably exploitation and exploration so that multiple solutions can be achieved. The simulation of nmPPA is carried on ten selected benchmarks test function. nmPPA aimed to solve the problem of objective values being trapped in local optimum and to find multiple solutions at the same time.

Keywords: prey-predator; optimisation; metaheuristics

ABSTRAK

Al-Khwarizmi metaheuristik merupakan penyelesaian hampiran bagi masalah pengoptimuman yang digunakan untuk meningkatkan kualiti penyelesaian sedia ada secara lelaran dan rawak. Sejak kebelakangan ini, pelbagai penyelidikan telah dijalankan dalam membina al-Khwarizmi metaheuristik baharu dan mengubah suai atau memperbaiki al-Khwarizmi yang sedia ada bagi meningkatkan prestasi dalam pencarian penyelesaian yang optimum. Dalam kajian ini, penambahbaikan al-Khwarizmi Mangsa-Pamangsa yang dikaji oleh Tilahun dan Ong telah dilakukan. Al-Khwarizmi Mangsa-Pemangsa adalah al-Khwarizmi metaheuristik yang diilhamkan berdasarkan interaksi haiwan antara mangsa dengan pemangsa. Al-Khwarizmi Mangsa-Pemangsa meniru cara pemangsa berlari dan memburu mangsanya yang setiap mangsa cuba untuk berada dan bergerak dalam kumpulan bagi mencari tempat untuk berlindung dan melarikan diri daripada pemangsa. Dalam penambahbaikan al-Khwarizmi Mangsa-Pemangsa, bilangan kedua-dua mangsa dan pemangsa ditingkatkan bagi menghasilkan lebih banyak eksploitasi dan penerokaan yang lebih munasabah supaya pelbagai penyelesaian dapat dicapai. Simulasi nmPPA telah diuji terhadap sepuluh fungsi ujian terpilih bagi kayu ukur terhadap ujian. Simulasi nmPPA bertujuan untuk menyelesaikan masalah nilai objektif yang terperangkap dalam optimum tempatan dan mencari berbilang penyelesaian pada masa yang sama.

Kata kunci: mangsa-pemangsa; pengoptimuman; metaheuristik

1. Introduction

Mathematical optimisation is the study of designing optimisation problem which consists of objective function, either simple or multiple and a set of constraints via mathematical tools. However due to complexity of the problem, the traditional analytical methods are not sufficient to solve complex models. In complex problems, it is also impractical to search for all the possible combination of solutions in the solution space to find the optimal solution.

Metaheuristic optimisation algorithm is one of the alternative methods to improve the situation.

Metaheuristic algorithm is defined as algorithm which employs ‘educated guess’ with randomisation in performing solution search and improves the solution quality through repeated search as the algorithm start with a set of randomly generated solution by implementing exploration and exploitation in the solution space (Tilahun & Ong 2014). The ‘educated guess’ indicates a local search. As metaheuristic algorithm extends from heuristic algorithm, it performs better compared to the latter one by introducing randomisation and local search as the search basis which prevents the solution search from being trap in the local optimum. The purpose of exploitation is to execute local search by exploiting the region surrounding the current good solution in search of if this region exists an improved solution. On the other hand, exploration function explores the feasible solution space randomly on a global basis by generating diverse solutions. Randomisation allows a more thorough search where the search is at a global basis which makes metaheuristic algorithm a more appropriate option for global optimisation. The best solution can be found by combining both exploration and exploitation. Thus global optimum can be found through metaheuristic algorithm (Yang 2010).

The development of metaheuristics is influenced by several inspirations. Generally, the main three inspirations are the human brain, the Darwinian evolution and the social behaviour of nature. Among the recently developed metaheuristic algorithm, there are several algorithms which are inspired by nature. For example, the search mechanism of Prey-Predator algorithm is inspired from the interaction between the role of prey and predator in animals (Tilahun & Ong 2015), and the search mechanism for Simulated Annealing algorithm is based on the concept of changes in temperature (Kirkpatrick & Vecchi 1983).

The main objective of our study is to extend the Prey-Predator algorithm by increasing the number of best preys and predators so that multiple solutions can be achieved. We are also interested in comparing the objective function values by assigning different number of predators and best preys with fixed iterations on selected benchmark test functions.

2. Methodology

2.1. *Prey-Predator Algorithm (PPA)*

In this study, the focus is on the Prey-Predator algorithm, developed by Tilahun & Ong (2015) and inspired by interaction between prey and predator among animals. A prey is an animal hunted by another animal which is called the predator. Hence, the predators have managed to hunt and catch the weak preys, while the other hand the preys have also learnt how to escape from predators, searching for hiding place and survive.

In the algorithm, a set of initial feasible solution will be generated and each solution number, x_i, will be assigned with a survival value, $SV(x_i)$ based on the objective function of the optimisation problem. A higher survival value implies better performance in the objective function. This means for solution x_i and x_j, if x_i performs better than x_j in the objective function then $SV(x_i) > SV(x_j)$. In general:

$$SV(x_i)\ \alpha\ (SV(x_j))^{\alpha} \tag{1}$$

where $SV(x_i)$ is the survival value for x_i, $f(x_i)$ is the objective function for x_i and a is -1 and 1 for minimisation and maximisation problem. After generating survival value for each solution number, solution with the smallest survival value will be assigned as predator, $x_{predator}$

and the rest as preys. Once assigned the prey and predator, each prey needs to run away from the predator and tries to follow a better prey in terms of their survival value or look for a hiding place at the same time. The best prey does not need to look for hiding place, as it is considered as a prey which has already found hiding place and do not need to explore for hiding place. Thus the prey, $x_{best\ prey}$ is said to be best prey if $SV(x_{best\ prey}) > SV(x_i)$, for all i which means the best prey has the highest survival value among all the solutions (Tilahun & Ong 2014).

The predator does the exploration whereas the best prey focuses on exploitation. The best prey is considered as a prey that has found the hiding place and is not hunted by the predator, thus it does not need to run away from the predator. On the other hand, the rest of the prey will do both exploitation and exploration but mainly focus on exploration. They follow preys with better survival values while running away from predator and at the same time doing a local search. The movement of prey and predator involve two basic factors, the direction and the step length.

In the algorithm, the movement of the prey depends on the follow up probability, p_f. If the follow up probability is met, the prey will follow the other prey with better survival value and does a local search and at the same time search for hiding place. However if the follow up probability is not met, the prey will run away randomly from the predator. Suppose the follow up probability is met and there are x_1, x_2, …, x_s preys which have better survival value compared to x_i. Since most of the preys tend to follow the nearest pack therefore the movement direction is dependent on the distance between x_i and preys with better survival value, x_j. Hence the direction of the movement of x_i can be calculated as follows (Tilahun & Ong 2014):

$$y_i = \sum_j e^{SV(x_j)^v - r_{ij}} (x_j - x_i) \tag{2}$$

where $r_{ij} = \|x_j - x_i\|$ and v plays the role of magnifying or diminishing the effect of the survival values in the distance y_i. By changing the value for v, it is possible to adjust the dependency of the direction on the survival value and the distance and the assigning a large or small value does not affect the jump size of x_i. Thus the unit direction will be used to represent the direction as:

$$u_i = \frac{y_i}{\|y_i\|} \tag{3}$$

Besides that, the local search is done by generating q random directions and check if there is any possible direction that can increase the survival value of x_i if the solution moves in that direction. If such direction exists, that direction will be taken as a local search direction and be represented by y_l. If such a direction does not exist, y_l will be put as a zero vector. However if the follow up probability is not met, the prey will run away from the predator randomly. This is done by generating a random direction y_r and comparing the distance between the predator and prey to decide if the prey moves in direction y_r or $-y_r$. The direction which takes the prey far from the predator will be chosen.

The best prey will perform only local search. It only moves to direction which can improve its survival value from a randomly generated q directions or stays in its current position if no such direction exist among the q directions.

For the case of predator, its main task is to motivate the prey for exploring the solution space while it also does the exploration of the solution spaces. Thus, it will chase after the prey with the least survival value and also moves randomly in the solution space.

In real prey predator situation, those preys which are nearer to the predator need to run faster compared to those who are far from the predator. Same situation is also applied in the algorithm. Solution with small survival values needs to run faster and thus they need to have larger step length in comparison with solutions with larger survival values. Therefore the step length, λ is inversely proportional to survival values of prey. The simplified formula is shown as:

$$\lambda = \lambda_{\max} rand \tag{4}$$

Generalizing all the points across, the movement of the prey, excluding the best prey can be summarised as:

$$x_i \leftarrow x_i + \left(\frac{y_i}{\|y_i\|} \right) \lambda_{\max} rand + \left(\frac{y_l}{\|y_l\|} \right) \lambda_{\min} rand \tag{5}$$

if the follow up probability is met. If the follow up probability is not met, the movement of the prey will be given as:

$$x_i \leftarrow x_i + \left(\frac{y_i}{\|y_i\|} \right) \lambda_{\max} rand \tag{6}$$

For the step length movement of the best prey, it is given as:

$$x_{best\ prey} \leftarrow x_{best\ prey} + \left(\frac{y_l}{\|y_l\|} \right) \lambda_{\min} rand \tag{7}$$

For the case of the predator, which carries the role of chasing the weaker preys and updating its location, its formula is given as:

$$x_{predator} \leftarrow x_{predator} + \left(\frac{y_r}{\|y_r\|} \right) \lambda_{\max} rand + \left(\frac{x'_i - x_{predator}}{\|x'_i - x_{predator}\|} \right) \lambda_{\min} rand \tag{8}$$

where y_r is a random vector for the randomness movement of the predator, *rand* is a random number range between 0 and 1 from a uniform probability distribution and x'_l is a prey with the least survival value. The Prey-Predator algorithm is summarised in Figure 1.

2.2. *nmPPA*

Some optimisation problems may have multiple local and global optimums. Moreover, in some cases the global optimum is far from the local solution. Thus it is a challenge in metaheuristic algorithm when objective value tends to be trapped in the local optimum. In the Prey-Predator algorithm, the number of both best preys and predators are increased resulting in a more reasonably exploitation and exploration so that multiple solutions can be achieved. The extension of PPA is aimed to solve the problem of objective value being trapped in local optimum and to find multiple solutions at the same time by setting *n* number of solutions as predators and *m* as best preys. The *n* number of predators will be focused on exploration, while the *m* best preys will focus on exploitation. The assigning of *n* predators will be from *n* lowest survival values assigned while the *m* best preys will be the *m* top solutions having the highest survival values. The movement of these solutions will be governed in the same way as mentioned in PPA, where *m* = *n* = 1.

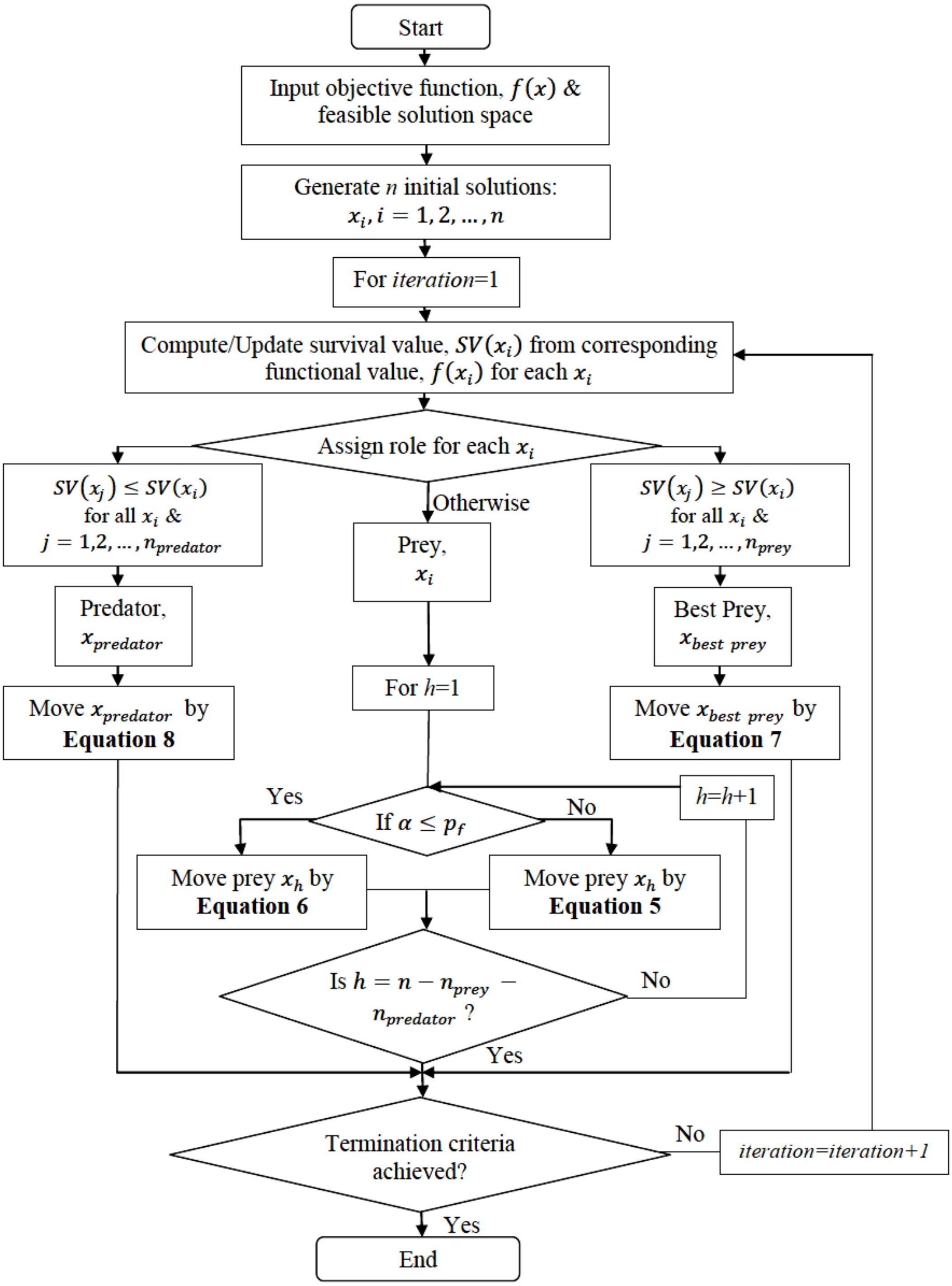

Figure 1: Flow chart of the extended Prey-Predator algorithm

3. Simulation Results and Discussions

3.1. *Benchmark Test Function*

The benchmark test functions are selected from different categories, which include continuity, differentiability, separability and dimensionality of unimodel or multimodel objective functions. For uniformity purpose, all the problems are taken as minimisation problems and those which are maximisation problems are converted to minimisation problem by multiplying the objective function with negative one. Thus all benchmark test functions can be formulated as follows:

$$\min\ f\left(x_1, x_2, ..., x_n\right)$$

$$\text{such that } \left(x_1, x_2, ..., x_n\right) \in \mathbb{R}^n \tag{9}$$

(1) Ackley function (Ackley 1987) is a two dimensional continuous, differentiable, non-separable unimodel function. It has only one global minimum, located at the origin# $x^* = (0,0)$ #with objective function value $f_1(x^*) = -200$, where

$$\min\ f_1(x) = -200e^{-0.02\sqrt{x_1^2 + x_2^2}}$$

$$\text{such that } -32 \le x_1, x_2 \le 32. \tag{10}$$

(2) Bartels Conn function (Jamil & Yang 2013) is a two dimensional continuous, non-differentiable, non-separable multimodal function. The global minimum of the function is located at $x^* = (0,0)$ with objective function value $f_2\left(x^*\right) = -1.00$ where

$$\min\ f_2\left(x\right) = -\left|x_1^2 + x_2^2 + x_1x_2\right| + \left|\sin\left(x_1\right)\right| + \left|\cos\left(x_2\right)\right|,$$

$$\text{such that } -500 \le x_1, x_2 \le 500 \tag{11}$$

(3) Beale function (Jamil & Yang 2013) is a two dimensional continuous, differentiable, non-separable unimodal function. The global minimum of the function is located at $x^* = \left(3, 0.5\right)$ with objective function value $f_3(x^*) = 0.00$, where

$$\min\ f_3(x) = \left(1.5 - x_1 + x_1x_2\right)^2 + \left(2.25 - x_1 + x_1x_2^2\right)^2 + \left(2.625 - x_1 + x_1x_2^3\right)^2$$

$$\text{such that } -4.5 \le x_1, x_2 \le 4.5 \tag{12}$$

(4) Bird function (Mishra 2006) is a two dimensional continuous, differentiable, non-separable multimodal function. The two global minimum of the function is located at $x^* = \left(4.70104, 3.15294\right), \left(-1.58214, -3.13024\right)$ with objective function value $f_4\left(x^*\right) = -106.764537,$ where

$$\min\ f_4(x) = \sin(x_1)e^{(1-\cos x_2)^2} + \cos(x_1)e^{(1-\sin x_1)^2} + (x_1 - x_2)^2$$

$$\text{such that} -2\pi \le x_1, x_2 \le 2\pi \tag{13}$$

(5) Branin RCOS function (Branin 1972) is a two dimensional continuous, differentiable, non-separable multimodal function. It has three global minima, located at $x^* = \left(-\pi, 12.275\right), \left(\pi, 2.275\right), \left(3\pi, 2.425\right)$ with objective function value $f_5\left(x^*\right) = 0.3978873,$ where

$$\min\ f_5\left(x\right) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos\left(x_1\right) + 10$$

$$\text{such that } -5 \le x_1 \le 10, 0 \le x_2 \le 15 \tag{14}$$

(6) Camel – Six Hump function (Branin 1972) is a two dimensional continuous, differentiable, non-separable multimodal function. The two global minima of the function is located at $x^* = (-0.0898, 0.7126), (0.0898, -0.7126)$ with objective function value $f_6(x^*) = -1.0316$, where

$$\min f_6(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{4}\right)x_1^2 + x_1 x_2 + \left(4x_2^2 - 4\right)x_2^2$$

such that $-5 \le x_1, x_2 \le 5$ (15)

(7) Egg Crate function (Jamil & Yang 2013) is a two dimensional continuous, separable multimodal function. The global minimum of the function is located at $x^* = (0,0)$ with objective function value $f_7(x^*) = 0$, where

$$\min f_7(x) = x_1^2 + x_2^2 + 25\left(\sin^2(x_1) + \sin^2(x_2)\right)$$

such that $-5 \le x_1, x_2 \le 5$ (16)

(8) Leon Function (Lavi & Vogel 1966) is a two dimensional continuous, differentiable, non-separable unimodal function. The global minimum of the function is located at $x^* = (1,1)$ with objective function value $f_8(x^*) = 0$, where

$$\min f_8(x) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$$

such that $-1.0 \le x_1, x_2 \le 1.2$ (17)

(9) Mishra's function No. 06 (Mishra 2006) is a two dimensional continuous, differentiable, non-separable multimodal function. The global minimum of the function is located at $x^* = (2.88631, 1.82326)$ with objective function value $f_9(x^*) = -2.28395$, where

$$\min f_9(x) = -\ln\left[\sin^2\left(\left(\cos(x_1) + \cos(x_2)\right)\right)^2 - \cos^2\left(\sin(x_1) + \sin(x_2)\right) + x_1\right]^2 + 0.01\left(\left(x_1 - 1\right)^2 + \left(x_2 - 1\right)^2\right)$$

such that $-5 \le x_1, x_2 \le 5$ (18)

(10) Shubert's function (Molga & Smutnicki 2005) is a two dimensional continuous, differentiable, non-separable multimodal function. It has 760 minimum solutions among which 18 are global minimum with objective function value $f_{10}(x^*) = 186.7309$. The 18 global minima of the function are located at

x^* = (–7.0835, 4.8580), (–7.0835, –7.7083), (–1.4251, –7.0835)
(5.4828, 4.8580), (–1.4251, –0.8003), (4.8580, 5.4828)
(–7.7083, –7.0835), (–7.0835, –1.4251), (–7.7083, –0.8003)
(–7.7083, 5.4828), (–0.8003, –7.7083), (–0.8003, –1.4251)
(–0.8003, 4.8580), (–1.4251, 5.4828), (5.4828, –7.7083)
(4.8580, –7.0835), (5.4828, –7.7083), (4.8580, –0.8003)

$$\min f_{10}(x) = \left[\sum_{i=1}^{5} i\cos\left((i+1)x_1 + i\right)\right]\left[\sum_{i=1}^{5} i\cos\left((i+1)x_2 + i\right)\right]$$

such that $-10 \le x_1, x_2 \le 10$ (19)

3.2. *Results and Discussions*

This simulation is conducted whereby different number of preys and predators are tested on each benchmark test functions on fixed number of iterations, under same set of randomly generated feasible initial solutions, same local search direction and same maximum and minimum jumps with follow-up probability set to $p_f = 0.5$. Observation is done on the effectiveness of nmPPA in the accuracy of obtaining minimum solution of each functions.

(1) For the first benchmark test function, Ackley Function, the parameters are set fixed for each simulation. It is observed that we do obtain better objective function value when increasing the number of best preys and predators to more than one. When number of best preys is fixed as $n_{\text{best prey}} = 6$, the objective function value becomes closer to the exact global minimum solution as the number of predators increase. The objective function also shows increasing accuracy when number of best preys are increased under fixed number of predators. However there is no sight of increasing accuracy while both number of best prey and predator are increased. In conclusion, nmPPA does show increasing accuracy for the case of fixed *n*, increasing *m* and fixed *m*, increasing *n* on Ackley function. It is also shown that nmPPA on Ackley function gives a more accurate objective value compared to PPA.

(2) For the second function, Bartels Conn Function, two dimensional multimodel function. The exact global minimum solution for Bartels Conn function is $f_2\left(x^*\right) = -1.00$ with $x^* = (0,0)$. It is observed that objective function values from nmPPA are nearer to the exact global minimum compared to the objective function value for one best prey and one predator algorithm which is PPA. When numbers of best preys are fixed as $n_{\text{best prey}} = 6$ and the number of predators increases, the objective value show a slight trend of increasing accuracy. On the other hand, while number of best preys increases under fixed number of predators, a small increase trend is shown when number of best preys increases from 10 to 14. However there is no sight of increasing accuracy when both number of bests prey and predators are increased. Thus, the simulation on Bartels Conn function does show slight increment in accuracy under both conditions of fixed *n*, increasing *m* and fixed *m*, increasing *n*. Most nmPPA on this function also have more accurate objective value compared to PPA.

(3) The third function, Beale Function is a two dimensional unimodel optimisation problem. The exact global minimum solution for Beale function is located $x^* = (3.0, 0.5)$ with $f_3\left(x^*\right) = 0.00$. It is observed that objective function value from PPA of one best prey and predator has the most accurate objective value compared to objective values of other nmPPA. However it is observed that, the x^* points found by nmPPA is closer to the actual global minimum point. The result obtained when number of best preys is fixed and increasing the number of predators show a small decrease in accuracy when number of best preys is increased from 6 to 10. On the other hand, the results do not show any pattern while number of best preys increases under fixed number of predators. There is also no sight of increasing accuracy when both number of best preys and predators are increased. In conclusion, the nmPPA simulation on Beale function only show increasing accuracy trend when *m* is fixed with increasing *n* but not on the other two cases. It can also be concluded that nmPPA in this function performs better as it generates optimal points that are closer to the actual global optimum point.

(4) The fourth function, the Bird Function is a two dimensional multimodel optimisation problem. The exact global minimum solution for Bird function is $f_4\left(x^*\right) = -106.7645$ with two global minimum points $x^* = (4.70104, 3.15294), \left(-1.58214, -3.12024\right)$. It is observed that most trials show objective function value from the point $x^* = (-1.58214, -3.13024)$. Although there is no trend of increasing accuracy when number of predators increases with fixed number of best prey, but the result tend to

achieve the other global minimum point as the number of predators increase. This indicates that as the number of predators increases, more exploration is done in the solution space which enables a wider search for better survival values. While number of best preys increase under fixed number of predators, a small increase trend is shown when number of best preys increases from 4 to 10. However there is no sight of increasing accuracy while both number of best preys and predators are increased. It can be concluded that, under both condition of fixed *m*, increasing *n* and fixed *n*, increasing *m* the accuracy increases since another global minimum is found under fixed *m*, increasing *n*. Apart from that some nmPPA simulation of Bird function also show better accuracy compared to PPA.

(5) The fifth function, Branin RCOS function is a two dimensional multimodel function. Branin RCOS function has three global minima, located at $x^* = (-\pi, 12.275), (\pi, 2.275), (3\pi, 2.425)$ with all three having the same objective function value $f_5(x^*) = 0.3978873$. Simulation with 2 predators and 6 best preys generates the best objective value with the smallest error. There is no significant pattern shown when increasing the number of predators with fixed number of best preys or increasing the number of best preys with fixed number of predators. However, when the number of best preys and predators are increased to more than one, we did manage to obtain optimum solution from the other two global minimum point compared to one best prey and one predator PPA. Thus we can conclude that simulation on Branin RCOS function shows better result on nmPPA compared to PPA since by increasing the number of n and m enables more exploration and exploitation to take place which enables the algorithm to locate more global minimum point in the solution space. The result also show that the simulation on Branin RCOS function does not have any trend on both the cases of fixed *n*, increasing *m* and fixed *m*, increasing *n*.

(6) For the sixth function, Camel Six-Hump Function, the parameters are set fixed for each trial of simulation. Camel Six-Hump Function has two global minima with objective function value $f_6(x^*) = -1.0316$. The two global minimum points of the function are located at $x^* = (-0.0898, 0.7126)$ and (0.0898, –0.7126). nmPPA with 6 predators and 8 best preys generate the best objective value with the smallest error. There is no significant pattern shown when increasing the number of predators with fixed number of best prey or increasing the number of best preys with fixed number of predators. Only small trend of decreasing error when number of predators increase from 2 to 6 with fixed number of best preys $n_{\text{best prey}} = 6$. In conclusion, the nmPPA simulation for Camel Six Hump function performs better compared to PPA of this function. However the simulation on Camel Six Hump function only show small increasing accuracy trend on the fixed *m*, and increasing *n*.

(7) The seventh function, Egg Crate Function is of two dimensional multimodal optimisation problem. The global optimum for Egg Crate function is located at $x^* = (0,0)$ with objective function value $f_7(x^*) = 0$. There is no significant pattern shown when increasing the number of predators with fixed number of best preys or increasing the number of best preys with fixed number of predators. In conclusion, result show that the simulation on Egg Crate function does not have any trend on both the cases of fixed *n*, increasing *m* and fixed *m*, increasing *n*. However, the result by nmPPA simulation of this function does shows that the nmPPA on Egg Crate function enables the algorithm to get a more accurate result compared to PPA.

(8) The eighth function, Leon Function is a two dimensional unimodel function. Since Leon Function is a unimodel function, its only global minimum of the function is located at

$x^* = (1,1)$ with objective function value $f_8(x^*) = 0$. PPA with only one best prey and predator shows best objective value with least error which is only 0.0001. Apart from that, there is also no significant pattern shown when increasing the number of predators with fixed number of best preys or increasing the number of best preys with fixed number of predators. In conclusion, result show that the simulation on Leon function does not have any trend on both the cases of fixed *n*, increasing *m* and fixed *m*, increasing *n*. It is also shown that the PPA simulation of this function is better compared to the nmPPA simulation.

(9) Mishra's function No. 06, the ninth function, is two dimensional multimodel optimisation problem. The global optimum for Mishra's function No. 06 is located at $x^* = (2.88631,\ 1.82326)$ with objective function value $f_9(x^*) = -2.28395$. The best objective value is obtained at the simulation where when the number of predators is set to 6 and number of best preys set to 2 and also when the number of predators is set to 12 and number of best preys set to 6. There is also no sign of pattern shown when increase the number of predator with fixed number of best preys or increasing the number of best prey with fix number of predator for this function. Thus it is concluded that the simulation on Mishra 6 function does not have any trend on both the cases of fixed *n*, increasing *m* and fixed *m*, increasing *n*. However, the result by nmPPA simulation of Mishra 6 function does prove that the nmPPA on Mishra 6 function enables the algorithm to get a more accurate result compared to PPA.

(10) For our last function, Shubert's Function which is a two dimensional multimodel function. Shubert's Function contains 18 global minima out of 760 minimum solutions with objective function value $f_{10}(x^*) = 186.7309$. From simulation result, we managed to get different global minimum points when increasing the number of best prey and predator. It is also observed that some simulation results from nmPPA do give out more accurate objective solution compared to one best prey and one predator PPA. However the simulation on Shubert's function does not have any trend on both the cases of fixed *n*, increasing *m* and fixed *m*, increasing *n*. In this simulation the main task is more on exploring for the global minimum points than exploiting the best result around that specific points as this function contains 18 global minimum points.

After simulation on all the 10 benchmark test functions, we can summarise that nmPPA in most functions performs better compared than the PPA. The objective values of nmPPA shows better accuracy to the exact global optimum and generates optimum points that are closer to the global optimum point. Trend of increasing accuracy is observed when either number of best preys or predators are being fixed while increasing the other in unimodel function (Ackley function) and multimodel function with only one global optimum (Bartels Conn function). The functions with only one global optimum leads the algorithm to search for the best result at that certain point instead of exploiting for other better points in the solution space. However the result of multimodel functions with multi global optimum points does not show any increase in accuracy both the cases of fixed *n*, increasing *m* and fixed *m*, increasing *n*. As these multimodel functions contain more than one global optimum, the increasing of number of best preys and predators enable the nmPPA to spread out the exploration and exploitation to search global optimum in several optimum points. Thus there is no increased accuracy as it is more focused on exploring for different global optimum points.

Table 1: Simulation Result for 10 Benchmark Functions

Objective Function	$n_{predator}$	$n_{best\ prey}$	f_x	x_1	x_2	Error
1. Ackley Function	1	1	-199.8862	0.0366	0.0346	0.1138
	8	6	-199.7902	0.0263	0.1624	0.2098
	10	6	-199.9042	0.0758	0.0136	0.0958
	12	6	-199.9754	0.0017	-0.0059	0.0246
	6	8	-198.7910	0.0085	0.3708	1.2090
	6	10	-199.8563	0.0004	-0.0421	0.1437
	6	12	-199.3842	-0.0086	0.3897	0.6158
	6	14	-199.9233	-0.0192	0.0264	0.0767
2. Bartels Conn Function	1	1	-1.00149	-0.00122	0.02426	0.00149
	2	6	-1.00018	0.00002	-0.01778	0.00018
	4	6	-1.00114	-0.00079	-0.02572	0.00114
	6	6	-1.00060	0.00058	0.00632	0.00060
	6	10	-1.00077	0.00067	-0.01496	0.00077
	6	12	-1.00071	0.00055	-0.03688	0.00071
	6	14	-1.00047	0.00244	0.03770	0.00047
3. Beale Function	1	1	-0.0168	3.4337	0.5911	0.0168
	6	6	-0.0195	3.4378	0.5913	0.0195
	8	6	-0.0197	3.4431	0.5901	0.0197
	10	6	-0.0207	3.4555	0.5935	0.0207
	6	6	-0.0195	3.4378	0.5913	0.0195
	6	8	-0.0207	3.4536	0.5939	0.0207
	6	10	-0.0188	3.4273	0.5898	0.0188
4. Bird Function	1	1	-106.7877	-1.5896	-3.1230	0.0262
	10	6	-106.7877	-1.5909	-3.1234	0.0261
	12	6	-106.7877	-1.5896	-3.1229	0.0262
	14	6	-106.7261	4.7151	3.1809	0.0355
	6	4	-106.7877	-1.5896	-3.1232	0.0262
	6	6	-106.7873	-1.5840	-3.1234	0.0257
	6	8	-106.7813	-1.5804	-3.1197	0.0197
	6	10	-106.7678	-1.5789	-3.1385	0.0062
	2	4	-106.7581	-1.6124	-3.1261	0.0034
5. Branin RCOS Function	1	1	-0.3978878	9.4242531	2.2481362	0.0000008
	2	6	-0.3978874	9.4247711	2.2495269	0.0000004
	4	6	-0.3980952	3.1363159	2.2663702	0.0002082
	6	6	-0.3978905	3.1417095	2.2476581	0.0000035
	6	4	-0.3978875	3.1417115	2.2491557	0.0000005
	6	6	-0.3978905	3.1417095	2.2476581	0.0000035
	6	8	-0.3978881	9.4256200	2.2507725	0.0000011
	2	4	-0.3978875	9.4249259	2.2501726	0.0000005
6. Camel Six-Hump Function	1	1	-1.031627982	-0.089356470	0.712623827	0.000027982
	2	6	-1.031628452	-0.089848143	0.712671260	0.000028452
	4	6	-1.031628445	-0.089886855	0.712669794	0.000028445
	6	6	-1.031627408	-0.089759567	0.713255139	0.000027408
	6	2	-1.031628444	-0.089863601	0.712626788	0.000028444
	6	6	-1.031627408	-0.089759567	0.713255139	0.000027408
	6	8	-1.031626822	-0.091070397	0.712279976	0.000026822

To be continued…

…Continuation

7. Egg Crate Function	1	1	-0.0168	3.4337	0.5911	0.0168
	2	6	-0.0195	3.4375	0.5924	0.0195
	4	6	-0.0199	3.4434	0.5924	0.0199
	6	6	-0.0195	3.4378	0.5913	0.0195
	6	2	-0.0197	3.4416	0.5902	0.0197
	6	4	-0.0199	3.4421	0.5916	0.0199
	6	6	-0.0195	3.4378	0.5913	0.0195
8. Leon Function	1	1	-0.00010	1.01009	1.02027	0.00010
	2	6	-0.00060	0.97543	0.95154	0.00060
	4	6	-0.00092	0.96975	0.94012	0.00092
	6	6	-0.00112	0.96560	0.93242	0.00112
	6	2	-0.00282	0.94678	0.89532	0.00282
	6	4	-0.00097	0.96898	0.93860	0.00097
	6	6	-0.00112	0.96560	0.93242	0.00112
9. Mishra 6 Function	1	1	-2.28389	2.89258	1.81567	0.00006
	4	6	-2.28392	2.88191	1.82755	0.00003
	8	6	-2.27126	2.93153	1.75056	0.01269
	12	6	-2.28395	2.8866	1.82699	0.00000
	6	2	-2.28395	2.88623	1.82338	0.00000
	6	8	-2.28002	2.91892	1.78515	0.00393
	6	14	-2.27294	2.92591	1.75466	0.01101
10. Shubert's Function	1	1	-186.7295	6.0841	0.4256	0.0012
	4	6	-186.7309	-6.4829	-5.8581	0.0002
	14	6	-186.7302	0.4249	6.0840	0.0005
	6	2	-186.4025	6.0790	-5.8470	0.3282
	6	6	-186.3530	-0.2082	0.4348	0.3777
	6	10	-186.7307	6.0838	-5.8579	0.0000
	8	10	-186.7295	6.0841	0.4256	0.0012

4. Conclusion

Optimisation problems may have multiple local and global optimums. In some cases, global optimum is far from the local solutions and being trapped in a local optimum is a challenge for metaheuristic algorithms. In this study, the PPA is being extended to nmPPA by assigning n of the solutions as predators and m of the solutions as best preys in which predators act in exploration and best preys do so in exploitation. This extension of nmPPA aimed to overcome the challenge of metaheuristic algorithm being trapped at local optimum and find multiple solutions at the same time.

Referring to the simulation results on selected ten benchmark function with various characteristics, it is observed that nmPPA does generates better solution compared to PPA. Results from nmPPA are closer to the exact global objective values and its global optimum points. Apart from that, nmPPA also enables the algorithm to locate more than one solution when dealing with multimodel function with several global optimum points.

Furthermore, nmPPA generally show a trend of increasing accuracy on either cases of fixed n, while increasing m or fixed m, while increasing n when dealing on unimodel function. However this trend is not shown when applying nmPPA on multimodel function with multiple global optimum as the nmPPA tends to explore the different peaks with high objective values instead of concentrating on only one peak to exploit the best objective value.

Acknowledgements

This work was supported by Fundamental Research Grant Scheme (FRGS) No. 203/PMATHS/6711319.

References

Ackley D. H. 1987. *A Connectionist Machine for Genetic Hill-Climbing.* Norwell, MA: Kluwer Academic Publishers.

Branin F. H. 1972. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development* **16**(5): 504-522.

Jamil M. & Yang X.-S. 2013. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation* **4**(2): 150-194.

Kirkpatrick S., Gelatt C.D. & Vecchi M. P. 1983. Optimization by simulated annealing. *Science* **220**(4598): 671-680.

Lavi A. & Vogl T. P. 1966. *Recent Advances in Optimization Techniques.* NewYork: John Wiley & Sons Inc.

Mishra S. K. 2006. Global optimization by differential evolution and particle swarm methods: Evaluation on some benchmark functions. http://mpra.ub.uni-muenchen.de/1005. (6 July 2015)

Molga M. & Smutnicki C. 2005. Test functions for Optimization Needs. http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf. (6 July 2015)

Tilahun S. L. & Ong H.C. 2015. Prey-Predator Algorithm: A new metaheuristics algorithm for optimization problems. *International Journal of Information Technology & Decision Making* **14**(6): 1331-1352.

Yang X.-S. 2010. *Nature-inspired Metaheuristic Algorithms.* Frome, UK: Luniver Press.

School of Mathematical Sciences
Universiti Sains Malaysia
11800 USM Penang
MALAYSIA
E-mail: hcong@usm.my, ngchiajiun@gmail.com*

*Corresponding author