International Journal of Electronic and Electrical Engineering.
ISSN 0974-2174 Volume 7, Number 8 (2014), pp. 799-802

http://www.irphouse.com

Comparison of Nature Inspired Metaheuristic Algorithms

Kanika Malik [1] and Akash Tayal [2]

MTech Student ECE, IGDTUW, New Delhi
[2]ECE, IGDTUW, New Delhi

Abstract

Metaheuristics is basically a higher level procedure, which generates a simpler procedure to solve an optimization problem. Optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result. The input consists of variables; the process or function is known as the cost function, objective function, or fitness function; and the output is the cost or fitness. Since cost is something to be minimized, optimization becomes minimization.

By searching over a large set of feasible solutions, metaheuristics can often find good solutions with less computational effort than algorithms, iterative methods, or simple heuristics. It is a refinement to the exhaustive search includes first searching a coarse sampling of the fitness function, then progressively narrowing the search to promising regions with a finer toothed comb. It speeds convergence and increases the number of variables that can be searched but also increases the odds of missing the global minimum.

Metaheuristic algorithms are approximate and non-deterministic and are not problem specific. Metaheuristics are used for combinatorial optimization in which an optimal solution is sought over a discrete search-space, like the travelling salesman problem, where the search-space of candidate solutions grows faster than exponentially as the size of the problem increases, which makes an exhaustive search for the optimal solution infeasible.

Many metaheuristic algorithms have inspiration coming from Nature. There are many such examples where the organisms (a population or a swarm) have optimized and adapted themselves to survive in this world. Some such algorithms are: Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Cuckoo Algorithms and many more.

In this paper, the above mentioned algorithms are applied on Travelling Salesman Problem for comparison of their performance. The evaluation criteria is kept as the ”Time Taken to find the Optimum Solution”as benchmark

Keywords: Metaheuristics, Optimization, Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Cuckoo Algorithm, Travelling Salesman Problem.

Introduction

Optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result. It is basically method to find the best possible solution to a problem out of the various available solutions, for there could be no single right way of doing a thing but there could be multiple possible ways. Optimisation is the process to evaluate the different solution sets on basis of criteria and rank the solutions and thus find the best amongst them.

One of the simplest problem of such kind is Travelling Salesman Problem, in which a salesman has to cover number of cities spread over area and order of covering is to determined for minimising the travelling time.

Algorithms

Genetic Algorithm

Genetic algorithms emulate the evolutionary behaviour of biological systems. They generate a sequence of populations of candidate solutions to the underlying optimization problem by using a set of genetically inspired stochastic solution transition operators to transform each population of candidate solutions into a descendent population. Survival of the fittest translates into discarding the chromosomes with the highest cost.[1]

Two chromosomes are selected from the mating pool of N chromosomes to produce two new offspring. Random mutations alter a certain percentage of the bits in the list of chromosomes. Mutation is the second way a GA explores a cost surface. It can introduce traits not in the original population and keeps the GA from converging too fast before sampling the entire cost surface. A single point mutation changes a 1 to a 0, and visa versa. Mutation points are randomly selected. After the mutations take place, the costs associated with the offspring and mutated chromosomes are calculated, the bottom chromosomes are rejected. The number of generations that evolve depends on whether an acceptable solution is reached or a set number of iterations is exceeded. After a while all the chromosomes and associated costs would become the same if it were not for mutations. At this point the algorithm should be stopped.

Particle Swarm Optimization

The algorithm was inspired by the social behaviour of animals, such as bird flocking or fish schooling. PSO is similar to the continuous GA in that it begins with a random population matrix. Unlike the GA, PSO has no evolution operators such as crossover and mutation. Each particle moves about the cost surface with a velocity. The particles update their velocities and positions based on the local and global best

solution. The PSO algorithm updates the velocity vector for each particle then adds that velocity to the particle position or values. Velocity updates are influenced by both the best global solution associated with the lowest cost ever found by a particle and the best local solution associated with the lowest cost in the present population. If the best local solution has a cost less than the cost of the current global solution, then the best local solution replaces the best global solution.

Ant Colony Optimization

Ants can find the shortest path to food by laying a pheromone (chemical) trail as they walk. Other ants follow the pheromone trail to food. Ants that happen to pick the shorter path will create a strong trail of pheromone faster than the ones choosing a longer path. Since stronger pheromone attracts ants better, more and more ants choose the shorter path until eventually all ants have found the shortest path.

The ACO is a natural for the traveling salesperson problem [1]. It begins with a number of ants that follow a path around the different cities. Each ant deposits a pheromone along the path. The algorithm begins by assigning each ant to a randomly selected city. The next city is selected by a weighted probability that is a function of the strength of the pheromone laid on the path and the distance of the city. Short paths with high pheromone have the highest probability of selection. On the initial paths, pheromone is laid on inefficient paths. Consequently some of this pheromone must evaporate in time or the algorithm will converge on an inefficient path.

Cuckoo Algorithm

It was inspired by the peculiar behaviour of some cuckoo species of laying their eggs in the nests of other host birds (of other species). I f a host bird discovers the eggs are not their own, it will either throw these alien eggs away or simply abandon its nest. Some cuckoo species have evolved in such a way that they specialize in the mimicry in colours and pattern of the eggs of host species [1, 2, 4].

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.

Simulation Result

All the algorithms stated above were simulated in MATLAB for Travelling Salesman Problem for Number of different no. of cities (Nmaximum)=30. The time taken to complete the iterations is used to compare the results. The time is taken form the MATLAB profile which can give execution time of the code. the results are as follows:

Table 1: Simulation Time in seconds for different Algorithms for N no. of cities in TSP

No of cities	GA	ACO	Cuckoo	PSO
N	Total time	Total time	Total time	Total time
5	1.001	1.252	3.038	0.466
10	0.669	3.898	2.076	0.413
20	0.538	16.537	2.833	0.77
30	0.526	38.8	3.983	1.422

Results

With the above results it could be seen that for smaller number of N (cities), the simulation time for Genetic Algorithm and Particle Swarm Optimisation is less but increases with increase in N. Whereas in ACO algorithm, the simulation time explodes with increasing N. At the same time, Cuckoo Search algorithm though gives larger simulation time for small values of N also, but the time remains almost constant with increasing number of cities (N).

Cuckoo search algorithm has lesser controlling parameters as compared to Genetic Algorithm and Particle Swarm Optimisation, thus it could a preferred Algorithm for problem with large iterations required and where the controlling parameters are preferred minimum. For applications requiring less number of iterations and fine control over the search, Genetic Algorithm or PSO is a better option.

References

[1] X. S. Yang, Nature-inspired metaheuristic algorithms, Luniver Press, 2008.

[2] X. S. Yang and S. Deb, Eagle strategy using Levy walk and firefly algorithms for stochastic optimization, in Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, Springer Berlin, vol. 284, pp. 101-111, 2010.

[3] X.-S. Yang and S. Deb,"Engineering optimisation by cuckoo search,"International Journal of Mathematical Modelling and Numerical Optimisation, vol. 1, no. 4, pp. 330–343, 2010.

[4] X.-S. Yang,"Cuckoo search via l´evy flights,"in Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, 2009, pp. 210–214.

[5] M. H. Horng and T. W. Jiang,"The codebook design of image vector quantization based on the firefly algorithm,"Computational Collective Intelligence, Technologies and Applications, LNCS, Vol. 6423, pp. 438-447, 2010.

[6] M. H. Horng,"vector quantization using the firefly algorithm for image compression,"Expert Systems with Applications, Vol. 38, (article in press) 12 Aug. 2011

[7] http://www.math.tamu.edu/~mpilant/math614/chaos_vs_random.pdf