Volume 6, Issue 12, December 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Green Optimisation: A Nature Inspired Evolutionary Algorithm

K. S. Madhavan*
Sr. DGM/PCS, Corp. R&D, BHEL,
Hyderabad, India

Harsh Vardhan Tripathi
Sr. Engineer, Corp. Digital Transformation, BHEL
Hyderabad, India

Abstract— Green computing involves the usage of computer and related resources in an environmentally responsible manner. Though in the larger view point, it would mean implementation of energy efficient CPUs, servers and peripherals, proper disposal of electronic waste and reduced resource consumption. The idea of Green optimisation stems from this basic definition in using energy efficient and optimal algorithms that would eventually result in reduced resource consumption, optimal and efficient use of CPU time. In this paper a novel algorithm for Artificial Bee Colony optimisation is evolved and a comparative study of Artificial Bee Colony optimisation and Ant Colony Optimisation are done for an optimal power flow problem in terms of CPU utilization.

Keywords— Green optimisation, Stigmergy, Artificial Bee Colony Optimisation, Employed bees, Onlooker bees, Roulette Wheel Selection, Newton Raphson

I. INTRODUCTION

At present, in a typical computing environment, CPU consumes the major chunk of power as compared to the other components such as memory, power supply efficiency losses, PCI slot(s), motherboard etc. However, it is predicted that with the rapid surge in memory intensive applications, memory will consume more power than CPU in the recent future. Therefore, in order to achieve objectives of green optimisation it is important to optimize computational power with optimal usage of both time and space.

In general, CPU power consumption and percentage of CPU utilization have a linear relationship. By bringing down processing time we not only achieve reduced CPU power consumption but also achieve considerable reduction in CPU heat generated. This leads to huge reduction in the energy and cost required in maintaining proper air conditioning especially in case of data centres. Thus, implementation of these non-linear optimisation techniques in solving real world problems would lead to sustainable development. The cost optimisation of processes is an imminent factor.

We are presently aware of a plethora of optimization techniques and search algorithms in the field of evolutionary computation. The entire gamut of techniques range from gradient based genetic algorithms to harmony search algorithms to Bat algorithms. Particle Swarm Optimization techniques include Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) optimization etc. Inspired by the flocking and schooling patterns of birds and fish, the discovery of these series of optimization techniques is attributed to James Kennedy, Russell C. Eberhart and Yuhui Shi [1]. These are meta-heuristic algorithms. They are not based on calculation of gradient of the objective function, hence the function under consideration need to be approximately differentiable. It can thus be based on real-world problem domain. The data on which they are processed may be noisy, irregular and discontinuous.

The ABC algorithm is used in optimal power flow, control, multi-objective optimisation, manufacturing, bio-engineering, clustering systems, routing and wavelength assignment in all optical networks, structural optimisation and other optimisation problems. It has been used for MR brain image classification, face pose estimation, multi-level thresholding, nanoelectronic based phase locked loop optimisation, and to speed up physical design optimisation in real world problem domain.

ACO is based on pheromones secreted by ants that evolve optimal search paths to their food source and back. Ants use stigmergy to create physical order and organisation from which we may do study on traffic flows, organisational development etc. These are not fully Green studies in the true sense of the word, but they would definitely go a long way in reducing fuel costs by optimising traffic flows and developing eco-friendly organisations.

In the real world bee colony optimisation is greener than ant colony optimisation. In this paper we are simulating and comparing the performances of ABC and ACO in terms of computational time.

II. ARTIFICIAL BEE COLONY OPTIMISATION

Bee colony optimisation is a nature inspired algorithm. It mimics the food foraging behaviour of bees. It is a new swarm algorithm proposed by Karaboga in 2005 [2]. The bees have a range of about 14 km for foraging of food. They scout for flower patches that are useful for pollen and hives. The employed bees spread the information about the food source to the onlooker bees through their memory. The onlooker bees find out the best quality of the food source. These useful flower patches are called dance floors where the employed bees and the onlooker bees are attracted and they perform what is called a waggle dance. Once the bees finish exploitation of a flower source, they move to the next dance floor with the help of scout bees. This way rich and profitable food sources are exploited. But there is a symbiosis in the bee colony optimisation because of the pollination effect and that gives us a rich harvest of fruits and vegetables. Artificial bee colony optimisation is based on the above principle. It replicates the behaviour of bees found in nature. It has a forward pass and a backward pass. The forward pass involves the search for the food source. In the backward pass the artificial bees share information on the quality of solution. Though Artificial Bee Colony Optimisation is a parallel algorithm, it can be substituted as the best sequential algorithm for speed up and efficiency implementation issues [3].

The algorithm parameters that need to be set apriori are:
B – Number of bees in the hive
NC – Number of constructive moves during one forward pass
S(x) – Solution for x
The pseudo code for the Artificial Bee Colony Optimisation is as follows:
1. Initialization: n = 1, S(x) = null;
2. For every bee: // the forward pass
i. Set k = 1; //counter for NC;
ii. Evaluate ∀ NC;
iii. According to evaluation, choose one move using roulette wheel selection;
iv. k = k + 1; If k ≤ NC Go to step ii.
3. All bees are back to the hive; // backward pass starts;
4. Evaluate (partial) objective function for each bee;
5. Every bee randomly decides whether to continue its own search for food source and become an employed bee, or an onlooker bee;
6. For every onlooker bee, choose S(x) from employed bees by roulette wheel selection;
7. If ∃ x: S(x) = null; go to step 2;
8. Find best S(x).
9. n = n + 1; If n ≤ B Go to step 2;
10. Output best S(x).

The power flow using Newton Raphson method can be solved based on the position of the employed bees to generate optimal power flow for the grid system under consideration [4]. The grid system under consideration is a 39 bus system supplying captive power to a refinery. The objective is to minimise reactive power flow.

III. RESULTS AND DISCUSSIONS

The equivalent circuit was taken to conduct an optimal power flow study. This exercise was carried out and the comparative results for Ant Colony Optimisation (ACO) and Artificial Bee Colony optimisation (ABC) are enclosed for the CPU times used. The adjusted CPU time for ACO is calculated taking number of Flops of ABC as base. The formula used is:

Adjusted CPU (s) = (No. of Flops / No. of Flops of ABC) x CPU time of ACO

Method	Cores	Clock (GHz)	GFlops	CPU (s)	Adjusted CPU (s)
ACO	2	1.85	14.80	60.92	68.51
ABC	2	2.0	13.16	29.23	29.23

In fact it would be ambiguous at this stage to state that ABC is better than ACO in terms of CPU time used. Literature survey shows that there are others who state that the reverse is true. It is advisable to probe more into the subject before an open declaration can be made. In the above experiment it has been shown that ABC is better than ACO in terms of CPU time used.

Compact artificial bee colony optimisation algorithms reduce memory requirements by simulating the behaviour of ABC algorithm by employing its probabilistic representation instead of an actual population of solutions [5].

IV. CONCLUSIONS

The bee algorithm performs a faster search than many known algorithms. But the convergence on the solution is found to be slow. Though onlooker bees identify quality of food source, exploitation of food source is found to be poor. The algorithm easily gets stuck while handling complex multi-modal problems. In the above pseudo code, we have used a roulette wheel selection. This has to be substituted by a differential evolution technique to get better results. In which, differential evolution is used to generate new solutions and then bee algorithm is utilized to update the solutions based on their levels of fitness. Advanced research is going on to further improve the results from Artificial Bee Colony optimisation. These are largely called Interactive Artificial Bee Colony Optimisation techniques. One of them introduces the concept of Newton's laws of universal gravitation into the consideration of the affection between employed bees and the onlooker bees.

ACKNOWLEDGMENT

The authors are thankful to BHEL Management for according permission to publish this work.

REFERENCES

[1] James Kennedy, Russell C. Eberhart and Yuhui Shi, Swarm Intelligence, 1st edition, Elsevier (2001).

[2] Dervis Karaboga, Bahriye Akay, A comparative study of Artificial Bee Colony algorithm, Applied Mathematics and Computation 214(209)108-132.

[3] Teodorović D., Davidović T., Šelmić M., Bee Colony Optimization Overview, (2010).

[4] Luong Le Dinh, Dieu Vo Ngoc, Pandian Vasant, Artificial Bee Colony Algorithm for Solving Optimal Power Flow Problem, The Scientific World Journal (2013).

[5] Thi-Kien Dao, Shu-Chuan Chu, Trong-The Nguyen, Chin-Shiuh Shieh, Mong-Fong Horng, Compact Artificial Bee Colony, Modern Advances in Applied Intelligence, Springer (2014).

