International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 1, January-2014, pp: (502-507), Available online at: **www.erpublications.com**

Analytical Study of Metaheuristic Algorithms and their Applications

Meenakshi

Asst Prof., Dept of CSE, GGIAET, Gurgaon, Haryana, India

In nature-inspired metaheuristic algorithms, two key components are local intensification and global diversification, and their interaction can significantly affect the efficiency of a metaheuristic algorithm. However, there is no rule for how to balance these important components. In this paper, we provide a first attempt to give some theoretical basis for the optimal balance of exploitation and exploration for 2D multimodal objective functions. Then, we use it for choosing algorithm-dependent parameters. Finally, we use the recently developed eagle strategy and cuckoo search to solve two benchmarks so as to confirm if the optimal balance can be achieved in higher dimensions. For multimodal problems, computational effort should focus on the global explorative search, rather than intensive local search.

Keywords: Metaheuristic, Algorithms, Applications, properties, search, problems etc.

1. Introduction

Metaheuristics is a term given to a general class of algorithm used to find solutions to optimisation problems when exact techniques prove inadequate. There are many different metaheuristic algorithms: simulated annealing, tabu search, variable neighborhood search, guided local search, ant colony optimisation, particle swarm optimisation, genetic algorithms, and many many more. In computer science and mathematical optimization, a metaheuristic is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity. Metaheuristics sample a set of solutions which is too large to be completely sampled. Metaheuristics may make few assumptions about the optimization problem being solved, and so they may be usable for a variety of problems.

Compared to optimization algorithms and iterative methods, metaheuristics do not guarantee that a globally optimal solution can be found on some class of problems. Many metaheuristics implement some form of stochastic optimization, so that the solution found is dependent on the set of random variables generated. In combinatorial optimization, by searching over a large set of feasible solutions, metaheuristics can often find good solutions with less computational effort than optimization algorithms, iterative methods, or simple heuristics. As such, they are useful approaches for optimization problems. Several books and survey papers have been published on the subject. Most literature on metaheuristics is experimental in nature, describing empirical results based on computer experiments with the algorithms. But some formal theoretical results are also available, often on convergence and the possibility of finding the global optimum. Many metaheuristic methods have been published with claims of novelty and practical efficacy. Unfortunately, many of the publications have been of poor quality; flaws include vagueness, lack of conceptual elaboration, poor experiments, and ignorance of previous literature. The field also features high-quality research.

A. Genetic algorithm

Genetic Algorithm is a Meta-heuristic algorithm that aims to find solutions to NP-hard problems. The basic idea of Genetic Algorithms is to first generate an initial population randomly which consist of individual solution to the problem called Chromosomes, and then evolve this population after a number of iterations called Generations. During each generation, each chromosome is evaluated, using some measure of fitness. To create the next generation, new chromosomes, called offspring, are formed by either merging two chromosomes from current generation using a crossover operator or modifying a chromosome using a mutation operator. A new generation is formed by selection, according to the fitness values, some of the parents and offspring, and rejecting others so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or suboptimal solution to the problem. The process of GA can be represented as follows:

Step 1: Generate initial population.
Step 2 : Evaluate populations.
Step 3: Apply Crossover to create offspring.
Step 4: Apply Mutation to offspring.
Step5: Select parents and offspring to form the new population for the next generation.
Step 6: If termination condition is met finish, otherwise go to Step2

B. Tabu Search

Tabu search is the technique that keeps track of the regions of the solution space that have already been searched in order to avoid repeating the search near these areas. It starts from a random initial solution and successively moves to one of the neighbors of the current solution. The difference of tabu search from other Meta-heuristic approaches is based on the notion of tabu list, which is a special short term memory. That is composed of previously visited solutions that include prohibited moves. In fact, short term memory stores only some of the attributes of solutions instead of whole solution. So it gives no permission to revisited solutions and then avoids cycling and being stuck in local optima. During the local search only those moves that are not tabu will be examined if the tabu move does not satisfy the predefined aspiration criteria. These aspiration criteria are used because the attributes in the tabu list may also be shared by unvisited good quality solutions. A common aspiration criterion is better fitness, i.e. the tabu status of a move in the tabu list is overridden if the move produces a better solution.

The process of TS can be represented as follows:

Step 1: Generate initial solution x.
Step 2: Initialize the Tabu List.
Step 3: while set of candidate solutions X‟ is not complete.
Step 3.1: Generate candidate solution x‟ from current solution x
Step 3.2: Add x‟ to X‟ only if x‟ is not tabu or if at least one Aspiration Criterion is satisfied.
Step 4: Select the best candidate solution x* in X‟.
Step 5: If fitness(x*) > fitness(x) then x = x*.
Step 6: Update Tabu List and Aspiration Criteria
Step 7: If termination condition met finish, otherwise go to Step 3

C. Simulated annealing

Simulated Annealing is an early Meta-heuristic algorithm originating from an analogy of how an optimal atom configuration is found in statistical mechanics. It uses temperature as an explicit strategy to guide the search. In Simulated Annealing, the solution space is usually explored by taking random tries. The Simulated Annealing procedure randomly generates a large number of possible solutions, keeping both good and bad solutions. As the simulation progresses, the requirements for replacing an existing solution or staying in the pool becomes stricter and stricter, mimicking the slow cooling of metallic annealing. Eventually, the process yields a small set of optimal solutions. Simulated Annealing advantage over other methods is its ability to obviate being trapped in local minima. This means that the algorithm does not always reject changes that decrease the objective function or changes that increase the objective function according to its probability function: p= e Δf/T Where T is the control parameter (analogy to temperature) and Δf is the variation in the objective function. The process of SA can be represented as follows:

Step 1 Compute randomly next position.
Step2 Determine the difference between the next position and current position, call this different delta .
Step3 If delta < 0, the assign the next position to the current position.
Step4 If delta > 0, then compute the probability of accepting the random next position.
Step5 If the probability is < the e^(-delta / temperature), then assign the next position to the current position .
Step 6 Decrease temperature by a factor of alpha.
Step7 Loop to step 1 until temperature is not greater than epsilon.

D. Soft optimization approach

Soft optimization approach (SOA) for solving the MLLS problems is based on a general sampling approach. The main merit of soft optimization approach is that it does not require any structure information about the objective function, so it can be used to treat optimization problems with complicated structures. However, it was shown that random sampling (for example simple uniform sampling) method cannot produce a good solution. Several experiments had been derived to find the characteristics of an optimal solution, and as a result applying the solution structure information of the MLLS problem to the sampling method may produce a better result than that arising from the simple uniform sampling method. A heuristic algorithm to segment the solution space with percentage of number of 1s has been developed and the performance improvement of solving MLLS problem was confirmed. It should be pointed that the SOA based on segmentation still remains the essential property of random sampling but limited with the searching ranges, however the adopted new solution(s) does not remain any information of the old solution(s). Therefore the improvement of solution performance can only be achieved by increasing the numbers of samples or by narrowing the range of segment.

E. Ant colony optimization

A special Ant colony optimization (ACO) combined with linear program has been developed recently for solving the MLLS problem. The basic idea of ant colony optimization is that a population of agents (artificial ants) repeatedly constructs solutions to a given instance of a combinatorial optimization problem. Ant colony optimization had been used to select the principle production decisions, i.e. for which period production for an item should be schedules in the MLLS problems. It starts from the top items down to the raw materials according to the ordering given by the bill of materials. The ant's decision for production in a certain period is based on the pheromone information as well as on the heuristic information if there is an external (primary) or internal (secondary) demand. The pheromone information represents the impact of a certain production decision on the objective values of previously generated solutions, i.e. the pheromone value is high if a certain production decision has led to good solution in previous iterations. After the selection of the production decisions, a standard LP solver has been used to solve the remaining linear problem. After all ants of iteration have constructed a solution, the pheromone information is updated by the iteration best as well as the global best solutions. This approach has been reported works well for small and medium-size MLLS problems. However for large instances the solution method leads to high-quality results, but cannot beat highly specialized algorithms. ACO are solution construction algorithms, which, in contrast to local search algorithms, may not find a locally optimal solution. Many of the best performing ACO algorithms improve their solutions by applying a local search algorithm after the solution construction phase. Our primary goal in this work is to analyze the manufacturing related application capabilities of ACO, hence in this first investigation we do not use local search.

Classifications of Meta-heuristic algorithm

There are a wide variety of metaheuristics and a number of properties along which to classify them.

a) Local search vs. Global search: One approach is to characterize the type of search strategy. One type of search strategy is an improvement on simple local search algorithms. A well known local search algorithm is the hill climbing method which is used to find local optimums. However, hill climbing does not guarantee finding global optimum solutions.Many metaheuristic ideas were proposed to improve local search heuristic in order to find better solutions. Such metaheuristics include simulated annealing, tabu search, iterated local search, variable neighborhood search, and GRASP. These metaheuristics can both be classified as local search-based or global search metaheuristics.Other global search metaheuristic that are not local search-based are usually population-based metaheuristics. Such metaheuristics include ant colony optimization, evolutionary computation, particle swarm optimization, and genetic algorithms.

b) Single-solution vs. Population-based: Another classification dimension is single solution vs population-based searches. Single solution approaches focus on modifying and improving a single candidate solution; single solution metaheuristics include simulated annealing, iterated local search, variable neighborhood search, and guided local search. Population-based approaches maintain and improve multiple candidate solutions, often using population characteristics to guide the search; population based metaheuristics include evolutionary computation,genetic algorithms, and particle swarm optimization. Another category of metaheuristics is Swarm intelligence which is a collective behavior of decentralized, self-organized agents in a population or swarm. Ant colony optimization, particle swarm optimization, social cognitive optimization, PeSOA: Penguins Search Optimization Algorithm, andartificial bee colony algorithms are examples of this category.

c) Hybridization and memetic algorithms: A hybrid metaheuristic is one which combines a metaheuristic with other optimization approaches, such as algorithms from mathematical programming, constraint programming, and machine learning. Both components of a hybrid metaheuristic may run concurrently and exchange information to guide the search.On the other hand, Memetic algorithms represent the synergy of evolutionary or any population-based approach with separate individual learning or local improvement procedures for problem search. An example of memetic algorithm is the use of a local search algorithm instead of a basic mutation operator in evolutionary algorithms.

d) Parallel metaheuristics: A parallel metaheuristic is one which uses the techniques of parallel programming to run multiple metaheuristic searches in parallel; these may range from simple distributedschemes to concurrent search runs that interact to improve the overall solution.

e) Nature-inspired metaheuristics: A very active area of research is the design of nature-inspired metaheuristics. Many recent metaheuristics, especially evolutionary computation-based algorithms, are inspired by natural systems. Such metaheuristics include Ant colony optimization, particle swarm optimization, cuckoo search, and artificial bee colony to cite a few.

Properties of Metaheuristics Algorithm

These are properties that characterize most metaheuristics:

I. Metaheuristics are strategies that guide the search process.
II. The goal is to efficiently explore the search space in order to find near–optimal solutions.
III. Techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.
IV. Metaheuristic algorithms are approximate and usually non-deterministic.
V. Metaheuristics are not problem-specific
VI. They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
VII. The basic concepts of metaheuristics permit an abstract level description.
VIII. Metaheuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.
IX. Today's more advanced metaheuristics use search experience (embodied in some form of memory) to guide the search.

Applications of Metaheuristics

1. Solving NP-hard Optimization Problems:

a) Traveling Salesman Problem (TSP): This is a special case of the travelling purchaser problem and the vehicle routing problem. The TSP has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, such as DNA sequencing. In these applications, the concept city represents, for example, customers, soldering points, or DNA fragments, and the concept distance represents travelling times or cost, or a similarity measure between DNA fragments.

b) Maximum Clique Problem: An undirected graph is formed by a finite set of vertices and a set of unordered pairs of vertices, which are called edges. By convention, in algorithm analysis, the number of vertices in the graph is denoted by n and the number of edges is denoted by m. A clique in a graph G is a complete subgraph of G; that is, it is a subset S of the vertices such that every two vertices in S are connected by an edge in G. A maximal clique is a clique to which no more vertices can be added; a maximum clique is a clique that includes the largest possible number of vertices, and the clique number $\omega(G)$ is the number of vertices in a maximum clique of G.

c) Flow Shop Scheduling Problem: The flow shop scheduling problem is a combinatorial optimization problem known to be NP-hard, which has captured the interest of a great number of researchers. Many different methods have been applied to solve FSSP and have obtained effective results, but these methods are not satisfying. Based on the quantum theory and particle swarm optimization, this paper presents an HQPSO algorithm to solve FSSP.

d) P-Median Problem: The p-median problem is one of the basic models in discrete location theory. As with most location problems, it is classified as NP-hard, and so, heuristic methods are usually used to solve it. Metaheuristics are frameworks for building heuristics.

2. Search Problems in Many Applications:

A. Feature Selection in Pattern Recognition: feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for three reasons:

i. simplification of models to make them easier to interpret by researchers/users,
ii. shorter training times,
iii. enhanced generalization by reducing over fitting

B. Automatic Clustering: The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Evolutionary computation techniques are widely used by researchers to evolve clusters in the complex data sets. However, there is no adequate research progress to determine the optimal number of clusters.

C. Machine Learning (e.g. Neural Network Training): Machine learning is a subfield of computer science(more particularly soft computing) that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning is closely related to (and often overlaps with) computational statistics; a discipline which also focuses in prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is unfeasible.

Conclusions

The consideration of meta-heuristic is widely used in a lot of fields. Different meta-heuristic algorithms are developed for solving different problems, especially combinational optimization problems. In this chapter, we discussed a special case of MLLS problem. First, the general definition of MLLS problem was described. We shown its solution structure and explained its NP completeness. Second, we reviewed the meta-heuristic algorithms which have been use to solve the MLLS problem and pointed their merits and demerits. Based on the recognition, third, we investigated those implement techniques used in the meta-heuristic algorithms for solving the MLLS problems. And two criteria of distance and range were firstly defined to evaluate the effective of those techniques. We briefly discussed the mechanisms and characteristics of the techniques by using these two criteria, and provided simulation experiments to prove the correctness of the two criteria and to explain the performance and utility of them.

References

[1]. Gerald Paul, "Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem ", Operations Research Letters 38 (2010) 577–581, 2010.
[2]. Houck C.R, Joines J.A, Kay M.G, "Characterizing Search Spaces For Tabu Search", Currently under second review in European Journal of Operational Research., 2011.
[3]. Afentakis, P, & Gavish, B. (1986). Optimal lot-sizing algorithms for complex product structures, Operations Research, 34, 237-249.
[4]. Almeder, C. (2010). A hybrid optimization approach for multi-level capacitated lotsizing problems, European Journal of Operational Research, , 200, 599-606.
[5]. Benton, W. C, & Srivastava, R. (1985). Product structure complexity and multilevel lot sizing using alternative costing policies, Decision Sciences, , 16, 357-369.
[6]. Blackburn, J. D, & Millen, R. A. (1985). An evaluation of heuristic performance in multi-stage lot-sizing systems, International Journal of Production Research, , 23, 857-866.
[7]. Coleman, B. J, & Mcknew, M. A. (1991). An improved heuristic for multilevel lot siz- ing in material requirements planning. Decision Sciences, 22, 136-156.

[8]. Crowston, W. B, & Wagner, H. M. (1973). Dynamic lot size models for multi-stage as- sembly system, Management Science, , 20, 14-21.
[9]. Osman, I.H., and Laporte,G. "Metaheuristics:A bibliography". Ann. Oper. Res. 63, 513–623, 1996.
[10]. Blum, C., and Andrea R. "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison". ACM Computing Surveys, 35(3), 268–308, 2003.