Dynamic Allocation of Neighborhood Search in the Bees Algorithm

Hasan Al-Jabbouli, Jean-Claude Ngatchou

Computer Science Department, New Jersey City University, Jersey City, NJ, USA.

ABSTRACT - The main objective of this paper is to improve the performance of the Bees Optimization Algorithm- for the purposes of solving function optimization problems. The Bees Algorithm mimics the food foraging behavior of honey bees and uses local and global search to find optimum solutions to avoid local optimum. In this paper, the concept of dynamic neighborhood search is introduced. The resulting algorithm shows a significant improvement over the original Bees algorithm. The paper gives the results obtained for a number of benchmark problems, demonstrating the efficiency of the new improved algorithm when compared to the original algorithm.

KEYWORDS- Bees Algorithm, Function Optimization, Local Optimum, Dynamic neighbourhood Search

I. INTRODUCTION

The function optimization, which is defined as a process of finding the best solution from all possible solutions [1], is considered a complex problem, especially for multi-variable optimization where the complexity of the problem is directly proportional to the number of variables (dimensions). Optimization problems can be classified as continuous optimization where variables are continuous or combinatorial optimization where variables are discrete [2].

Giving the multi-dimensional nature of this type of problems, they are best dealt with using optimization algorithms. The Bees Algorithm, which is an optimization algorithm simulates the food foraging behavior of the honeybees, was already used for function optimization [3]. This paper shows that the algorithm performance can be improved further by using dynamic allocation of neighbourhood search. The structure of the paper is as follows. Section 2 describes the Bees Algorithm and some of its applications. Section 3 introduces the proposed solution which is a major improvement over the original Bees Algorithm. The results of applying the dynamic allocation of neighborhood search to the Bees algorithm were obtained using some functions and are addressed in Section 4. Section 5 concludes the paper.

II. THE BEES ALGORITHM

The Bees Algorithm is an optimization algorithm proposed by Pham et al [3], has become one of the most successful optimization algorithms. It simulates the food foraging behavior of the honey-bees in nature. Using a population of bees to sample the solution space, the process starts by generating scout bees to search for high fitness places randomly (global search). The second and subsequent iterations use both local and global search, the local recruits more bees to search around the highest fitness values and the global will randomly search elsewhere. Sequences of global and local search are repeated until an adequate fitness is revealed, or a given number of iterations have passed.

The basic steps of the Bees Algorithm are explained in figure 1. The Algorithm requires a number of parameters to be set, namely, number of scout bees (n), number of sites selected for neighborhood search (m), number of best “elite” sites out of m selected sites (e), number of bees recruited for the best e sites (nep), number of bees recruited for the other (m-e) selected sites (nsp) and stopping criterion.

1- Initialize population with random solutions by sending (n) scout bees.
2- Evaluate fitness of the population.
3- While (stopping criterion not met)
    a. Select (m) sites for neighborhood search.
    b. Recruit bees for selected sites (more bees (nep) for best (e) sites and (nsp) for (m-e) sites) then evaluate fitness.
    c. Select the fittest bee from each patch.
    d. Assign remaining bees to search randomly and evaluate their fitness.
4- End While

Fig. 1

The algorithm presented successful implementation on optimization problems in different disciplines such as Manufacturing [4], Control [5], clustering [6], Multi-objective optimization [7] and other various optimization problems [8]. One of the main advantages of the Bees algorithm is the ability to avoid local optima by using both the local and global search in the same time.

III. THE PROPOSED SOLUTION

With the aim of improving the performance of the algorithm, the performance cycle of the solution found by the bees Algorithm was analyzed. Figure 2 illustrates the graph of the number of evaluations versus the optimum value found by the original algorithm when applied on different functions. These functions are explained in details in Table1.

We call latency period (labeled by arrows in figure 2), the period during which the algorithm returns the same optimum value. As it can be noticed below (fig. 2), the improvement of the solution in the original Bees algorithm has a set of long latency periods (labeled by arrow in fig. 2). Each one of them takes large number of iterations which makes the algorithm take long time to have better solution (using local and global search simultaneously).

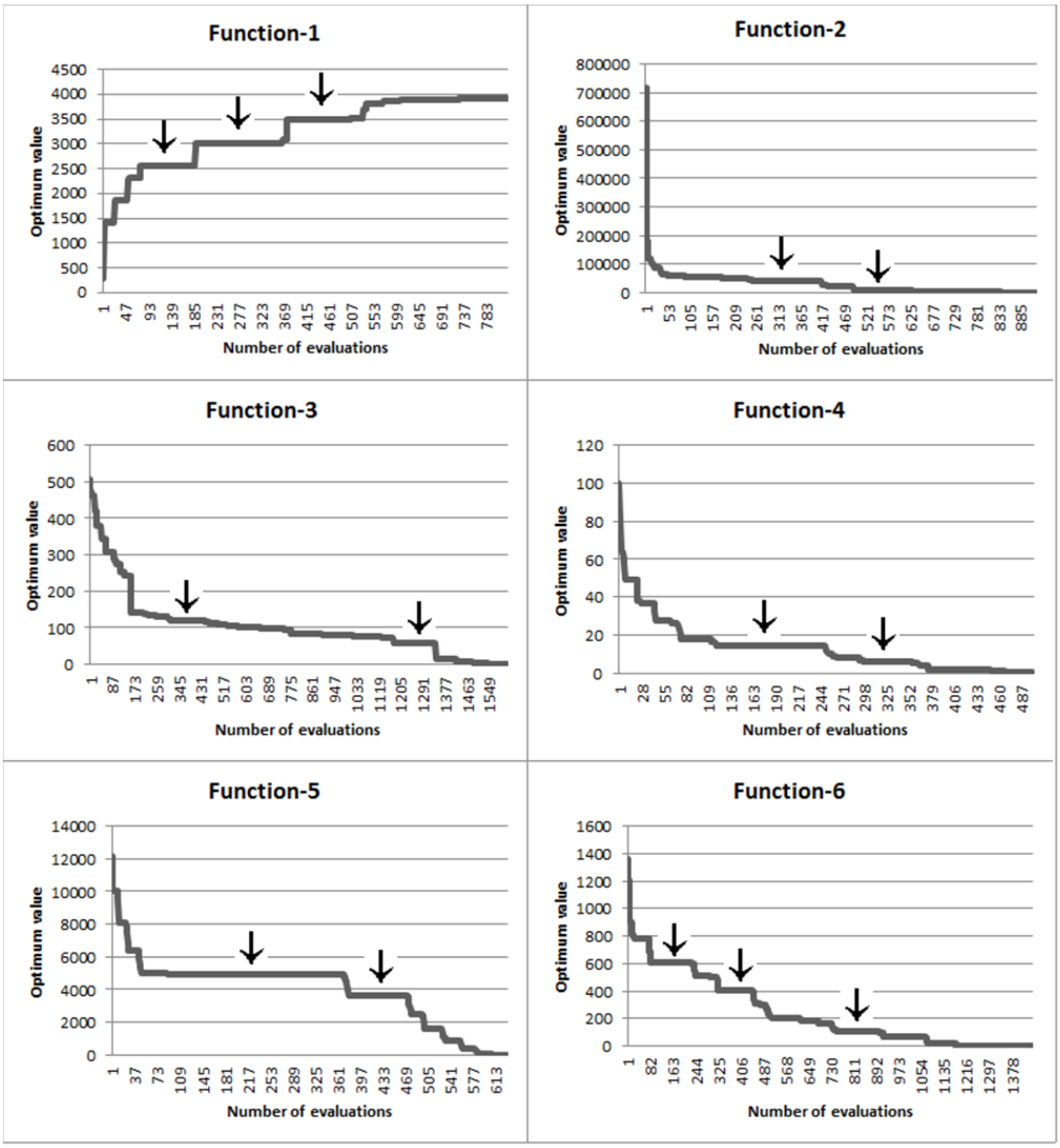

Fig. 2

In order to find a better solution, the concept of dynamic neighborhood search (ngh) is introduced. Local search are performed by reducing (shrinking) or increasing (expanding) the length of radius around each site of the (m) selected sites. This dynamic process has yielded a shorter latency period and produced the best solutions with less number of iterations. Expanding or shrinking a neighborhood is randomly performed, if there is no improvement after a specific number of iterations (i.e. dependent of the latency). The size of each neighborhood search area will be increased (expanded) or decreased (shrunk) using a pre-defined threshold.

Figure 3 shows the pseudo code of the proposed algorithm in its simplest form. The algorithm follows the same structure as the original Bees algorithm except the underlined steps. It starts with initial number of scout bees placed randomly in the search space in step 1 with fitness evaluation in step 2.

1- Initialize population with random solutions by sending (n) scout bees.
2- Evaluate fitness of the population.
3- While (stopping criterion not met)
   a. Select (m) sites for neighborhood search.
   b. Recruit bees for selected sites (more bees (nep) for best (e) sites and (nsp) for (m-e) sites) then evaluate fitness.
   c. Select the fittest bee from each patch.
   d. Assign remaining bees to search randomly and evaluate their fitness.
   e. <u>Evaluate the best fitness improvement</u>
   f. <u>If (there is no improvement) then</u>
      i. <u>Change the size of (ngh) (shrinking or expanding)</u>
   g. <u>End If</u>
4- End While

Fig. 3

The algorithm then continuously assigns bees around the best selected sites to perform the local search and the remaining bees for random search. The fitness of all recruited bees is evaluated and best bees sites are selected for next iteration. In each iteration: the best fitness found in this iteration is compared to the one found already (in step 3.e). If there is no improvement, the algorithm then modifies the size of ngh (shrinking or expanding) aiming to find better fitness around best sites. The proposed algorithm alters the size of (ngh) randomly by increasing the size (expanding) or decreasing it (shrinking) as the main goal is to get out of the no-improvement status of the algorithm.

IV. EXPERIMENTS

The proposed algorithm was applied to eight benchmark functions [9] and the results were compared to those obtained using the original Bees algorithm. The test functions and their optima are shown in Table 1.

Table 1.

Test Functions

No	Function Name	Interval	Function	Global Optimum
1	De Jong	[-2.048, 2.048]	$\max F = (3905.93) - 100\,(x_1{}^2 - x_2)^2 - (1 - x_1)^2$	X(1,1) F=3905.93
2	Goldstein & Price	[-2,2]	$\min F = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1{}^2 - 14x_2 + 6x_1x_2 + 3x_2{}^2)] * [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_2{}^2 + 48x_2 - 36x_1x_2 + 27x_2{}^2)]$	X(0,-1) F=3
3	Branin	[-5,10]	$\min F = a(x_2 - bx_1{}^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e$	X(-22/7,12.275) X(22/7,2.275) X(66/7,2.475) F=0.3977272
4	Martin & Gaddy	[0,10]	$\min F = (x_1 - x_2)^2 + ((x_1 + x_2 - 10)/3)^2$	X(5,5) F=0
5	Rosenbrock	(a)[-1.2,1.2] (b)[-10,10]	$\min F = 100\,(x_1{}^2 - x_2)^2 + (1 - x_1)^2$	X(1,1) F=0
6	Rosenbrock	[-1.2,1.2]	$\min f = \sum_{i=1}^{3}(100(x_i{}^2 - x_{i+1})^2 + (1 - x_i)^2)$	X(1,1,1,1) F=0

Table 2 presents the results obtained by the proposed algorithm and those by the original Bees Algorithm.

Table 2.

Results of running each algorithm

Function	Original Bees Algorithm		The Proposed Algorithm	
	Success %	Mean no. of evaluations	Success %	Mean no. of evaluations
1	100	825	100	239
2	100	922	100	126
3	100	1619	100	985

4	100	502	100	221
5	100	631	100	115
6	100	1452	100	113

The results shown are averages of 10 independent runs. Table 3 shows the empirically derived Bees Algorithm parameter values used with the different test functions.

Table 3.

Parameters values used for the test

Population	n=25	**Number of selected sites**	m=3
Initial patch size	ngh=3	**Number of elite sites**	e=1
Number bees around elite sites	nep=7	**number of bees around other elected sites**	nsp=2

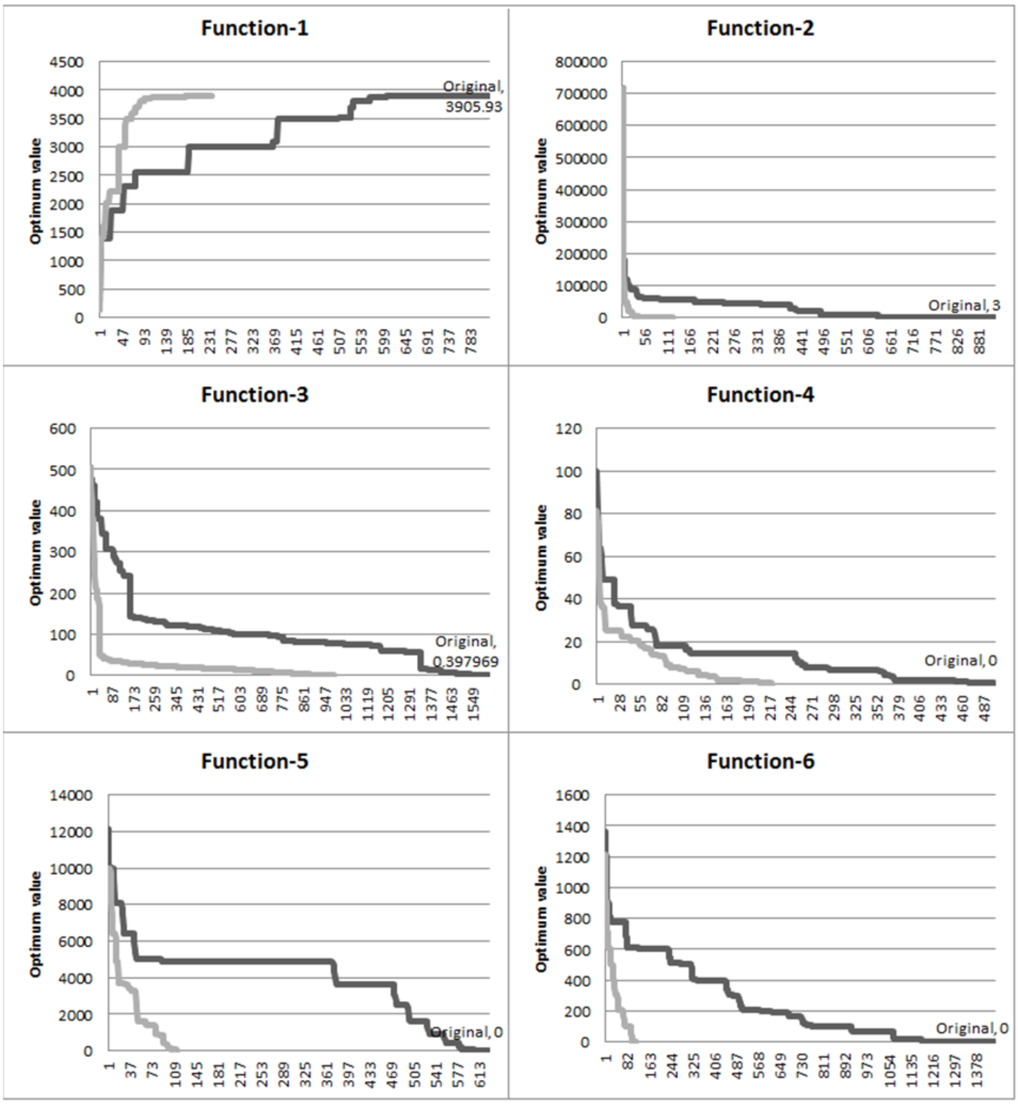

Fig. 4 Comparison between the fitness values obtained by different algorithms

Figure 4, shows a comparison between the fitness values obtained by the original Bees Algorithm and those of the proposed solution when applied to different functions. It can be seen clearly that in the proposed solutions the latency periods was shortened by using the shrinking and expanding techniques.

V. Conclusion

This paper has introduced a dynamic allocation of neighborhood search method to improve the performance of the Bees Algorithm. Experimental results on multi-modal functions in n-dimensions show that the proposed algorithm has remarkable results, producing a 100% success rate in all cases. The proposed algorithm generally outperformed other Original Bees Algorithm in terms of optimization speed.

VI. REFERENCES

[1] C. T. Reviews, e-Study Guide for: Practical Optimization. Cram101, 2012, p. 41.

[2] E. Pappalardo, P. Pardalos, and G. Stracquadanio, Optimization Approaches for Solving String Selection Problems. Springer Science & Business Media, 2013, p. 57.

[3] D. Pham, A. Ghanbarzadeh, and E. Koc, “The Bees Algorithm–A Novel Tool for Complex Optimisation,” … Syst. I* …, 2011.

[4] D. T. Pham, A. J. Soroka, E. Koc, A. Ghanbarzadeh, and S. Otri, “Some applications of the bees algorithm in engineering design and manufacture.” ICMA, 02-Oct-2007.

[5] A. A. Fahmy, M. Kalyoncu, and M. Castellani, “Automatic design of control systems for robot manipulators using the bees algorithm,” Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng., vol. 226, no. 4, pp. 497–508, Nov. 2011.

[6] D. Pham, S. Otri, and A. Afify, “Data clustering using the bees algorithm,” Proc. 40th …, 2007.

[7] E. Mastrocinque, B. Yuce, A. Lambiase, and M. S., “A Multi-Objective Optimisation for Supply Chain Network Using the Bees Algorithm,” Int. J. Eng. Bus. Manag., p. 1, Aug. 2013.

[8] A. M. Nafchi, A. Moradi, A. Ghanbarzadeh, A. Rezazadeh, and E. Soodmand, “Solving engineering optimization problems using the Bees Algorithm,” in 2011 IEEE Colloquium on Humanities, Science and Engineering, 2011, pp. 162–166.

[9] M. Mathur, S. B. Karale, S. Priye, V. K. Jayaraman, and B. D. Kulkarni, “Ant Colony Approach to Continuous Function Optimization,” Ind. Eng. Chem. Res., vol. 39, no. 10, pp. 3814–3822, Oct. 2000.