Indian Journal of Geo-Marine Sciences
Vol. 40(2), April 2011, pp. 250-266

A simulation based fly optimization algorithm for swarms of mini autonomous surface vehicles application

Zulkifli Zainal Abidin[1], Mohd Rizal Arshad[1], Umi Kalthum Ngah[2]
[1]Underwater Robotics Research Group (URRG), School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, 14300, Pulau Pinang, Malaysia
[2] Imaging & Computational Intelligence Research Group (ICI), School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, 14300, Pulau Pinang, Malaysia
[Email: [1]rizal@usm.eng.my, [2]zzulkifli@iiu.edu.my]

Received 23 March 2011; revised 28 April 2011

Present paper intends to provide a detailed description of a new bio-inspired Metaheuristic Algorithm. Based on the detailed study of the Drosophila, the flowchart behaviour for the algorithm, code implementation, methodologies and simulation analysis, a novel Fly Optimization Algorithm (FOA) approach is presented. The optimal simulation parameters can be used for the real application. FOA is suitable for applications that need a small number of agents; in the range of 8 to 24 only. The objective of the simulation is to understand the effect of the algorithm parameter on searching pattern strategy, as well as the possibility and the effectiveness of the proposed technique for the Swarm of mini Autonomous Surface Vehicles' (ASVs) application.

[**Keywords**: Metaheuristic, Fly Optimization Algorithm, Drosobots, Drosophila]

Introduction

The term "meta-heuristic" was suggested by Fred Glover[1]. Metaheuristics is a top-level general strategy which guides other heuristics to search for feasible solutions in domains where the task is hard and difficult. Generally, it is applied to problems classified as NP-Hard or NP-Complete according to the theory of computational complexity. However, it could also be applied to other combinatorial optimization problems for which it is known that a polynomial-time solution exists but is not practical. Metaheuristics is one of the best methods available for "good enough/fast enough/cheap enough" solutions. Some examples of animal inspired metaheuristics[2] adopted are the Ant Colony Optimization (ACO)[3], Particle Swarm Optimization (PSO)[4], Monkey Search[5], Bee Algorithm (BA)[6], Firefly algorithm (FA)[7], and Artificial Bee Colony Algorithm (ABC)[8]. Animals live in harmony and help to sustain the environment's lifecycle. Without human intervention, these creatures carry out their 'spontaneous routine' jobs and contribute towards balance in nature. Although their brains are very small compared with the size of the human brain, amazingly, the ways in which they arrive at a certain decision are very impressive and intriguing, at times surpassing that of a human being.

Unlike the Multi-Autonomous Ariel Vehicles trajectory planning using Ant Colony Optimization (ACO)[9] approaches, our project objective is mainly to perform contour mapping of lakes[10-11] and ultimately determine the deepest location within the vicinity of the lake in a short period of time, through the use of 16 agents or populations. When compared with classical benchmark functions, it was found that the existing algorithms needed numerous numbers of agents (population of $n>30$) when deployed in real world applications[12]. On the other hand, observing the fruit fly and its foraging pattern provides as an interesting alternative. Though small in size, the intelligence that they exhibit, while searching for food and mates, may be adopted in finding an optimal path. The most important factor is that they normally move in a small number. Therefore, here we propose the "Fruit Fly" as agents and their searching patterns as an alternative algorithm[13]. Some classical benchmark functions have been used to verify this finding[12].

Present study is organized as follows: Section 2 will review the biological background of the fruit flies while foraging and mating; Sections 3 and 4 will explain their behaviour modeling and algorithm implementation; Sections 5 and 6 will give the experimental results and discussion analysis while

Sections 7 and 8 will provide an overview of the field testing implementation and the conclusions derived.

Biological Background

Current trends in research development are focusing more on the *Drosophila Melanogaster* (scientific name for the fruit fly) species[14-16]. Therefore, the initial ground work in this study has also singled out and focused on this particular specie. Individual flies vary in body length from 1 to over 20mm (Fig. 1). The females of most species insert their eggs in living, healthy plant tissues. The larvae live and feed in the stalks, leaves, fruits, flower heads, or seeds[17]. It would be almost impossible to list out the number of investigations associated with Drosophila's family tree. Originally, work involving studies on the fruit fly was confined to the field of genetics; for example, leading to findings that were related to proteins and the study of genetic inheritance[18]. More recently, work involving their study has evolved and is used mostly in developmental biology, neural development, locomotion, and even in 'NASA space'[19].

The main idea behind this algorithm (Fig. 2) is based upon the fruit fly's biological behavior: 1) The fly hunts for food and a mate within duration of one to two months lifespan, 2) It moves randomly with Lévy flight motion[16,20-22] 3) It smells the potential location (attractiveness), 4) It would then taste. If it is not to its liking (fitness/profitability), it rejects and goes to another location. To the fly, attractiveness is not necessarily profitable[23]. 5) While foraging our mating, the fly also sends and receives messages with its friends about its food and their mates[14-15,24-25]. The main steps of the fly behaviour algorithm are given as in the flowchart of Fig. 2. When a fly decides to go for hunting, it will fly randomly (with Lévy flight motion) to find the location guided by a particular odor. While searching, the fly also sends and receives information from its neighbours and makes comparison about the best current location and fitness. If a fly has found its favourable spot, it will then identify the fitness by taste. If the location no longer exists or the taste is 'bitter', the fly will go off searching again. The fly will stay around at the most profitable area, sending, receiving and comparing information at the same time. The total number of flies depends upon the number of sources. However, since most of the flies are near to the food source location, the next generation of flies is considered to be already close by to the potential food location.

The Fly Optimization Algorithm (FOA) is a new animal-inspired algorithm based on Drosophila, a specie of fruit fly. Many algorithms have been

Fig. 1—The Fruit Fly

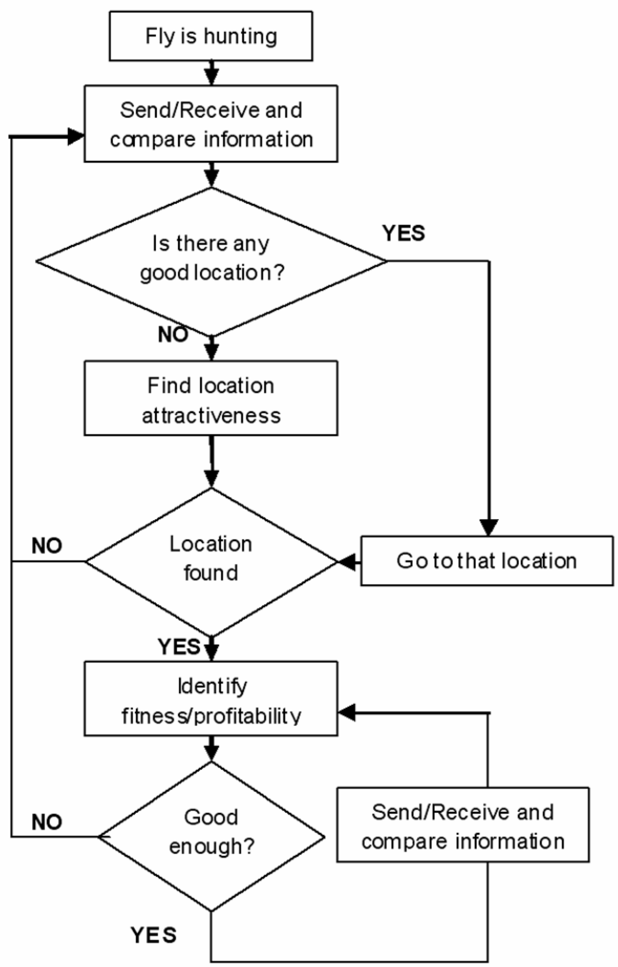

Fig. 2—The flowchart of fruit fly behavior while searching.

developed with each algorithm having its own advantages and disadvantages. FOA is developed with a different purpose; emphasizing not only on the swarming behaviour itself but also on the application involved. Thus, we are also considering whether a real agent can achieve the particular optimization movement or not. Although the development phase of the algorithm is still undergoing, the basic technical principle has been largely formed. Further developments will focus on the effect of the identified parameters on the results achieved.

Code Implementation

In contrast to the other existing algorithms, the fly algorithm will compute the best surrounding direction before moving towards that direction. Similar to the other algorithms such as the Bee Algorithm[12], shrinking is ultimately important in this algorithm. Currently, this new algorithm is focused upon examining the best shrinking method. Although this algorithm is mainly based upon gradient information, randomization is imposed in order to solve the local optimum problem. Its capability to solve problem with multiple peaks will also be harnessed.

As the purpose of this algorithm is mainly for swarming robotics, this study focuses on the possibility of real swarming with the utilization of sensors. The existing algorithm concentrates purely on fast varying motions that can be performed by the organism or applied to the nature of particles[12]. Although FOA is still based on the motion of the fly, it has also introduced features that can actually be performed by slow varying agents such as an autonomous surface vehicle (ASV). A real agent will collect data along the path. In our case, it is considered as a crucial point and must be taken into consideration. Thus, this algorithm is based on the scenario where a fly is actually collecting data in its path and it changes directions according to stochastic conditions. Thus, under real circumstances, each agent would be able to investigate each peak in a particular confined area. The searching process also can be described in the form of pseudo code as below:

- Initialization using Lévy Flight motion
- Choosing the best location
- While (terminating condition is not met)
- Examine surrounding points and identify best heading direction (find location attractiveness, simply smelling)
- Examine points on that direction with different distance (go to location found). In this algorithm, this is known as shooting process
- Select the best point to be next reference point and loop again
- Terminate while location is in range

FOA process begins with initialization by using Lévy Flight motion. Each fly will dispatch and find its own current best location. In the mean time, each fly also "smells" if there is any other source of food better than the points it visited and eventually examining the best heading direction for the next iteration process. Like the "fruit flies", they share information among them and stop at the location where the targeted location is considered as the most profitable among them. The detailed implementation of this pseudo code is described in the next section.

The surrounding of the known best points is then explored. Based on simulation process, in order to yield sensible results, the smelling process must involve 10 flies or more. A circle with a small radius is set around the best known point (Fig. 3). The flies are distributed on the perimeter of the circle at certain angles. The best direction will then be chosen. This is the basic idea of the smelling algorithm. After selecting the best direction, a tracking process can be conducted along the direction that is selected. Instead

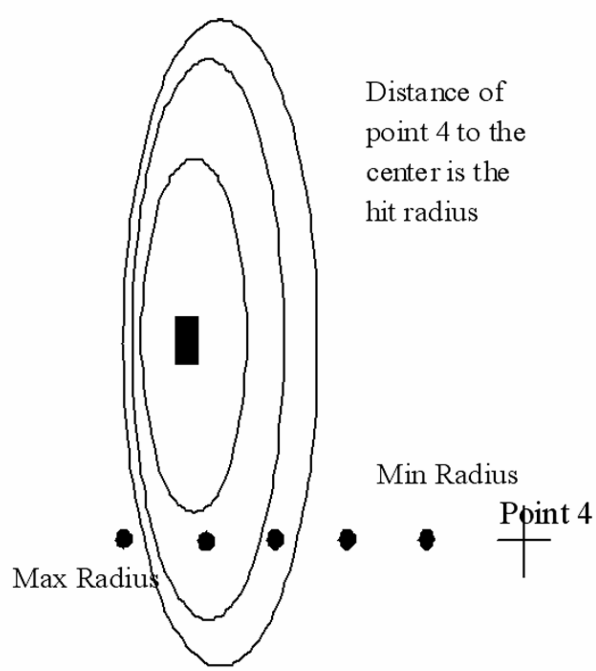

Fig. 3—The Shooting Process

of having the varying angles as in the smelling process, the tracking process will have its radius varying from the known best point for the flies involved. The point which is the best and better than the known best point will be selected as the new known best point.

Materials and Methods

FOA will always start off with an initialization. To ensure a good starting point, the fly will be sent to each sector of the environment. Good location will then gain more visits of the fly. To prevent convergence on local extremes, the fly will be distributed and dispatched again according to the Levy's flight (which is the model of the movement of the fly). After the initialization process, the known best location will be selected for the smelling process. The smelling process is used to detect the path which will yield better direction of attractiveness. The best direction will be chosen for further explorations. This is known as the tracking process. The tracking process will be able to identify the point where the direction is no longer suitable for further explorations. The smelling and tracking process will be conducted in each of the iterations until terminating conditions are met. However, after 30-50 iterations, randomization is made to occur again in order to prevent convergence at local optimums. All the tests are conducted using 16 flies.

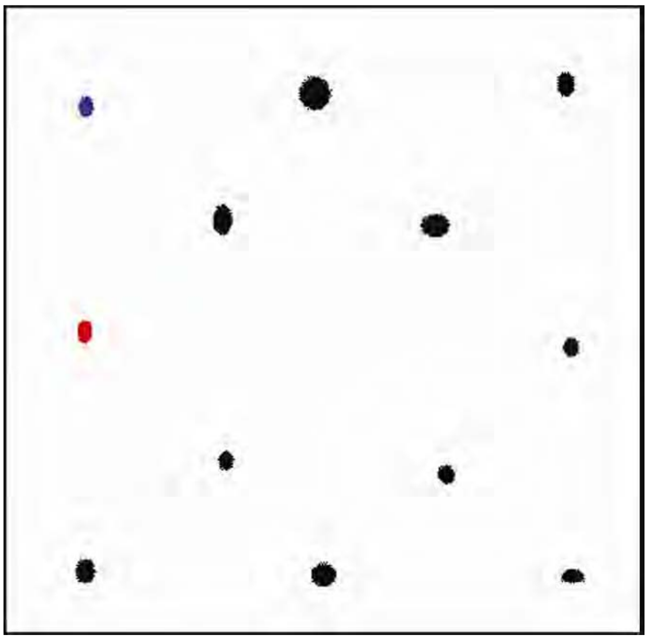

Fig. 4—1st step of Initialization.

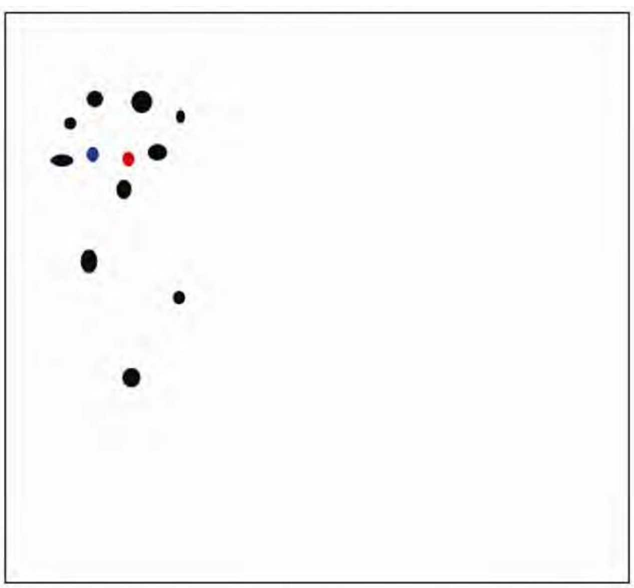

Fig. 5—The 2nd step of initialization.

Initialization

The initialization of FOA is an imitation of the fly swarm behaviour itself when the swarm is exploring the designated area for the first time. The flies come from every direction towards the targeted area (in the real world would be the fruit). By choosing the best point, further randomization around the best sector is then made. This attempt will enhance the chances to obtain points nearer to the global optimum. Finally, randomization is to be made in all the areas but is centered at the known best point. Indeed, initialization can be done by pure randomization but this approach would generally lead to a slower converging process or to be trapped at local optimum points. Initialization by 8 flies near to the boundary and another 4 flies near to the center might give a good starting point[12]. Further randomization around the first few best points will further enhance the probability to get to the nearer points to the global optimum. However, randomization which involves all the areas cannot be spared or the chances of being trapped in local optimum will increase.

Fig. 4 shows the 1st step of Initialization with blue being the best among the 12; red, the second among the 12 flies. Next, the flies will flock around two best known areas (Fig. 5). The best will gain more attention. Notice that fly algorithm does not require a fly to be stationed at the former known best point. At this point, "the blue" being the new known best point. The 3rd step of initialization process in Fig. 6 shows that, although the randomization is on the entire sector, it is still centered on the known best point. However, a lower value of alpha should be set for Levy Flight motion with a larger variance.

For multiple-peak problems, more evaluations must be made for the initialization. The second and third steps can be done for a multiple of times. The second step should be done based on the Levy Flight motion with the value alpha nearer to 2 and a smaller value for variance. Moreover, a strict bound around the known best and the few best points must be set.

Smelling Algorithm

After a series of controlled randomizations, the flies will focus on the known best point. The surrounding of the known best points will be explored. This is the basic idea of the smelling algorithm. However, a form of direct application as such yields poor results. Firstly, the radius must not be fixed all of the time. A shrinking process must be properly applied onto the radius. Due to the fact that the accuracy is set to a certain resolution, the smelling radius will fall to a constant and not to zero. Thus, exponential approximation could be made but it must be expressed in a piece-wise manner. The first part illustrates the falling value of the radius at the onset to half of its initialized value. The second part illustrates the falling value of the radius from half the initialized value to its minimum.

$$\textit{smelling radius, } R = 1 - 0.5e^{\frac{2(i-B)}{D_1}} \quad \textit{when } i < B \qquad \ldots (1)$$

$$R = R_{\min} + (0.5 - R_{\min})e^{-\frac{2(i-B)}{D_2}} \quad \textit{when } i \geq B \qquad \ldots (2)$$

Where D_1 and D_2 are shaping factors. The larger the shaping factors, the slower the converging of the exponential term. Secondly, the radius must bounce within a predetermined range. It is found that a bounce of 30% above and below the smelling radius yields better result. This will eliminate the effect of faster or slower shrinking rate of smelling radius. Thirdly, an offset angle must be accounted. If there is no random offset angle, only certain angles will be examined, and this will reduce the converging rate. However, the recommended offset angle is within the angles in between the flies set earlier.

Thus, a general equation for smelling process would be:

$$x_k = x_{best} + R(A + B(Rand(1)))\cos(\frac{2k\pi}{n} + (Rand(1))\frac{2\pi}{n}) \qquad \ldots (3)$$

$$y_k = y_{best} + R(A + B(Rand(1)))\sin(\frac{2k\pi}{n} + (Rand(1))\frac{2\pi}{n}) \qquad \ldots (4)$$

Where n is the number of flies that are involved in the smelling process, k is the number of flies among the involved flies while A and B are constants. A good estimation for A and B would be $A + \frac{B}{2} = 1$ with both A and B around 0.6.

Tracking (Shooting) Process

After selecting the best direction, tracking process can then be conducted. Instead of having a varying angle as in the smelling process, the tracking process will instead have its radius varying for the flies involved. For this process, the number of the flies should be more than 16. Among these flies, the best fly will be chosen. The radius of the chosen fly is known as hit radius. Fig. 7 illustrates the fact that shooting at the exact direction misses a chance to get nearer to the global best.

Like the smelling radius, the maximum tracking radius and the minimum tracking radius shrinks over iterations. Logarithm scale is a good choice for the distance between tracking flies since a large leaping is unlikely to happen if compared to the points near to the known best point. It must be realized that known best only updates during the tracking process. Hence, the center will remain at the same point as long as no better solution is available. Defining *UL* to be the initial minimum radius, *LL* as the final minimum radius, *UU* as the initial maximum radius while *LU* is the final maximum radius, length is the number of iterations for the radius to settle down to the final

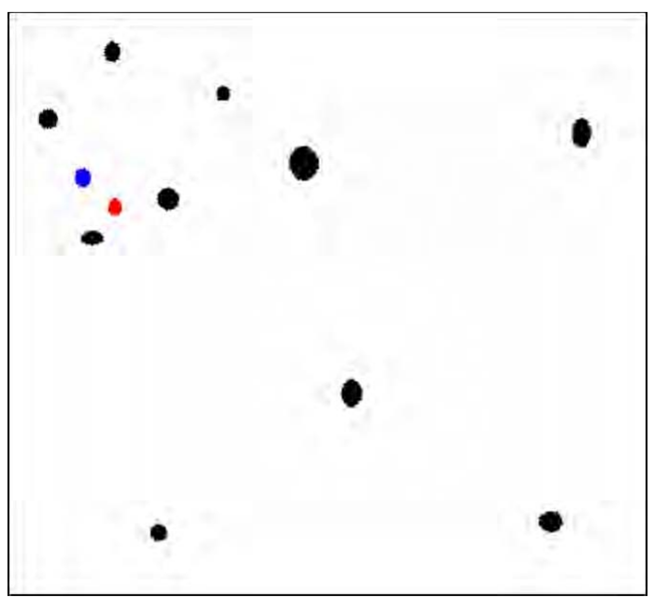

Fig. 6—The 3rd step of initialization.

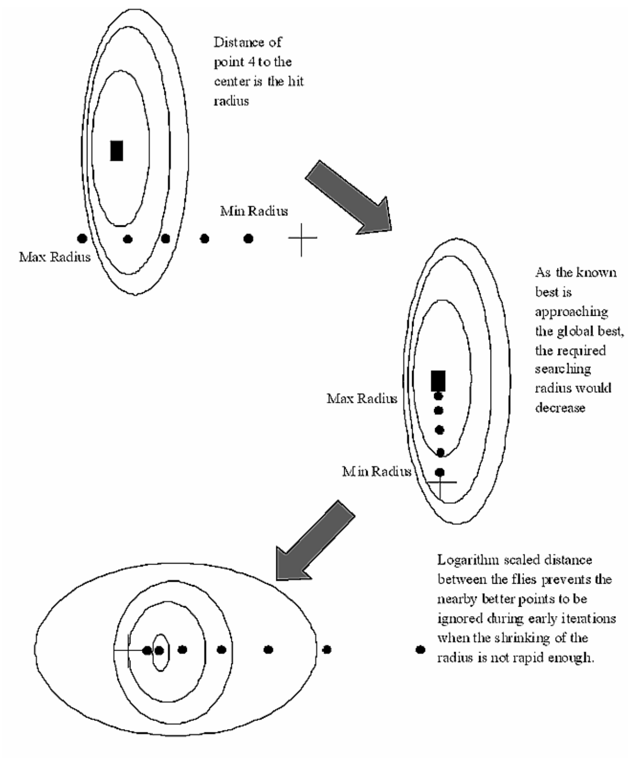

Fig. 7—Tracking process and shooting direction

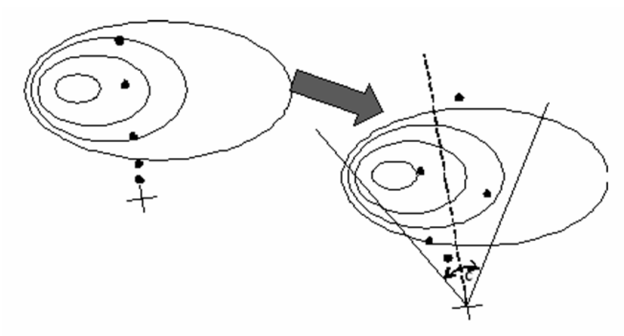

Fig. 8—The Shooting Range

radius, i is the number of current iterations, u and l are the current maximum radius and minimum radius respectively,

$$u = (-UU + UL)/(length - 1) * (i - 1) + UU \qquad \ldots (5)$$

$$l = (-LU + LL)/(length - 1) * (i - 1) + LU \qquad \ldots (6)$$

Defining $r(k)$ to be the tracking radius of fly k, *leng* be number of flies,

$$r(k) = 10\hat{}(\,(log\ u\text{-}log\ l)/(leng\text{-}1)*(k\text{-}1)+l\,) \qquad \ldots (7)$$

Due to the variation during the initialization process, the shrinking should be done at variable rate. Hence, if the hit radius decreases, the shrinking should be quickened. The reverse should also be true. However, shrinking must be controlled. Therefore, a shrinking scale is introduced. The shrinking scale is reduced (normally 0.975 of its original value) only when the hit radius is smaller than a very small threshold at 3 times the resolution. If it is not less than the threshold, the scale should increase at a very small rate (normally at 1.0005) so that the radius can be maintained. However, tracking in exact directions might end up giving poorer results. This is because the gradient is not always constant. Thus, randomization in terms of angles must be made. Based on the simulated result, the estimated range of angles should be in between 20 to 40 degrees at each side in order to minimize overshoot and undershoot.

Fig. 8 illustrates shooting at a suitable range of C will enhance the chances to get nearer to the global best point. Most problem domains do not exhibit a constant gradient. Thus, randomization in shooting angles will certainly help in terms of converging speeds. Undoubtedly, there would also be chances of obtaining poorer results. However, the probability for such a thing to happen declines when the number of flies involved in the tracking increases. The position of fly number k is:

$$X(k) = posx + scale * r(k) * \cos(\theta + (-\frac{1}{2} + rand) * C; \ldots (8)$$

$$Y(k) = posy + scale * r(k) * \sin(\theta + (-\frac{1}{2} + rand) * C; \ldots (9)$$

Where *posx* and *posy* are the coordinates of center, θ is the direction obtained from the smelling process while C is the angle bound in radian.

Randomization in the Midst of Smelling and Tracking

The purpose of randomization is to prevent the best known point to be trapped at the local minimum. In fact, the frequency of randomization is not as rapid as during smelling and tracking. Randomization should be started after the first 30-50 iterations and conducted at every 5 to 8 iteration after the first 30-50 iterations. The frequency of randomization should be increased when a number of local optimum appears. Since the main objective is to find the global optimum and then venture beyond the best known area, the alpha of Levy motion can now be decreased.

Simulation

All the tests in the simulation are conducted using 16 flies. With respect to the surrounding area, 10 evaluations will be used to determine the direction (smelling process) while 16 to 21 evaluations will be used to determine how far the path will be in that direction (shooting process). In other words, on average, 2 evaluations will be made by each fly in each of the iteration. Both the smelling radius and shooting radius will decay with iterations. Thus, these two parameters are crucial in determining the results. The number of evaluations in each shooting process is also crucial in determining the converging speed.

The tests are made based on several benchmark functions for which their equations are listed in Table 1. De Jong's function is a maximization problem while the rest of the functions are minimization problems (Fig. 9). Rosenbrook's function has a long ridge and tends to cause initialization yields point far from the optimums (Fig. 10). It is actually the inverted version of De Jong function but with smaller range. Both functions have 3 peaks and optimum point at (1, 1). Only the best point from the initialization will be chosen for further exploration. Goldstein (Fig. 11) and Martin were chosen as the third and forth functions for testing the algorithm. The test functions and their optima are shown in Table 1 and Table 2.

Results and Discussions

Although implementation and application aspects are the motivations behind this algorithm

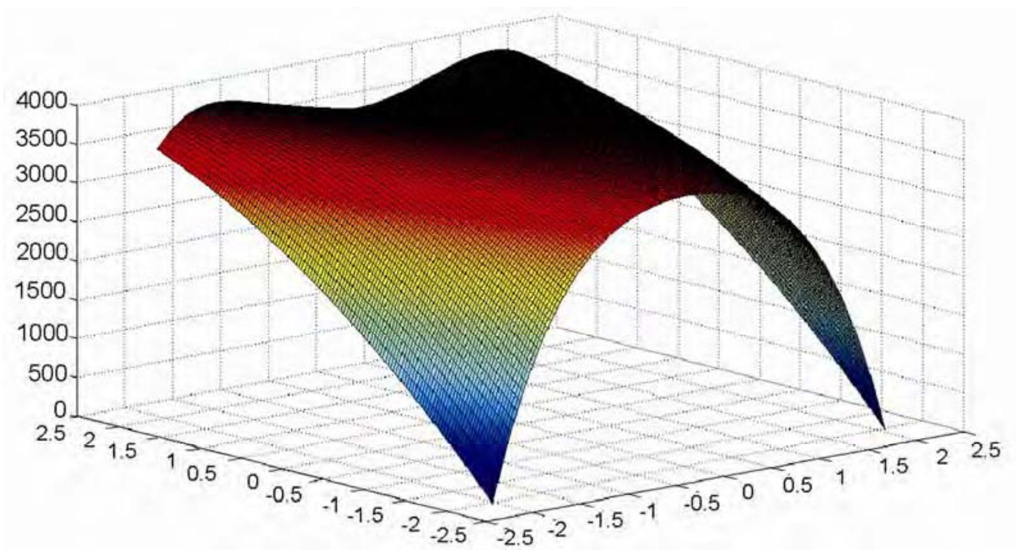

Fig. 9—De Jong function with a long ridge at the middle and it tends to trap the fly at the beginning iterations.

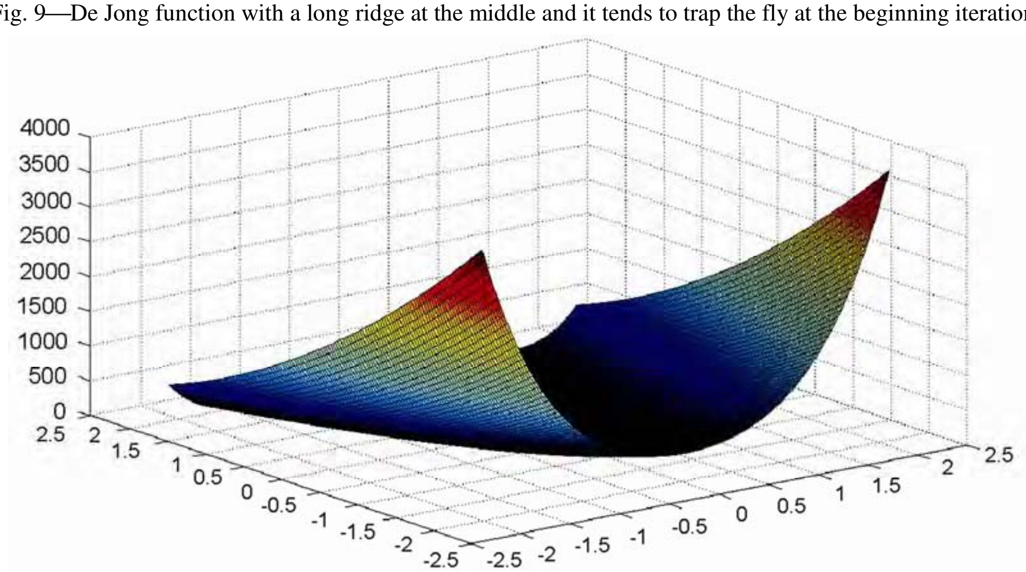

Fig. 10—Rosenbrook's Function which is the minimization problem.

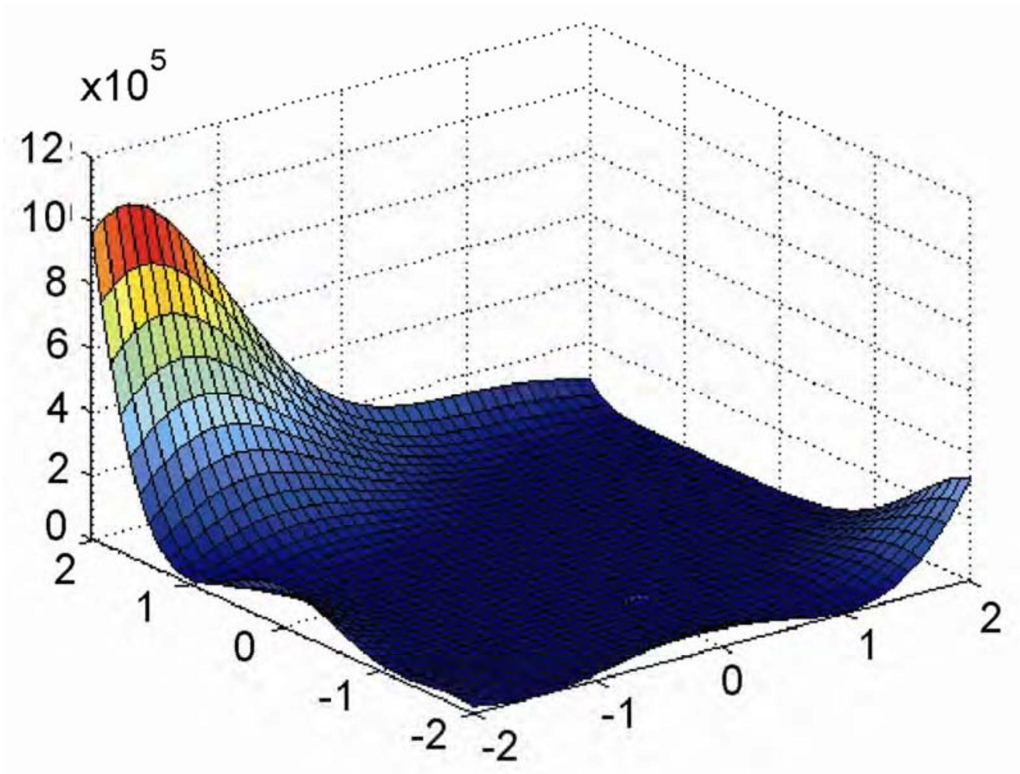

Fig. 11—3D surface of Goldstein Problem

Table 1—Boundary and equation for each test case.

Function Name	Boundary	Equation	
De Jong	[-2.048, 2.048]	Max	$3905.93 - 100(X^2 - Y)^2 - (1-X)^2$
Rosenbrook	i) [-1.2,1.2] ii) [-10,10]	Min	$100\,(X^2 - Y)^2 + (1-X)^2$
Goldstein	[-2,2]	Min	$((19-14x+3x^2-14y+6xy+3y^2)(x+y+1)^2+1)\ X$ $((18-32x+12x^2+48y-36xy+27y^2)(2x-3y)^2+30)$
Martin	0, 10]	Min	$\dfrac{(x-y)^2 + (x+-10)^2)}{9}$
	[-10, 5]	Min	$(y-0.129080578x^2\ 1.590909090x-6)^2 + 9.602272727\cos x + 10$

Table 2—Expected Result of Each Benchmark Functions.

Function Name	Expected Location	Optimum Fitness
De Jong	(1, 1)	3905.93
Rosenbrook	(1, 1)	0
Goldstein	(0,-1)	3
Martin	(5,5)	0
Branin	(π,2.275) (3π,2.47175) (-π,12.275)	0.3977

development, accuracy is still an important point. The test has been done on De Jong function with mean evaluation of 759.839 times (Table 3). Although statistically better than the previous existing algorithms, the process is skewed to the right (Fig. 12). The standard deviation is 460.14 evaluations and typically the best known fitness reaches the optimum point within the third iteration process (Fig. 13). These experiments are done based on a 0.001 bound (Table 4) on the location basis and the algorithm is run for 1000 times.

The test also has been done on the Rosenbrook's function. In the first case, the boundary is from -1.2 to 1.2 for the x and y dimension. The mean evaluation is 500 times while the standard deviation is 301.74 evaluations. Typical best known fitness is found on the 15th iteration process (Fig. 14). For the second case, when the boundary is in the range from -10 to 5 for each dimension, the average evaluation number is 1098.3 while the standard deviation is 490.8707. This

Table 3—The number of evaluations for the different test cases.

Function Name	ANTS[12]	Bee Algorithm[12]	Firefly Algorithm[26]	Fly Optimization Algorithm
De Jong	6000	868	730	766
Rosenbrook	i) 6842	i) 631	i) -	i) 500
	ii) 7505	ii)2306	ii)2923	ii) 1109
Goldstein	5330	999	-	310
Martin	1688	526	-	380
Branin	1936	1657	-	1468

Table 4—The Main Parameter for Each Benchmark Function

Function Name	Initial Smelling Radius	Initial Shooting Radius	Shooting Evaluation
De Jong	0.0018	0.26	21
Rosenbrook	i) 0.0008	i) 0.036	i) 16
	ii) 0.0008	ii) 0.30	ii) 21
Goldstein	0.0009	0.3	21
Martin	0.0008	0.18	21
Branin	0.0008	1.2	21

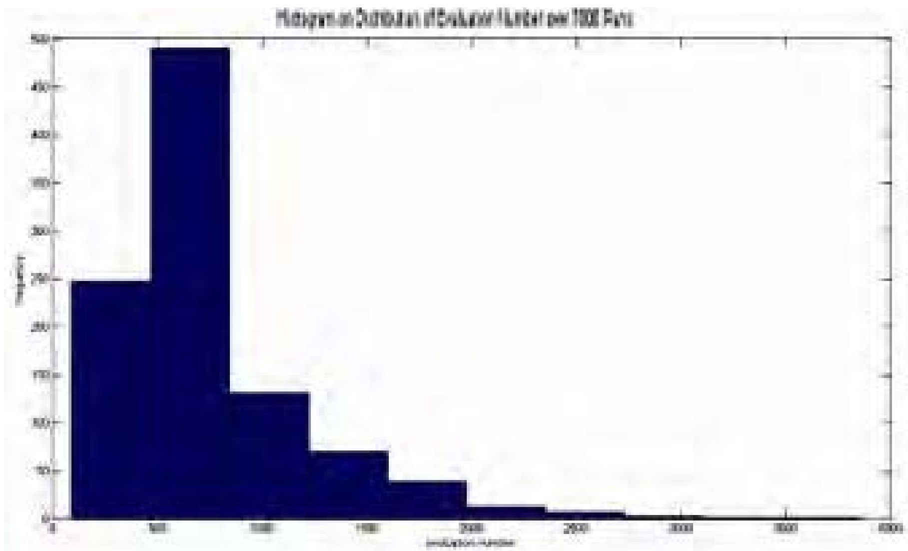

Fig. 12—Histogram on Distribution of Evaluation Number over 1000 runs for De Jong Function

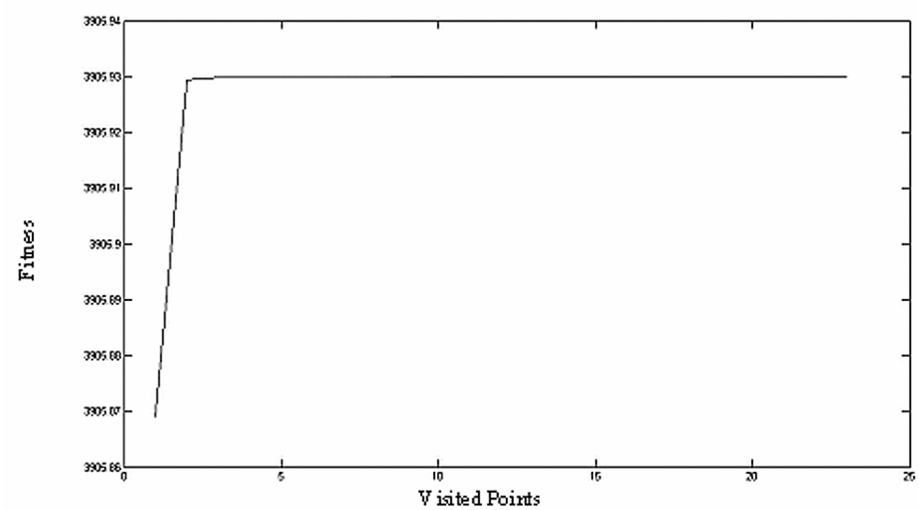

Fig. 13—Typical best known fitness at particular iterations for De Jong function.

shows that for the De Jong function (Fig. 12) and the Rosenbrook functions (Fig. 15 and Fig. 16), the algorithm shows similar characteristics since all the histograms are skewed to the right. This is due to a long ridge shape of contour. In the rest of the case, the contours have round shapes and thus show a normal distribution in the histogram.

For the Goldstein function, the average number of evaluations is 310.048, while the standard deviation is 66.3408. Goldstein function has no long ridge and

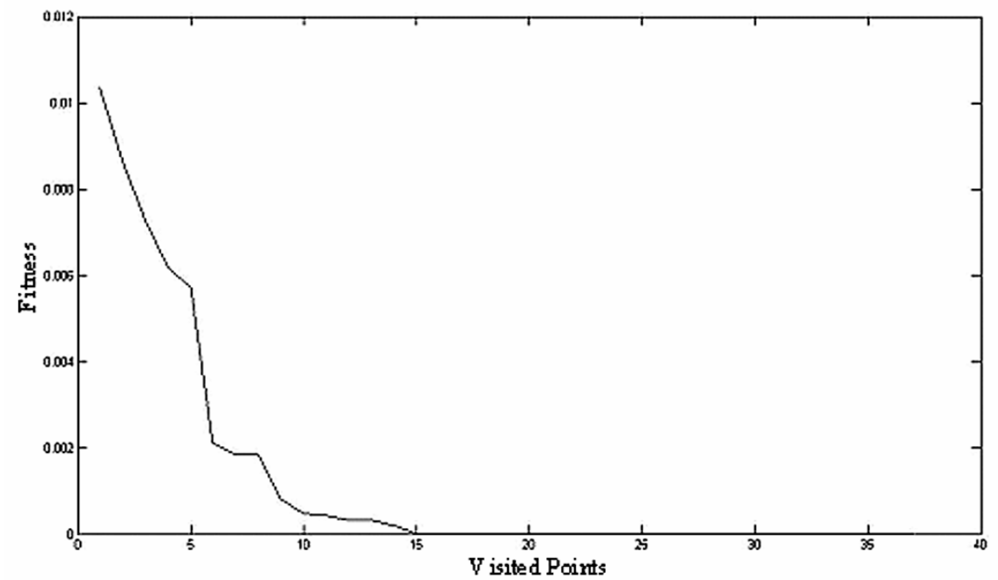

Fig. 14—Typical best known fitness at particular iteration for the Case 1 of Rosenbrook's function.

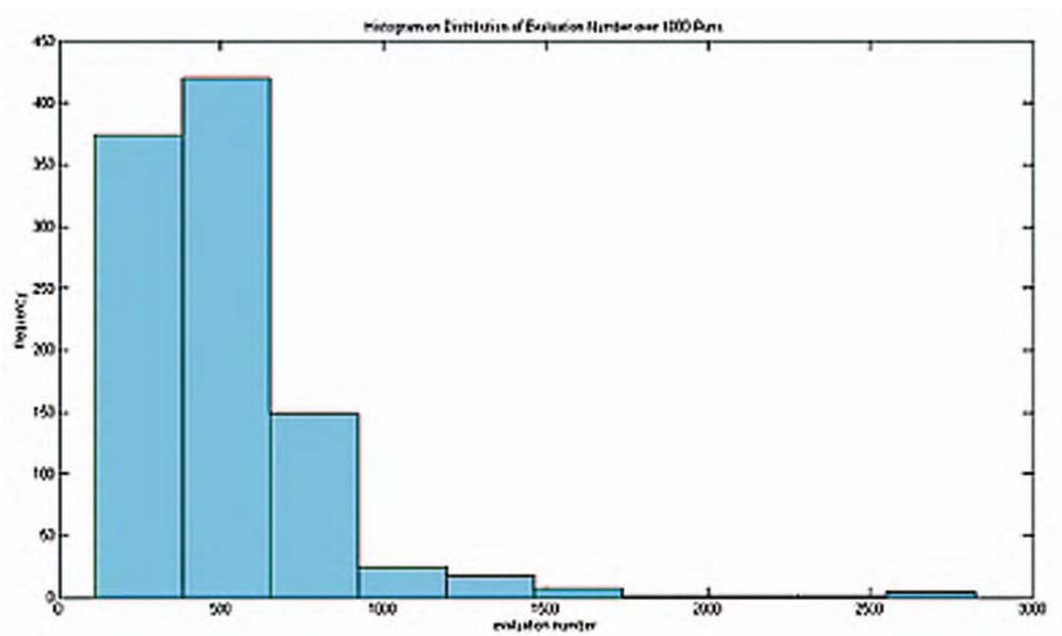

Fig. 15—Resultant Histogram in Case 1 of Rosenbrook's function

hence the process is normally distributed in terms of the number of evaluations (Fig. 17) and found its typical best known fitness on starting from the 4th iteration process (Fig. 18).

As for the Martin function, the average evaluation is 380.077 times while the standard deviation is 100.418 times (Fig. 19). As for the case of the Branin's Function, the mean number of evaluations is 1467.9 while the standard deviation is 435.2853 (Fig. 20).

The Fly Algorithm is designed for a small number of agents. This is because in reality, the number of agents is limited. To investigate the effect of the number of agents on the number of evaluations, the case 2 of Rosenbrook's function is repeated for 8 and 24 agents. For the case of 8 flies, the average number of evaluations is 1355 while the standard deviation is 788 (Fig. 21).

The results show that the number of flies must be optimized. When the number of flies is as small as 8,

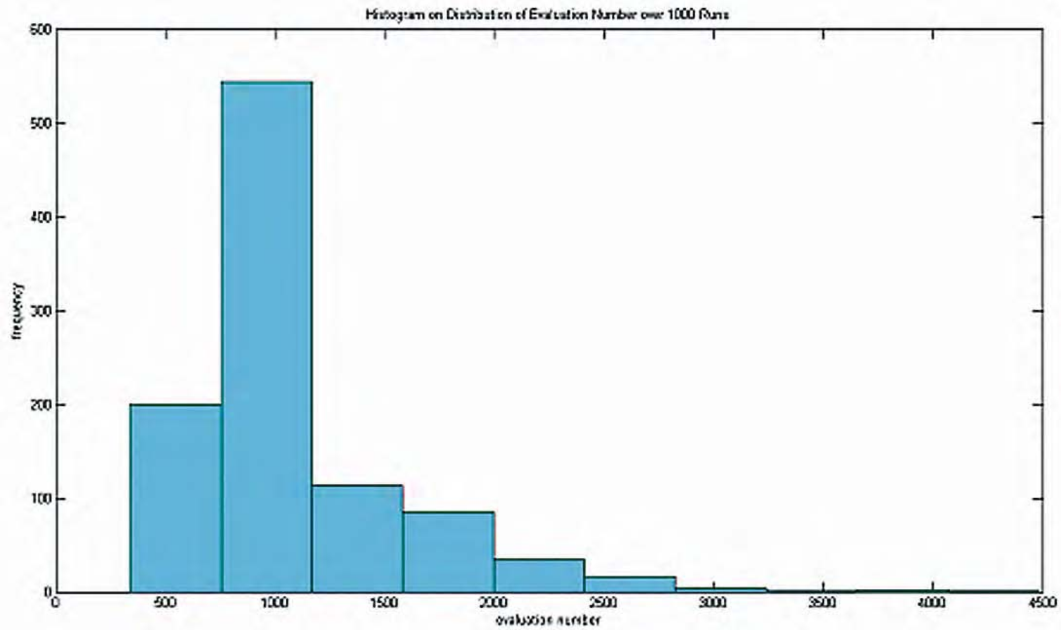

Fig. 16—Resultant Histogram in Case 2 of Rosenbrook's Function

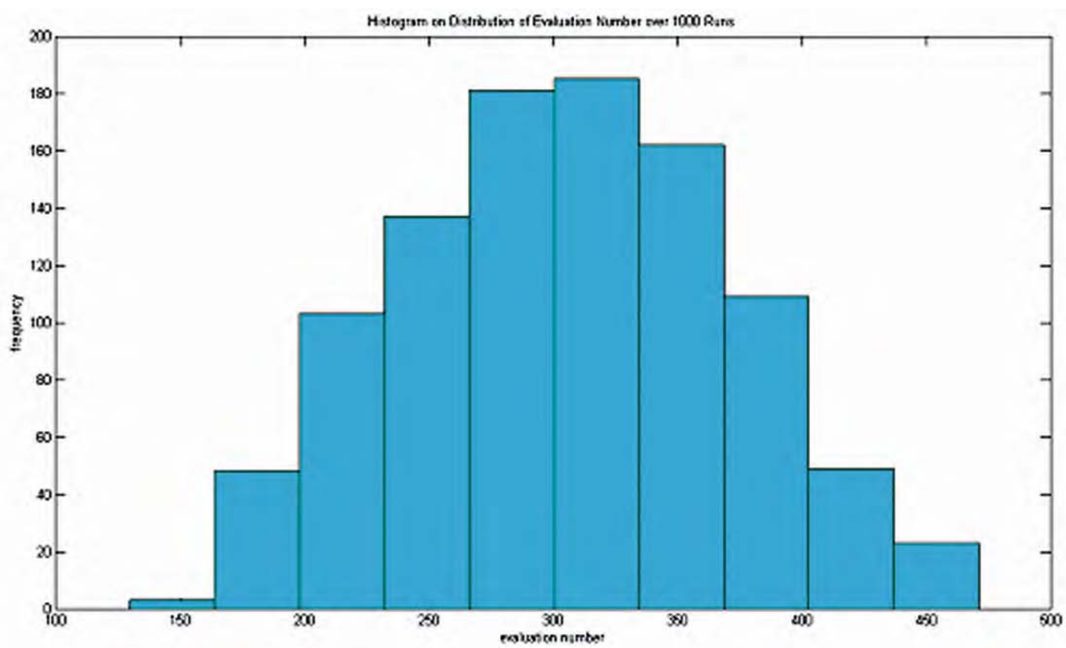

Fig. 17—The resultant Histogram in Test on Goldstein's Function

the number of evaluations for each shooting process is reduced, thus the average is as high as 1355. For the case of 16 flies, the average number of evaluations is 1109 while the standard deviation is 507 (Fig. 22). However, when the number of flies is increased to 24, the redundant evaluation in each shooting process is increased (Fig. 23). In this case, the average number of evaluations is 1358 while the standard deviation is 517. In conclusion, the standard deviation increases when the number of flies increases. However, further increment lead to more redundant evaluation, the standard deviation is maintained at the same level.

Potential Application of FOA with Drosobots Project

Prior to the development of the algorithm[27], we constructed a group of 8 simple ASVs (Fig. 24). This

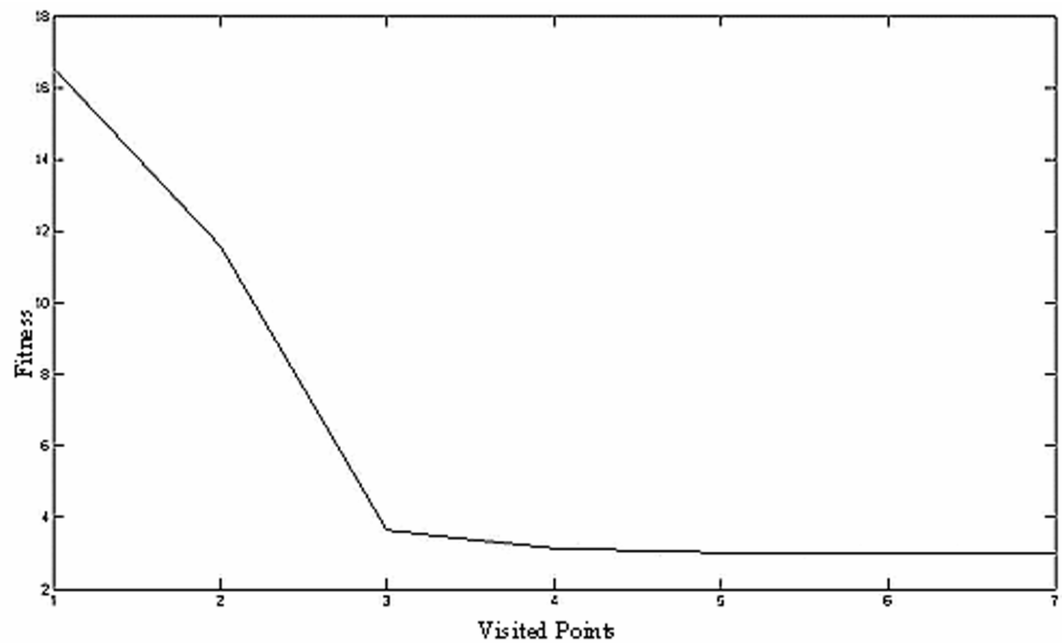

Fig. 18—Typical Convergence in Goldstein Case.

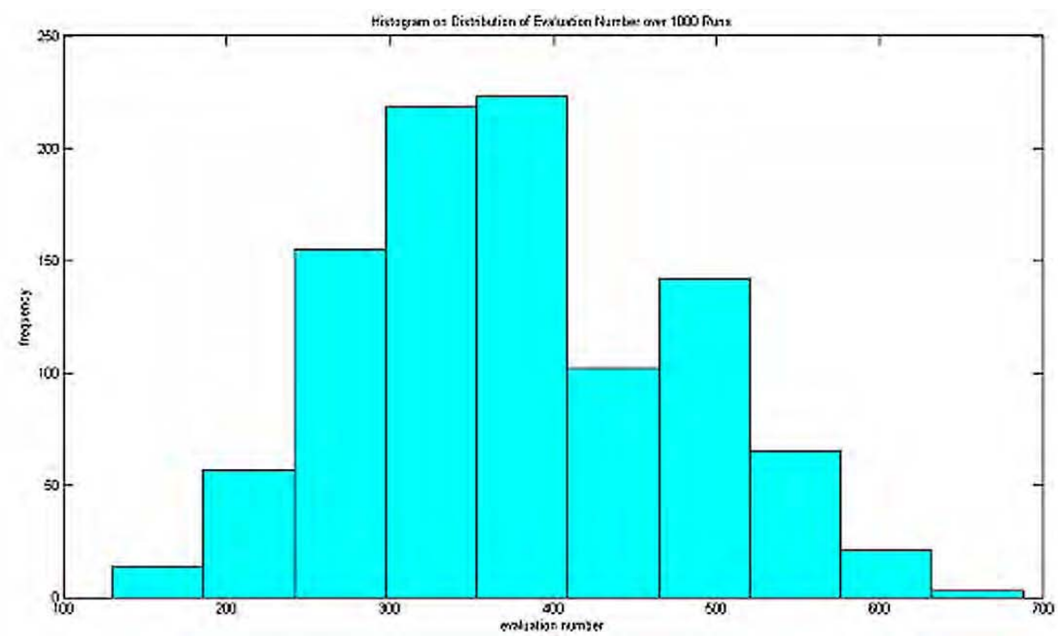

Fig. 19—The Resultant Histogram of Martin's Function

is done in order that, the practical benefits of the algorithm and its approach may be demonstrated in a real application. Lake mappings using cooperative multi-agent robotic system are practically non-existent. Therefore, this provided us the niche to indulge in this additional venture i.e. to further explore the usage of underwater robotics technology as swarming agents to serve the FOA objective. The overview of *Drosobots* project is shown in Fig. 25. By using a 900 Mhz RF transceiver, these units are able to communicate amongst themselves and also a central station via distributed control architecture system with a maximum distance of 32 km (clear line of sight). These units are designed to evaluate swarming algorithms in calm, open water environment equipped with active depth transducers

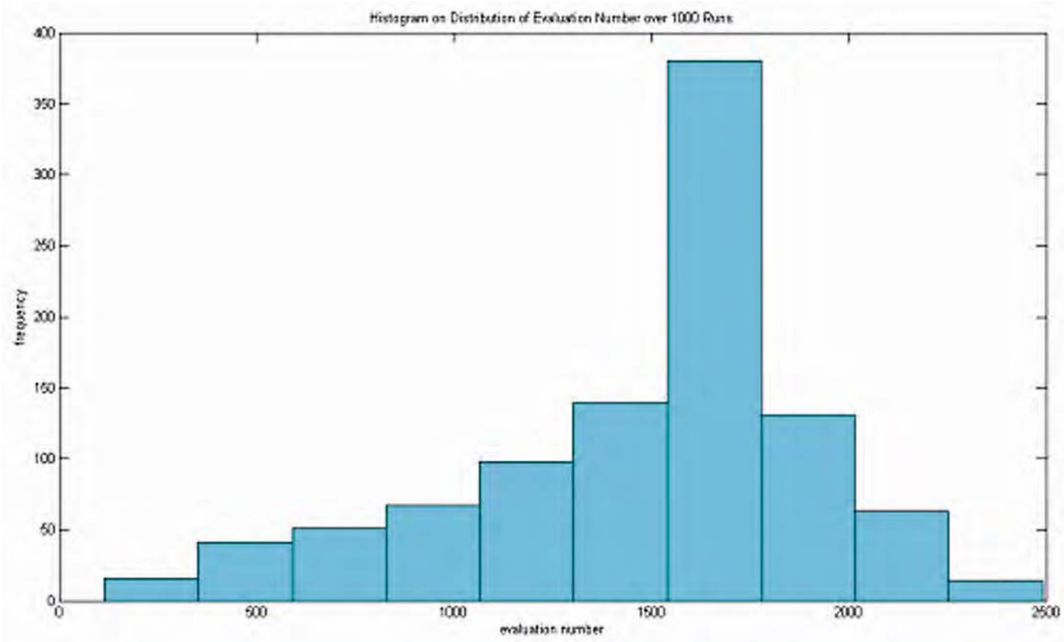

Fig. 20—The Resultant Histogram of Branin's Function

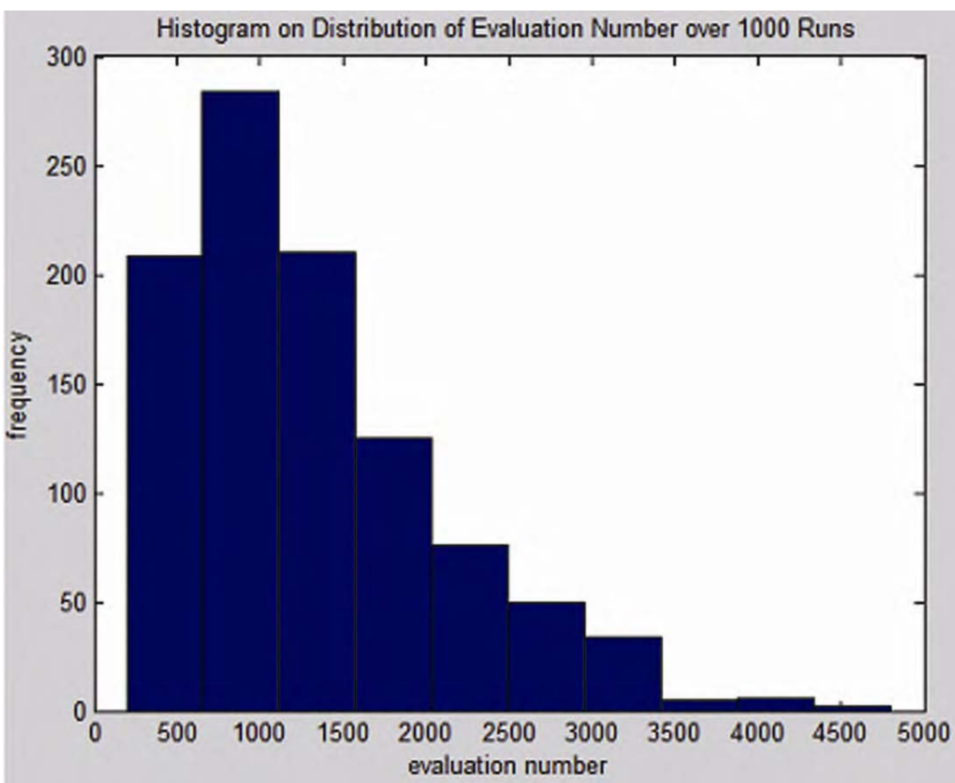

Fig. 21—Histogram on distribution for the case of 8 flies.

each having depth rating of up to a maximum of 300 m. Each vehicle measures approximately 38 cm. in diameter, is propelled with two slim line water pumps, and powered by lithium ion batteries. The GPS's accuracy is about 3 meters and can be improved up to until 1.3 meters by using GNSS infrastructure.

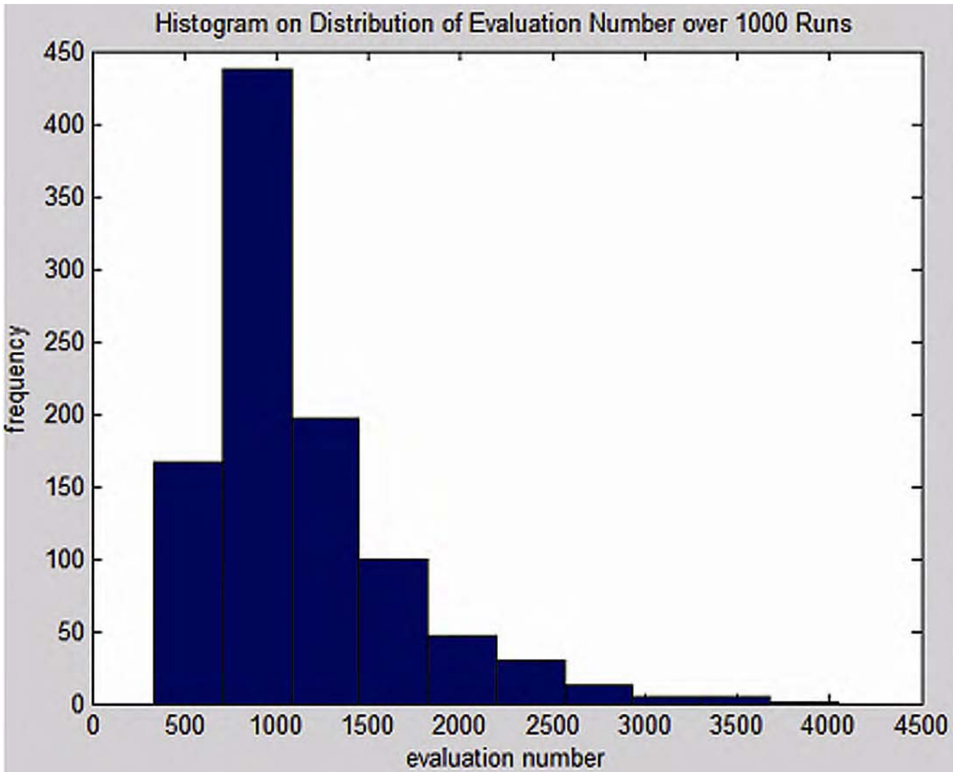

Fig. 22—Histogram on distribution for the case of 16 flies.

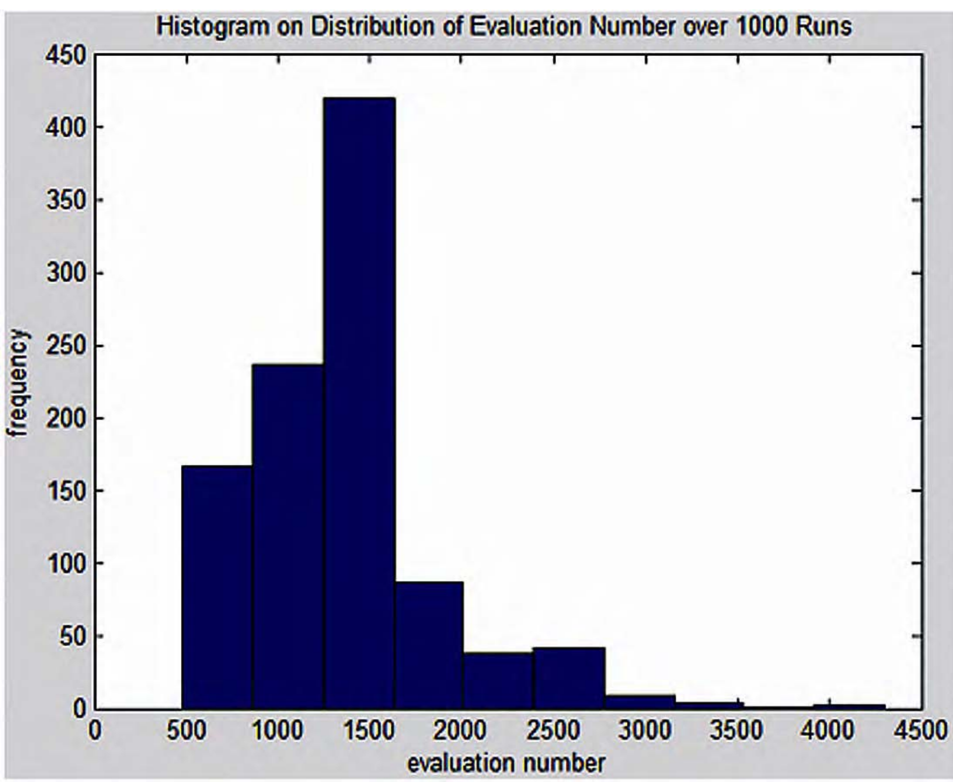

Fig. 23—Histogram on distribution for the case of 24 flies.

Fig. 24—Actual *Drosobots* ready for deployment (left). The robots in action (right).

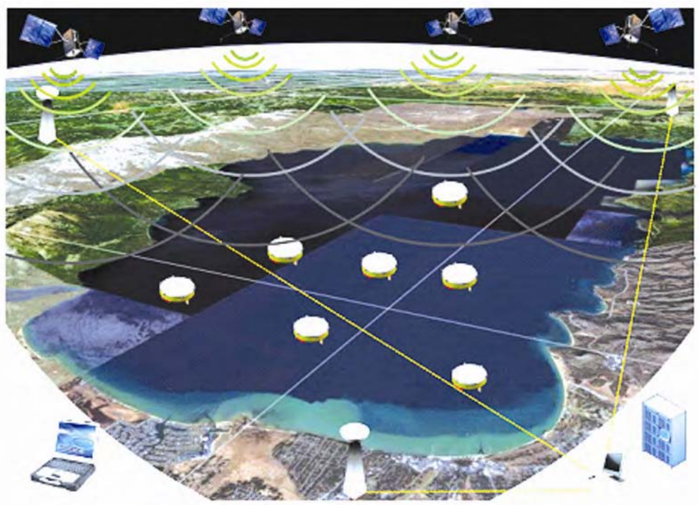

Fig. 25—Overview of the *Drosobots* Project.

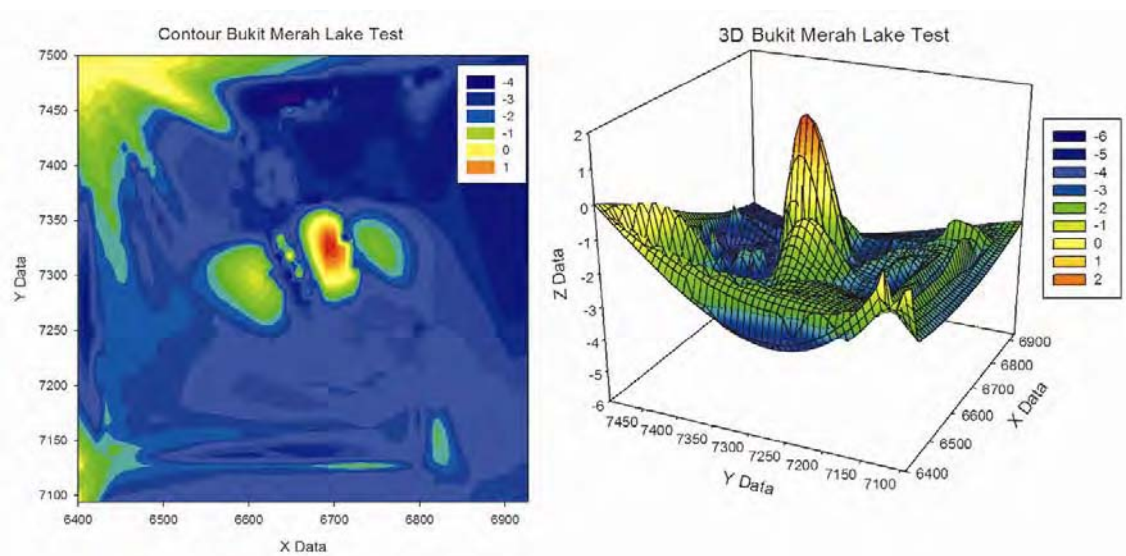

Fig. 26—2D and 3D mapping of Bukit Merah Lake.

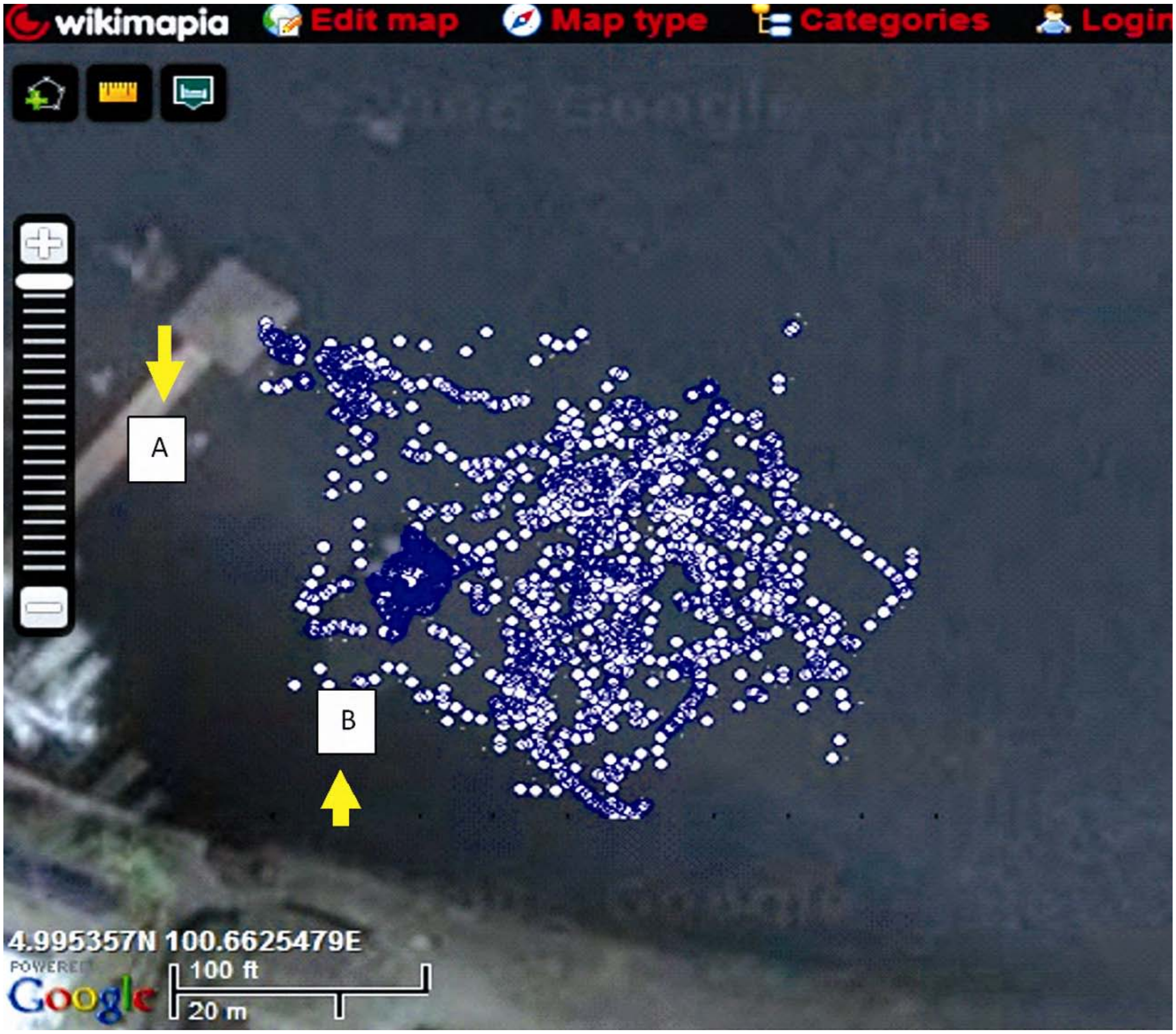

Fig. 27—Bukit Merah scattered plot trial. Deployment station (A). Obstacle (B)

We had conducted an experiment at Bukit Merah Lake (GPS coordinate: N4.9949294, E100.6603432). The scattered plots of *Drosobots* show that, the system is able to perform the navigation mission. However, certain unforeseen obstacles (such as drifting bushes) which are not detected on the GPSS system posed as a problem in that, the agents become trapped and could not proceed with the next way point (Fig. 26 & 27). In the near future, we plan to add an avoidance detection system to counter such problems and finally to integrate the FOA into the *Drosobots*.

Conclusion

Basic features of the Fly Algorithm have been illustrated at great length in the present study . Performance analysis as presented in the paper has assisted in deeper understanding of its efficiency as an optimal searching algorithm as compared to other animal inspired metaheuristic algorithms. Currently, its performance has enabled further development and improvements to this algorithm and most importantly to proceed with the specific mission: the *Drosobots* project. Development of this algorithm will further focus on multiple peak scenarios and enhancement of the shrinking method. The main idea of the algorithm which is based upon the Drosophila's biological behavior shows that it has promising potential for solving optimization problems which may be harnessed for other applications.

Acknowledgment

This research is sponsored by the National Oceanography Department, Malaysia, under grants NOD-USM 6050124 and USM Research University, under grant 1001/PELECT/814059.

References

1 Glover, F., *Future Paths for Integer Programming and Links to Artificial Intelligence.* Computers & Operations Research, 1986. 13(5): 533-549.

2 Zang, H., Zhang, Shujun,Hapeshi, Kevin, *A Review of Nature-Inspired Algorithms.* Journal of Bionic Engineering, 2010. 7(Supplement 1): , S232-S237.

3 Dorigo, M., *Optimization, Learning and Natural Algorithms*. 1992, Ph.D. thesis, Politecnico di Milano: Italy.

4 Eberhart, R.C.a.K., J. *A new optimizer using particle swarm theory*. in *Proceedings of the Sixth International Symposium on Micromachine and Human Science*. 1995. Nagoya, Japan.

5 Kammerdiner, A., Mucherino, Antonio,Pardalos, Panos, *Application of Monkey Search Meta-heuristic to Solving Instances of the Multidimensional Assignment Problem*, in *Optimization and Cooperative Control Strategies*. 2009. 385-397.

6 Nakrani, S. and C. Tovey, *On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers.* Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems, 2004. 12(3-4): 223-240.

7 Yang, X.-S., *Nature-Inspired Metaheuristic Algorithms*. 2008: Luniver Press. 128.

8 Karaboga, D. and B. Basturk, *On the performance of artificial bee colony (ABC) algorithm.* Applied Soft Computing, 2008. 8(1): . 687-697.

9 Duan, H.-b., Zhang, Xiang-yin,Wu, Jiang,Ma, Guan-jun, *Max-Min Adaptive Ant Colony Optimization Approach to Multi-UAVs Coordinated Trajectory Replanning in Dynamic and Uncertain Environments.* Journal of Bionic Engineering, 2009. 6(2): . 161-173.

10 Abidin, Z.Z., Arshad, M. R.,Ngah, U. K.,Ong Boon, Ping,. *Control of mini autonomous surface vessel*. in *OCEANS 2010 IEEE - Sydney*. 2010.

11 Zulkifli Zainal Abidin, K.I.A.R., Mohd Rizal Arshad,Umi Kalthum Ngah,. *Bukit Merah Lake Contour Mapping using Swarm of mini ASVs*. in *International Conference on Intelligent Unmanned Systems (ICIUS)*. 2010. Bali, Indonesia.

12 D.T. Pham, A.G., E. Koç, S. Otri , S. Rahim , M. Zaidi. *The Bees Algorithm - A Novel Tool for Complex Optimisation Problems*. in *Proceedings of IPROMS 2006 Conference*. 2006. Virtual Internet.

13 Zulkifli Zainal Abidin, M.R.A., Umi Kalthum Ngah,. *A Survey: Animal-Inspired Metaheuristic Algorithms*. in *Proceedings of the Electrical and Electronic Postgraduate Colloquium EEPC2009*. 2009. Jawi, Penang, Malaysia.

14 Kent, C., Azanchi, R.,Smith, B.,Formosa, A.,Levine, J., *Social Context Influences Chemical Communication in D. melanogaster Males.* Current Biology, 2008. 18(18): . 1384-1389.

15 Nemenman, I., Lewen, Geoffrey D.,Bialek, William,de Ruyter van Steveninck, Rob R., *Neural Coding of Natural Stimuli: Information at Sub-Millisecond Resolution.* PLoS Comput Biol, 2008. 4(3): . e1000025.

16 Reynolds, A.M. and M.A. Frye, *Free-Flight Odor Tracking in Drosophila Is Consistent with an Optimal Intermittent Scale-Free Search.* PLoS ONE, 2007. 2(4): 354.

17 Christenson, L.D. and R.H. Foote, *Biology of Fruit Flies.* Annual Review of Entomology, 1960. 5(1): . 171-192.

18 Etter, P.D. and M. Ramaswami, *The ups and downs of daily life: Profiling circadian gene expression in Drosophila.* BioEssays, 2002. 24(6): 494-498.

19 Elland, C. and B.J. Navarro. *Flies in Space*. 2006 September 2009]; Available from: http://quest.nasa.gov/projects/flies/index.html.

20 Bartumeus, F., da Luz, M. G. E.,Viswanathan, G. M.,Catalan, J., *Animal search strategies: a quantitative random-walk analysis* Ecology, 2005. 86(11): . 3078-3087.

21 Maye, A., Hsieh Chih-hao,Sugihara George,Brembs B, *Order in Spontaneous Behavior.* PLoS ONE, 2007. 2(5): 443.

22 Viswanathan, G.M., Buldyrev, Sergey V.,Havlin, Shlomo,da Luz, M. G. E.,Raposo, E. P.,Stanley, H. Eugene, *Optimizing the success of random searches.* Nature, 1999. 401(6756): 911-914.

23 Frye, M.A., M. Tarsitano, and M.H. Dickinson, *Odor localization requires visual feedback during free flight in Drosophila melanogaster.* J Exp Biol, 2003. 206(5): 843-855.

24 Joshua J. Krupp, C.K., Jean-Christophe Billeter, Reza Azanchi, Anthony K.-C. So, Julia A. Schonfeld, Benjamin P. Smith1, Christophe Lucas, and Joel D. Levine, *Social Experience Modifies Pheromone Expression and Mating Behavior in Male Drosophila melanogaster.* 2008. 18(18): 1373-1383

25 S. Tinette, L.Z., A. Robichon,, *Cooperation between Drosophila flies in searching behavior.* Genes, Brain & Behavior, 2004. 3(139-50.

26 Yang, X.-S., *Firefly algorithms for multimodal optimization*, in *Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications.* 2009, Springer-Verlag: Sapporo, Japan. . 169-178.

27 Abidin, Z.A., Arshad, M. R.,Ngah, U. K.,Kok Chee, Hou,. *GPS boundary navigation of DrosoBots using MATLAB simulation*. in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*. 2010.