



A Survey on Metaheuristic Approaches and Its Evaluation for Load Balancing in Cloud Computing

Deepak Garg^(✉) and Pardeep Kumar

Department of Computer Science and Application, Kurukshetra University,
Kurukshetra, Haryana, India

erdeepakgarg21@gmail.com, pmittal@kuk.ac.in

Abstract. In daily life there exist many problems whose objective are to either maximize or minimize some value with following some constraints (like load balancing in cloud with aim to maximizing QoS, Travelling salesman problem with aim to minimize total length of trip). These types of problems are optimization problems. Out of these problems there exist many problems which comes under NP-Hard category. To get nearby optimal solution of these problems in polynomial time the metaheuristics approaches are used. Metaheuristics are nature inspired algorithms which provides optimal solution by utilizing combination of exploration and exploitation. This paper provides a survey of Metaheuristic approaches (consisting of need, applications, characteristics, general classification and fourteen approaches under it). Compared all approaches corresponding to key parameters, mechanism. On the basis of literature survey and comparison, cuckoo search has been considered better due to global search via levy flight and generality (because of single parameter setting in cuckoo search). Implemented Randomized algorithm, Genetic Algorithm and Cuckoo Search to solve Load Balancing problem in Cloud Computing with aim to minimize makespan time and proved through results that cuckoo search is better. These experimental results were obtained using CloudSim 3.0.3 toolkit by extending few base classes.

Keywords: Cloud computing · Cuckoo search · Genetic algorithms
Load balancing · Metaheuristic algorithms

1 Introduction

Meta- It means beyond or higher level.

Heuristics- It means to search solution by trial and error.

The word “metaheuristic” was coined by Fred Glover et al. in his seminal paper in 1986 [1], and a metaheuristic can be assumed as a “master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality”.

Furthermore, Glover call all the modern nature inspired algorithms as metaheuristics [2].

1.1 Need of Meta-Heuristic

Heuristics: It is a procedure that provides a possible solution for a problem, which is approx. optimal or can prove that no solution exists. But it is not generally applicable.

- It's design is usually problem centred.
- Easily gets stuck at local optima.

Metaheuristics: These are more general as compared to heuristic alone, these combine general structure & strategy guidelines to develop a specific heuristic approach to fit a particular problem. Below are reasons as need of metaheuristic:

- Meta-heuristic fits to many problems.
- Applicable on Multimodal Optimization.
- Applicable on Discontinuous, non-linear functions etc.

1.2 Components of Meta-Heuristic

Intensification or Exploitation: It is identifying the best fitted solution within current local region knowing that a better solution was found in the region. It helps in convergence.

Diversification or Exploration: It is identifying the better solution globally via randomization. It prevents to get stuck at local optima and at same time, it increases the diversity within population of solutions.

In a good metaheuristic algorithm a good mixture of these 2 components is required to get global solution.

1.3 Characteristics of Metaheuristic

Following are the important characteristics that almost all meta-heuristics approaches share.

- Inspired from nature. (Based on physics, biology or ethology etc. principles).
- Use stochastic components. (use of random variables).
- No restricted use of Hessian matrix or gradient.
- Generally have several parameters that need to be configured corresponding to problem.

1.4 Applications of Metaheuristic Optimization

1. NP hard Optimization Problems solutions in polynomial time.
 - Flow Shop Scheduling Problem
 - Load Balancing/Task Scheduling in Cloud Computing
 - Travelling Salesman Problem
 - P-Median Problem.

2. Complicated Search Problems in Many Applications.

- Automatic Test Data Generation
- Feature Selection in Pattern Recognition
- Automatic Clustering.

2 General Classifications

2.1 Gradient/Derivative Based and Non-gradient Based

Gradient Based: The key trait in this class is focusing on derivative or gradient of a function under consideration. For example to optimize function $f(x)$, Evaluation of first derivative ($f'(x) = 0$) is done to find the possible potential locations for that function & using the second derivative at these potential locations ($f''(x)$) to ensure that if the solution maximizes or minimizes.

Derivative-Free Algorithms: These class of algorithm do not use derivative function but these directly uses values of function itself like $f(x)$.

Generally, function in this category have discontinuities. That's why derivatives cannot be used. Eg: Nelder-Mead downhill simplex become very useful.

2.2 Single Agent/Trajectory Based and Population Based

Trajectory-based algorithm: These type of algorithms uses a single trajectory or agent or individual to find solution at a time. Or we can say in these algorithms the population size of individuals is one only.

Eg: Hill climbing comes under this category, it starts from a single seed position & links that with the final maximum or minimum point via searching.

Population - Based Algorithm: On the other hand, It uses more than one agents at a same time which will interact and trace out multiple paths as the iterations goes on.

Eg: Particle Swarm Optimization Can be Parallelized.

2.3 Deterministic and Stochastic

Deterministic: In this category of algorithm, solution is found by only using fixed deterministic way without taking help of any randomness. The main feature of algorithm under this is that these will reach the same final solution position if started with the same initial point.

Eg: Hill-climbing is good example of deterministic algorithms.

Stochastic: These category algorithm takes the help of randomness in addition with fixed approach.

Thus on starting with same initial point the final solution may be different always.

2.4 Local and Global Search

Local search algorithms: Algorithm which mostly get premature converged to local optima falls under this category.

Furthermore, these can't escapes from local optima generally.

Eg: Hill-climbing an example of this.

Global Search: These tend to be suitable for global exploration of search space & optimization, though not always successful or efficient.

– The key feature in this is role of randomness.

Eg: Hill Climbing suffers from local optima but random starting point, random jumps can turn this into global search.

3 Literature Survey

In 1975, Holland proposed a nature inspired algorithm named as Genetic algorithm [3] It is motivated from the "evolution in nature" which is the theory of Darwin. It replicates the method of natural selection for survival. The Performance of Genetic Algorithm depends upon four factors which are population size, mutation probability, crossover probability & number of generations or iterations. In each iteration a sequential process of selection, crossover, mutation, replacement of individuals is done. To determine survival chances of each individual within population a fitness function is determined corresponding to phenotype structure of individual. After selecting good individuals corresponding to fitness function they are crossover to form new offspring. Crossover is performed on genotype structure. It is kind of exploitation of good features within parents to form better offspring. Mutation is like exploration by random changes. It is also like global search. In replacement out of old population and generated population new offspring population has been selected. This sequence of steps are iterated till fulfillment of maximum iterations or till some convergence condition.

In 1983, Kirpatrick et al. proposed Simulated Annealing. It is type of random search & mimics the method of slowly cooling of molten metal to get the minimum energy function value in this minimization problem [4]. Initially temperature starts from high, which shows that the selection pressure is low. Then the temperature is gradually decreased, which means that raise in selection pressure. In the minimization problem, any better moves or changes in individual solution that lowered the value of objective function is permitted; however, the main feature is that it also accept not better steps with some probability.

In 1986, Glover et al. proposed Tabu search. It is inspired from the mechanisms of human memory [2]. In this a list or memory has been created named as tabu list, which starts to store searched better distinct moves and does not permit to coming back at already searched move. Thus it prevents endless repeated cycling of same solution search and within this process it also allow to accept not better move as compared to repeated move. The key feature of this search is length of tabu list. Short length allows local search while long length allows global search.

In 1992, Dorigo [5] proposed algorithm based on the ants food searching behavior and named it as Ant colony optimization. In the search of food Ants intelligently discover the shortest path between ant nest and the food source. Initially ants moves randomly in search of food leaving pheromone in their path and when they find food they came back leaving pheromones again. Other ants follows the path with strongest pheromones scent. While some ants continuously search for other good food source or search for other better path nearby current good path. Each ant is considered as a potential solution to the objective function. Pheromones concentration in each path is considered as quality of solution. Within this whole process rate of pheromone evaporation and pheromone deposition plays role. In computer science problems, ACO is considered as iteration cycle of construction of ant solutions, updation of pheromones, daemons actions and this cycle is repeated till satisfaction of convergence criteria.

In 1992, Moscato et al. suggested a memetic method for the TSP (traveling salesman problem) [6]. Memetic means study of memes and their social and cultural effects. It is also motivated by Dawkin's theory which is evolution in nature. As compared to genetic algorithm in memetic algorithm, a set memes are assumed to form the chromosomes rather than genes. In the GA, after the selection of the individuals the crossover and mutation methods starts immediately but in Memetic Algorithm, an individual solution takes time to acquire experience memes & then these crossover or mutation like operations are performed. In Memetic algorithm local search may include after every step (like after crossover, mutation). For example after crossover two off-spring are produced then they are allowed to search locally by using hill climbing for better solutions.

In 1995, Kennedy et al. suggested Particle Swarm Optimization [7]. Particle Swarm Optimization is centered on the flocking behavior (together flying of large no. of birds for migration) of birds. The birds fly in a possible solution space and the flocking behavior finds the optimum position or velocity as solution. Birds follow some path randomly or depending upon personal best and global best to reach to their food destination. Particular or personal best position or solution is the shortest path followed by that bird. Global best solution is best shortest path originate so far by any bird or particle. Then Bird tends to flew or move towards its personal or local best position solution. Birds also keep track of the gbest (global best) solution. Every particle or bird is linked with a velocity parameter, by which bird moved towards the its local best path & the global best path, keeping track of the position in 'n' dimension & position with respect to global best and localbest. Every Birds interact with each other to follow this strategy and regularly updates important parameters after a certain time. This whole process is iterated till satisfaction of convergence criteria.

In 2002, Passino et al. developed Bacterial foraging optimization Algorithm [8]. It involves 3 key steps: chemotactic, reproduction and elimination-dispersal step. In first step i.e. chemotactic, bacteria moves or swims towards the direction of rich nutrient area while it get tumbled when a harmful or noxious area has been contacted. The main objective of this algorithm is to minimize the cost movement of bacteria towards rich nutrient area. After this first step, all the bacteria are sorted in decreasing order of movement cost or fitness value. In second step i.e. reproduction step, the first half of bacteria having high cost get died because they did not got necessary nutrient to survive

and the every remaining (half population) bacteria got split into two bacteria thus keeping population size fixed. In last step i.e. elimination- dispersal step, bacteria scattered or dispersed in complete surface for searching good nutrient surface. Newly reproduced bacteria takes position of died or eliminated bacteria. Bacteria with least minimum cost or best fitness value, represents the solution. This sequence of steps is repeated till fulfillment of coverage criteria.

In 2003, Eusuff et al. proposed Shuffled Frog Leaping Algorithm (SFLA) [9]. It mixes the best features of 2 algorithms: MA & PSO. It is motivated from leaping & shuffling nature of frogs to share information related to food search. Every frog is considered as a possible solution of the problem. Fitness value of each frog is evaluated and then they are sorted into decreasing order of their fitness. After this all frogs are divided in number of groups named as memeplexes. In every memeplex local best solution is evaluated and local evolution is done within every memeplex. After a fixed no. of memetic evolutions, for global evolution all the frogs are shuffled together. This whole process is iterated unless stopping criteria is satisfied.

In 2005, Karaboga et al. developed Artificial Bee Colony Algorithm (ABC) [10, 11]. The algorithm mimics the honey bees for food searching behavior. Authors termed bee as artificial bee because of difference in behavior with actual bee as compared to assumed one. There are three types of bees, Scout bee, employed bees and onlooker bees. Scouts randomly searches for food source. Employed bees exploits searched food positions by scouts and communicate to onlooker bees the nectar quantity of food found at particular position. Onlooker bees keeps and updates the best food position source (having maximum nectar quantity) and sends the employee bees to neighborhood of best food source to find much better food position. Thus output is food source with highest nectar quantity.

In 2007, Xin-She Yang et al. developed Firefly Algorithm (FFA) [12]. The algorithm is motivated by mimicking the flashing pattern or behavior of the fireflies for attracting other fireflies for mating purpose and to attract prey for food [13]. Fireflies are unisex thus every firefly gets attracted towards other firefly depended upon brightness of each other. Brightness increases or decreases corresponding to distance between the flies. Fireflies having lower light intensity flies or move to the high light intensity firefly, thus with decrease in distance updating is done towards its own brightness. When there is no brighter firefly then that firefly will move randomly. Desired output is the firefly with high brightness & least distance.

In 2008, Dan Simon et al. proposed BBO (Biogeography Based Optimization) [14]. It is inspired from the migrating behavior of the species in the habitat. Here, every habitat assumed as a possible solution of the problem. In this the key attribute is HSI (Habitat suitability index) corresponding to every habitat. It represents the desirability of living in habitat. There are two other parameters which depends upon this one is habitat immigration (no. of species arriving in that habitat) & other is emigration (no. of species leaving from that habitat). High HSI habitat is considered as good solution and supposed to have suitable environment for reproduction and feeding, thus high HSI habitat contains large no. of species than the low HSI habitat. Emigration rate is also higher as compared to low HIS habitat. On Migration, the migrating species passes the characteristics of the high Habitat suitability index habitat to the low Habitat suitability index habitat. Every habitat suitability is indicated as suitability index variable (SIV).

During reproduction within a habitat mutation causes unexpected species in the habitat which reasons disturbance in the equilibrium state. Equilibrium state of habitat is when emigration rate & immigration rate are same or equal. This change reasons the change in the value of SIV. This iterative process lasts up to required no. of iterations or unless certain convergence criteria occurred.

In 2009, Xin-She Yang et al. developed Cuckoo Search Algorithm (CSA) [15, 16] after inspiring from the breeding behavior of the cuckoo bird. Rather than creating their own nest cuckoo bird lay its egg in some others bird nest for reproduction and drop down the host bird's egg. When host bird arrives, then the host bird either drop the cuckoo egg or leave the whole nest. Few female cuckoo mimics their eggs in terms of shape, size, weight, color corresponding to host bird egg, sometimes even before laying of host bird egg. This causes increase in probability of cuckoo chick survival. In real life problems, every egg in nests assumed as one possible solution & cuckoo bird's egg assumed as a new solution prepared using levy flight. Levy flight is like combination of global random walk and local random walk. In this there is only one key parameter that is discovering probability of alien egg by host bird "Pa". If host bird identifies cuckoo egg as not his own egg then host bird abandon nest and builds new nest using levy flight [17] The whole process is continued upto specified iterations or unless convergence criteria is satisfied.

In 2010, Xin-She Yang et al. proposed Bat Algorithm [18]. It is inspired from the echolocation behavior of the microbats as the microbats can produce very high echolocation to find its prey or which echoes back with a frequency. Method of detecting the position of object by originating sound and getting back reflected sound is termed as echolocation. By identifying the reflected or bounced sound frequency, bats are also able to differentiate between the obstacle & prey and also can make idea about distance between them and to their nearby surroundings. Initially bats fly randomly with any velocity, sound (loudness or frequency) in search for food. Overall aim of bat is to find prey at the minimum distance. The zooming around the particular possible position and frequency parameters keeps the balance between exploitation and exploration. This whole process is continued till satisfaction of some convergence criteria.

In 2012, Xin-She Yang proposed FPA (Flower Pollination Algorithm) [19]. It is motivated from the fertilization (pollination) method of flowers. In this method, there are two types of pollination one is abiotic or self-pollination and other is biotic or cross pollination. First one is assumed as local while other one is considered as global pollination. Corresponding to optimization problem Yang [20] supposed that every plant has only one flower & every flower has only one pollen grain. Pollinators such as flies, insects or wind plays the role of pollination. There exist flower constancy which is assumed as the reproduction probability and which is proportional to similarity of the two flowers involved. That's why every flower (or pollen) is assumed as a potential solution. Objective function searches the best flower, who is capable of maximum pollination.

In 2016, Yang included [21] all the existing algorithms optimization algorithms, application areas, previous work that has been done. This paper has provided many models for computational problem solutions, consisting of optimizations centered on swarm intelligence displayed by fireflies, ants, bats and many more. Theoretically compared all important algorithm, their needs, challenges and applications in real life problems.

4 General Comparison of Cuckoo Search with Others

After deeply studying all above papers, it has been clear that metaheuristic optimization algorithm are good approaches to solve NP hard problems and for complicated searching problems. Xin-She Yang et al. has proposed many algorithms like Flower Pollination Algorithm, Cuckoo Search Algorithm, Firefly Algorithm, and Compared theoretically the traditional metaheuristic optimization algorithms with proposed algorithms. Furthermore, many studies & applications have showed that CS is better in terms of generality and global solution [17, 21, 23–25]. Overall in support of Cuckoo search some points has been given in Table 1.

Table 1. CS general difference with other algorithms

#	Parameter	Description in support of CS
1	Use of Levy flight/global search	Levy flights is considered to have infinite mean & variance. Thus, Cuckoo search can explore problem search space efficiently than the other approaches
2	No. of parameters	Only one that is Pa which is too less as compared to other algorithms
3	MetaPopulation Algo.	Every nest can have more than one host eggs represented as solutions
4	Multimodal & multiobjective	Can show all optima at once so good for multimodal & multi objective
5	Generally applicability	As no. of parameter is one, so can be applied on large no. of problems
6	Showing all optima	If No. of nest > no. of local optima can show All optima Simultaneously
7	Efficient-randomization	In CS step length is fully tailed thus effective randomization

A main advantage of cuckoo search is that the use of levy flight as global search as compared to simple random walks [17, 22]. In Levy Flight the mean and variance is supposed to be infinite. So, CS explore the search space more effectively than any other optimization algorithms.

Theoretical studies has also proved that cuckoo search fulfills the global search needs & that's why it shows and ensures global convergence features [22]. Cuckoo search has two search capabilities: one is local search and other is global search, which are balanced or controlled by a single parameter named as alien egg discovery probability Ps.

5 Cuckoo Search

It is motivated from the mimicking behavior of cuckoo bird species. Cuckoo bird does not builds its own nest but it lays egg in nest of host bird like warbler. Host bird accept those mimicked cuckoo eggs corresponding to some discovering probability. Host bird raises the cuckoo eggs till they hatched. Cuckoo do this by mimicking the color, texture & size of the host bird nest [15, 16].

To simply describe standard Cuckoo Search, following 3 rules has been used:

- Cuckoo lays a single egg at a time and drop it in a host nest (which is chosen randomly).
- Evolutionary aim is that the best nests having high-quality eggs will be forwarded to the next generations.
- The no. of host nests are fixed. When host bird arrives then it can identify cuckoo egg with a probability $p_a \in (0, 1)$. After identification host bird left the egg or abandon the nest & built a new nest at random location.

In addition to the algorithm steps, CS involves three important equations, two for the renewal (Eqs. 1 and 2) and last for reestablishment (Eq. 3) of the nests. Equations 1 & 2 can be considered as local random walk while Eq. 3 is considered as global random walk. These equations are given as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\beta) \tag{1}$$

$$\alpha = \alpha_0 (x_i^t - x_{best}^t) \tag{2}$$

$$x_i^{t+1} = x_i^t + \varepsilon \cdot (x_j^t - x_k^t) \tag{3}$$

For $i = 1, 2, \dots, N$. in this N represents number of the nests; x_i, x_j and x_k are the location of host nests within possible solution domain ($i \neq j \neq k$) and x_{best} is best solution in current generation; α represents step size (greater than zero) and α_0 is constant (value = 0.01); $Levy(\beta)$ is random vector step length. It is generated by using Eq. 4 & $\varepsilon \sim U(0,1)$ is zoom factor.

α_0 is taken as 0.01 by xin she yang considering that if it taken as too large in local random walk then it may happen that the new solutions jumps out of the specified domain which further result in wastage of evaluations.

5.1 Lévy Random or Levy(β)

Yang and Deb [15, 16] have used Mantegna’s algorithm as levy random generation method shown in below 4–6 equations.

$$s = u/|v|^{1/\beta} \tag{4}$$

$$u \sim N(0, \sigma_u), v \sim N(0, \sigma_v) \tag{5}$$

$$\sigma_u(\beta) = \left[(\Gamma(1 + \beta) \cdot \text{Sin}(\pi \beta/2)) / \beta \Gamma((1 + \beta)/2) \cdot 2^{((\beta-1)/3)} \right]^{(1/\beta)}, \sigma_v = 1 \tag{6}$$

In above equations, u and v represents random vector obeying normal distributions. β (beta) $\in [0.3, 1.99]$ is a distribution’s parameter & usually taken value as $\beta = 3/2$ in CS. Value of v is from 0 to 1 while value of u is from 0 to $\sigma(3/2)$. In Eq. 6, “ Γ ” is gamma function is extended version of factorial function, with argument shifted below by 1. Suppose, if n is positive integer, $\Gamma(n) = (n-1)!$. Steps of Cuckoo search is given below.

5.2 Cuckoo Search Algorithm (Pa)

begin

Input: N, pa

Objective function Minimize $f(x)$, $x = (x_1, x_2, \dots, x_p)^T$

Initial population of N host nests x_i ($i=1, 2, \dots, N$)

And evaluate its fitness f_i

$G_{best} = \min(f_1, f_2, \dots, f_N)$

While ($t < \text{MaxIterations}$ of stop criterion)

Get a host nest randomly say i with fitness f_i

Generate a cuckoo egg corresponding to chosen host nest using eq. 1.1, 1.2 (local random walk)

Evaluate fitness of cuckoo egg say F_i

If ($F_i < f_i$)

Replace x_i by generated cuckoo egg.

Endif

If ($\text{rand} < \text{pa} = 0.25$) // discovering probability is //greater

That nest is abandoned

Built new nest using eq. 2 (global random walk) accept it & Evaluate its fitness F_k .

Endif

$G^* = \min(f_1, f_2, \dots, f_N)$

If ($G^* < G_{best}$)

Update G_{best} , X_{best}

Endif

Endwhile

Output: G_{best} , X_{best}

end

Pros: Xin she yang et al. (creator of CS, FA, FPA) & many other researchers conclude that cuckoo search is better among all existed algorithms in terms of following:

- Levy Flights is considered to have infinite mean and variance.
- Only one parameter that is Pa in CS so, it is generally applicable.
- Good for multimodal problem.

If No. of nest > no. of Local Optima then it can show All optima simultaneously.

6 Experimental Settings

Evaluation and comparisons of CS, GA and Randomized algorithms has been done for Load Balancing in Cloudsim 3.0.3 with the help of a suitable example. Due to space constraints 6 Cloudlets and 2 VMs has been considered in these experiments. At infrastructure level there are 2 processing elements. For every VMs and Cloudlets the required processing element for them is considered one. Out of 2 PEs, one is with 1000 MIPS and other is with 2000 MIPS processing capability. In case of Multipopulation algorithms no. of schedules or size of population has been taken as 4.

Table 2 represents six cloudlets with their size in terms of Millions of instructions and CL-ids.

Table 2. Cloudlets properties

CL-id	Length in MI (millions of instruction)
0	2227
1	1693
2	2725
3	1581
4	2014
5	2386

Table 3 represents two VMs with their processing capabilities MIPS and VM-ids.

Table 3. Processing capabilities of VMs

VM-id	MIPS (millions of instruction per second)
0	1000
1	2000

Table 4 represents a sample Load schedule by using randomized scheduling as example. Below is the sample code for randomized scheduling as example in java.

Table 4. Load schedules by using randomized scheduling

Pop [schedule, CL]	CL0	CL1	CL2	CL3	CL4	CL5
0	0	1	1	1	1	0
1	1	0	0	1	1	0
2	0	0	0	0	1	0
3	1	1	0	0	0	0

Sample Code for Randomized Scheduling:

```

c-No. of Cloudlets=6
v-No. of VMs=2
s-No. of Schedules=4
int pop[ ][ ]=new int[s][c];
Random robj1=new Random();
for (int i=0;i<s;i++)
{
    for (int j=0;j<c;j++)
    {
        pop[i][j]=robj1.nextInt(v+1);
    }
}

```

Where, `robj1.nextInt(v + 1)` returns a pseudo random, which is uniformly distributed int value between 0 (inclusive) and $v + 1$ (exclusive).

7 Experimental Results

In this sections experimental results have been performed corresponding to Average Makespan of schedule of Cloudlets criteria. These have been observed on 3 algorithms (Randomized algorithm, Genetic Algorithm, Cuckoo Search with Levy Flight) in Figs. 1 and 2. To conclude results average of time parameters has been taken corresponding to regularly increase in iterations of load balancing with `CloudletSchedulerSpaceShared` mode and `CloudletSchedulerTimeShared` mode.

Mode 1: `CloudletSchedulerSpaceShared`-

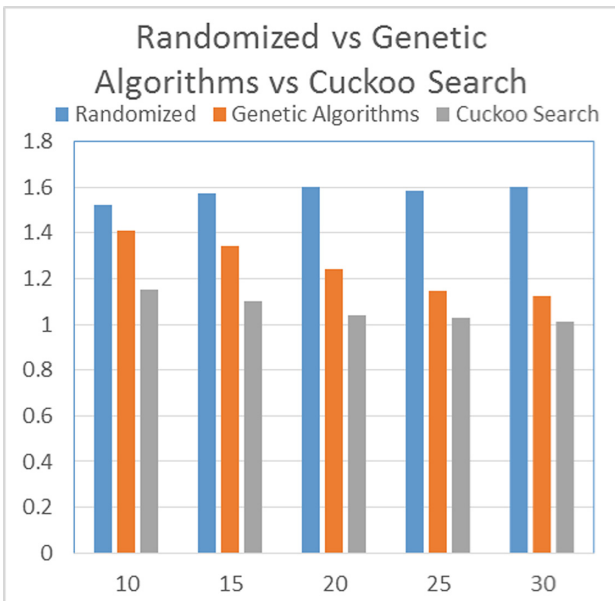


Fig. 1. Comparison of algorithms corresponding to Avg. MSSCL (Makespan time of schedule of Cloudlets) in `CloudletSchedulerSpaceShared` mode

Mode 2: CloudletSchedulerTimeShared-

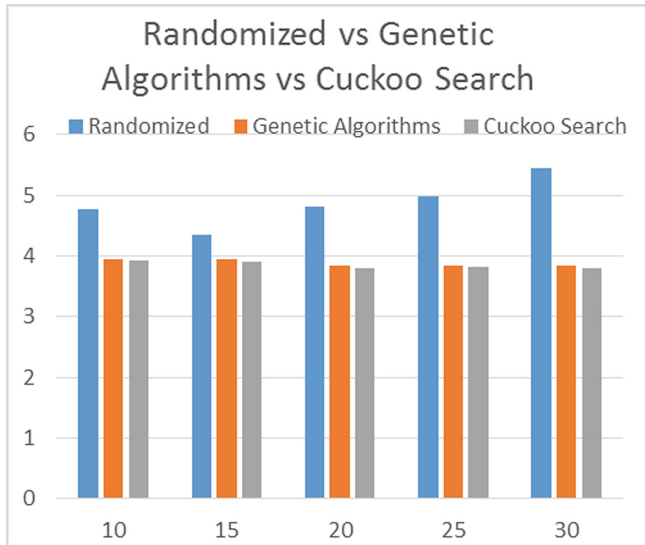


Fig. 2. Comparison of algorithms corresponding to Avg. MSSCL (Makespan time of schedule of Cloudlets) in CloudletSchedulerTimeShared mode.

In all cases X and Y axis are treated as below:

X axis: No. of Iterations the Load Balancing is done.

Y axis: Avg. of Makespan or Response Time of schedules (one in case of single population like randomized schedule or four in case of multi population like genetic algorithm) of Cloudlets after n iterations.

MSSCL-Makespan time of schedule of Cloudlets.

8 Conclusion

In this paper, the need of metaheuristic algorithms, its components, characteristics, classifications, literature survey have been done deeply on fourteen approaches. On the basis of literature survey, comparison has been done among fourteen approaches corresponding to key parameters, main mechanism and used applications areas. Theoretically found Cuckoo search as optimal algorithm out of all due to global levy flight and generality (because of one parameter only). Furthermore, load balancing in cloud computing problem has been solved by randomized algorithm, genetic algorithm and cuckoo search with aim to minimize makespan time to improve quality of services. Results have been compared and concluded that cuckoo search with levy flight is more general and optimal. Furthermore, a lot of work can be done in future to improve its performance by adjusting to optimal value of p_a (discovering probability).

References

1. Boussaid, I., Lepagnot, J., Patrick S.: A survey on optimization metaheuristics. In: Web of Science, Elsevier Information Science, vol. 237(5), pp. 82–117 (2013)
2. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Op. Res.* **13**(5), 533–549 (1986)
3. Holland, J.H.: *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Harbor, MI (1975)
4. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
5. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system- optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **26**(1), 29–41 (1996)
6. Moscato, P., Norman, M.G.: A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. In: *International Conference on Parallel Computing and Transputer Application*, pp. 86–177 (1992)
7. Kennedy, J., Eberhart R.: Particle swarm optimization. In: *IEEE International Conference on Neural Networks*, pp. 1942–1948 (1995)
8. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. In: *IEEE Control Systems Magazine*, pp. 52–67 (2006)
9. Eusuff, M., Lansey, K., Pasha, F.: Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Eng. Optim.* **38**(2), 129–154 (2006)
10. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **8**, 687–697 (2007)
11. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization- artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **39**, 459–471 (2007)
12. Yang, X.S., He, X.: Firefly algorithm- recent advances and applications. *Int. J. Swarm Intell.* **1**(1), 36–50 (2013)
13. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspir. Comput.* **2**(2), 78–84 (2010)
14. Simon, D.: Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **12**(6), 702–713 (2008)
15. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: *IEEE Conference Publication World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214 (2009)
16. Yang, X.S., Deb, S.: Engineering optimisation by cuckoo search. *Int. J. Math. Modell. Numer. Optim.* **1**(4), 330–343 (2010)
17. Yang, X.S., Deb, S.: Cuckoo search- recent advances and applications. *Neural Comput. Appl.* **24**(1), 169–174 (2014)
18. Yang, X.S.: Bat algorithm- literature review and applications. *Int. J. Bio-Inspir. Comput.* **5**(3), 141–149 (2013)
19. Yang, X.S., Karamanoglu M.: Multi-objective flower algorithm for optimization. In: *International Conference on Computational Science, Elsevier Science*, pp. 861–868 (2013)
20. Yang, X.S.: Flower pollination algorithm for global optimization, unconventional computation and natural computation. *Lect. Notes Comput. Sci.* **44**(5), 240–249 (2012)
21. Yang, X.S., Deb, S., Fong, S., Xingshi, H., Zhao, Y.: From swarm intelligence to metaheuristics- nature-inspired optimization algorithms. *IEEE Comput. Soc.* **49**(9), 52–59 (2016)

22. Wang, F., Yang, X.S., Yang, S.M.: Markov model and convergence analysis based on cuckoo search algorithm. *Comput. Eng.* **38**(11), 180–185 (2012)
23. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm- a metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **29**(1), 17–35 (2013)
24. Gandomi, A.H., Yang, X.S., Talatahari, S., Deb, S.: Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. *Comput. Math Appl.* **63**(1), 191–200 (2012)
25. Srivastava, P.R., Chis, M., Deb, S., Yang, X.S.: An efficient optimization algorithm for structural software testing. *Int. J. Artif. Intell.* **9**(12), 68–77 (2012)