ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 7, July 2017

Meta Heuristics Approach For A Class Of Supply Chain Optimization Problems

[1] D.Sabareswari, [2]N.Sathya, [3]K.Sharmila,
[1]PG research scholar, School of computing sciences, Vels University, Chennai, India.
[2]Research Scholar, Bharathiar University,Coimbatore,India.
[2]Assistant Professor, Department of Computer science, Vels University, Chennai, India.
[3]Assistant Professor, Department of Computer Science, Vels University, Chennai, India.

Abstract: ***In computer science, artificial intelligence, and mathematical optimization, a heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. This is achieved by trading optimality, completeness, accuracy, or precision for speed. A heuristic function, also called simply a heuristic, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow. In highly competitive and global marketplace the pressure on organizations to find new ways to create and deliver value to customers grows. The key to success in Supply Chain Management require heavy prominence on integration of activities, cooperation, coordination and information sharing throughout the entire supply chain, from suppliers to customers. The industry and the academia have become increasingly interested in SCM to be able to respond to the problems and issues posed by the changes in the logistics and supply chain. We present a brief discussion on the important issues in Supply Chain Management. We then argue that metaheuristics can play an important role in solving complex supply chain related problems derived by the importance of designing and managing the entire supply chain as a single entity. We will focus specially on the Simulated Algorithm (SA), Tabu Search and Genetic Algorithms (GAs) as the ones, but not limited to, with great potential to be used on solving the Supply Chain Management related problems.***

Keywords- ***Supply Chain Management, Metaheuristics, Genetic Algorithm, Tabu search, Simulated Algorithm.***

I. INTRODUCTION

Problems of combinatorial optimization are characterized by their well-structured problem definition as well as by their huge number of action alternatives in practical application areas of reasonable size. Especially in areas like routing, task allocation, or scheduling such kinds of problems often occur. Their advantage lies in the easy understanding of their action alternatives and their objective function. Therefore, an objective evaluation of the quality of action alternatives is possible in the context of combinatorial optimization problems. Utilizing classical methods of Operations Research (OR) often fails due to the exponentially growing computational effort. Therefore, in practice heuristics and meta-heuristics are commonly used even if they are unable to guarantee an optimal solution. Artificial Intelligence Heuristics otherwise called as Meta-heuristic techniques that mimic natural processes, developed over the last thirty years and have produced 'good' results in reasonable short runs for this class of optimization problems. Even though those bionic heuristics are much more flexible regarding modifications in the problem description when being compared to classical problem specific heuristics they are often superior in their results. Those bionic heuristics have been developed following the principles of natural processes: In that sense, Genetic Algorithms (GAs) try to imitate the biological evolution of a species in order to achieve an almost optimal state whereas Simulated Annealing (SA**)** was initially inspired by the laws of thermodynamics in order to cool down a certain matter to its lowest energetic state. Many optimization problems of practical as well as theoretical importance consist of the search for a “best” configuration of a set of variables to achieve some goals. They seem to divide naturally into two categories: those where solutions are encoded with *real-valued* variables and those where solutions are encoded with *discrete* variables. Among the latter ones we find a class of problems called Combinatorial Optimization (CO) problems.

II. Scope and Objectives of paper

A careful analysis of literature on the variants and methodologies of combinatorial optimization problems reveals the following: Some of the variants of combinatorial optimization problems are yet to be explored to solve using meta-heuristics techniques in the literature. These include Multiple Traveling Salesman Problem (mTSP) and Vehicle routing problem. Many of the authors (Potvin *et al.* 1996; Duhamel *et al.* 1997; Fisher and Jaikumar,1981; Toth and Vigo, 1999) have suggested the use of a constructive heuristic to obtain good initial solutions for a meta-heuristic so that its convergence can be accelerated.Only a few authors have considered the use of hybrid approaches to solve different variants of combinatorial optimization problems. Glover *et al.* (1995) and Osman and Kelly (1997) have pointed out that hybrid approaches focus on enhancing the strengths and compensating for the weaknesses of two or more complementary approaches. The aim is the generation of better solutions by combining the key elements of competing methodologies. The quality of solutions obtained by many of the proposed heuristic methods has not been established through comparative evaluation with optimal solutions. While meta-heuristics can yield better solutions, the computational effort required by them often inhibits their use. There is scope for the application of multi-phase heuristics that use a combination of intuitive and classical methods to construct good initial solutions which, in turn, serve as inputs for an intensive search using meta-heuristics. This could yield quality solutions at reasonable computation time (Johnson 1990).

III. Problem Background

The Traveling Salesman Problem (TSP) is arguably the most prominent problem in combinatorial optimization. The simple way in which the problem is defined in combination with its notorious difficulty has stimulated many efforts to find an efficient solution procedure. The TSP is a classic tour problem in which a hypothetical salesman must find the most efficient sequence of destinations in his territory, stopping only once at each, and ending up at the initial starting location. Connections between pairs of cities are called edges and each edge has a cost associated with it which can be distance, time or other attribute. If n is the input number of vertices representing cities, for a weighted graph G, the TSP problem is to find the cycle of minimum costs that visit each of the vertices of G exactly once (Skiena 1997). The traveling salesman problem (TSP) is one of the most intensively studied problems in computational mathematics and combinatorial optimization. It is also considered as the class of the NP complete combinatorial optimization problems. By literatures, many algorithms and approaches have been launched to solve such the TSP. However, no current algorithms that can provide the exactly optimal solution of the TSP problem are available. This paper proposes the application of AI search techniques to solve the TSP problems. Three AI search methods, i.e. genetic algorithms (GA), Tabu search (TS), and adaptive Tabu search (ATS), are conducted**.** They are tested against ten benchmark real-world TSP problems. As results compared with the exactly optimal solutions, the AI search techniques can provide very satisfactory solutions for all TSP problem.

3.1 TSP as a Combinatorial Optimization Problem

If n is the number of cities to be visited for the TSP then $(n\text{-}1)!$ is the total number of possible routes. Following this basic formulation, an exponential relationship exists between the number of cities and possible routes, for instance if there are 5 cities there are 24 possible routes, for 6 cities 120, for 10 cities 362,880, and so on. As the amount of input data increases the problem increases in complexity, thus the computational time needed renders this method impractical for all but a smaller number of cities. Rather than considering all possible tours, heuristic algorithms for solving the TSP are capable of substantially reducing the number of tours to be taken into consideration. A heuristic is considered “good” if the number of elementary computational steps is bounded by a polynomial in the size of the problem. (Lawler 2000). NP-complete problems are non-deterministic problems that cannot be solved in polynomial time (reasonable computing time, short time), where non-deterministic means that taking a guess in the solution is involved. However such efficient algorithm has not yet been found (Hoffman K 1996). The challenge of combinatorial optimization is to develop algorithms that use a reasonable amount of computer time. This challenge is of interest to mathematicians as well as to computer scientists. If a solution to this general problem is possible, it is expected to come from the study of fundamental combinatorial optimization rather than advances in computer technology (Lawler 2000). A prize of $1 million for solving the TSP, as one representative problem of a larger class of NP-complete combinatorial optimization problems, has been offered by the Clay Mathematics Institute of Cambridge, Massachusetts (CMI 2006).

3.2 Solution Procedures for the TSP

Efforts have concentrated on the development of heuristics that are not guaranteed to find the shortest tour, but are likely to quickly find either the optimal solution or a near-optimal alternative. The two most important categories in which solution procedures are classified are 1) exact and 2) approximate approaches.

1) Exact Solution Procedures – Integer Programming

Exact approaches to solving the TSP are successfully used only for relatively small problem sizes but they can guarantee optimality based on different techniques. These techniques use algorithms that generate both a lower and an upper bound on the true minimum value of the problem instance. If the upper and lower bound coincide, a proof of optimality is achieved. The standard technique for obtaining lower bounds on the TSP problem is to use a relaxation that is easier to solve than the original problem. A circular tour that visits every city once is a valid solution (Hoffman K 1996). The Cutting Plane method (Dantzig G 1954) models the problem as non integer linear program then solves the linear relaxation of it, where relaxations are gradually improved to give better approximations. The fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. The Branch and Bound method (Skiena 1997) performs a combinatorial search while maintaining upper and lower bounds on the cost of a tour or partial tour. Branching consists in splitting the feasible regions in smaller subregions. The Simplex method for linear programming (Danztig G 1949) is combined with Branch and Bound to solve the problem by systematically moving from one solution to another until the optimal is found, but it is unable to solve large instances (J. R. Evans 1992).

2) Approximate Solution Procedures - Heuristics

Heuristic solution procedures cannot guarantee optimality. They are approximate approaches based on algorithms that construct feasible solutions within reasonable computing time (Hoffman K 1996). As a result two very important criteria are considered in all the proposed heuristics: 1) speed, meaning the total computational time, and, 2) closeness to optimal solution meaning how far away in percentage from the optimal solution. The two main categories in which heuristics are classified are 1) constructive and 2) improvement. Constructive heuristics are algorithms that try to build up feasible solutions step by step and then stop when a solution is found and never try to improve upon its solution (Hoffman K 1996).These algorithms are seeking for the maximum benefit at each step (greedy) and iteratively construct tours. The Nearest Neighbor method is based on the idea to always visit the closest city. The Greedy heuristic gradually constructs a tour by repeatedly selecting the shortest edge and adding it to the tour. The Minimum Spanning Tree is a popular heuristic, where all the points are connected together by a subgraph called tree in such way that cost is minimized. The Insertion technique has many variants to insert new points (closest or farthest) into a partial tour one at a time, until the tour is complete. Improvement heuristics are algorithms that start with an initial feasible solution and successively improve it through a sequence of exchanges (Evans J. 1992). The initial solution can be chosen at random by means of nearest neighbor, for example. As such an initial feasible solution can be one that does not violate any of the constraints. In the TSP problem the constraints are to visit each of the cities exactly once and to return to the starting city. Improvement heuristics all fit within these simple steps:

1) Generate a feasible solution.
2) Try to find an improved possible solution by means of some transformation.
3) If a better solution is found replace the current one and repeat from step 2.
4) If a better solution cannot be found then the current solution is the local optimal solution.

The most common ways to improve an initial tour generated by construction heuristics are the two-optimal (2-opt) and three-optimal (3-opt) local searches. The 2-opt algorithm basically removes two edges from the tour then reconnects the two paths created. The 3-opt algorithm works in a similar way but does so by removing three edges. Lin-Kernighan algorithm uses a more complex edge exchange procedure where the number *k* of the edges to be exchanged is variable (Voudouris and Tsang 1999). Iterated Lin-Kernighan (LK) algorithm first obtains a local minimum then examines other local minimum tours near the current local minimum. Iterated LK used to be considered one of the best heuristic to solve the TSP, until metaheuristics were developed. A metaheuristic is a set of concepts that can be used to guide the other heuristics to find their way out of local optima by continuing the search for better areas of the solution space. (Voudouris and Tsang 1999).Local optima is a solution optimal within a limited area of the solution space while global optima is a solution optimal for the entire solution space. Examples of metaheuristics include simulated annealing, tabu search, iterated local search, evolutionary algorithms, genetic algorithms and ant colony optimization.The Tabu Search is widely considered to be the best approach to solving large vehicle routing problems. It performs best improvement local search selecting the best move in the neighborhood but only amongst those excluded by the tabu list which contains forbidden solution. Tabus may sometimes prohibit attractive moves, even when there is no danger of cycling. It is imperative therefore, to add some criteria that will allow the search to override the tabu list. These are called aspiration criteria.

IV. Metaheuristic

Algorithms with stochastic components were often referred to as heuristic in the past, though the recent literature tends to refer to them as metaheuristics. We will follow Glover's convention and call all modern nature-inspired algorithms metaheuristics .The word "metaheuristic" was coined by Fred Glover in his seminal paper, and a metaheuristic can be considered 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". In addition, all metaheuristic algorithms use a certain tradeoff of randomization and local search. Quality solutions to difficult optimization problems can be found in a reasonable amount of time, but there is no guarantee that optimal solutions can be reached. It is hoped that these algorithms work most of the time, but not all the time. Almost all metaheuristic algorithms tend to be suitable for global optimization. Two major components of any metaheuristic algorithms are: intensification and diversification, or exploitation and exploration. Diversification means to generate diverse solutions so as to explore the search space on a global scale, while intensification means to focus the search in a local region knowing that a current good solution is found in this region. A good balance between intensification and diversification should be found during the selection of the best solutions to improve the rate of algorithm convergence. The selection of the best ensures that solutions will converge to the optimum, while diversification via randomization allows the search to espace from local optima and, at the same time, increases the diversity of solutions. A good combination of these two major components will usually ensure that global optimality is achievable.

4.1 Metaheuristic Algorithms for Optimization

- Genetic Algorithm
- Tabu Search
- Differential Evolution

- Ant Colony Optimization
- Bee Algorithm
- Particle Swarm Optimization
- Cuckoo Search
- Harmony Search
- Firefly algorithm
- Simulated Annealing

V. Genetic Algorithms:

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The GA algorithm repeatedly modifies a population of individual solutions.

5.1 History of Genetic algorithm

Genetic algorithms were invented to mimic some of the processes observed in natural evolution. The idea with GA is to use this power of evolution to solve the optimization problems. The father of the original Genetic algorithm was John Holland who invented it. As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments.By the 1975, the publication of the book “Adaptation in Natural and Artificial Systems”, by Holland and his students and colleagues. Early to mid-1980s, genetic algorithms were being applied to a broad range of subjects. In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called his method "genetic programming" (GP).

5.2 What is Genetic Algorithm?

A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic Algorithm are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they take advantage of historical information to direct the search into the region of better performance within the search space.

5.3 Why Genetic Algorithms?

It is better than conventional AI in that it is more robust. Unlike older AI systems, they do not break easily even if the inputs changed slightly, or in the presence of reasonable noise. Also, in searching a large state-space, multi-modal state-space, or n-dimensional surface, a genetic algorithm may offer significant benefits over more typical search of optimization techniques.

5.4 Genetic Algorithm steps

The process of Genetic Algorithm can be represented as follow:

Step 1: Generate initial population.
Step 2: Evalaute the populations.
Step 3: Apply the Crossover to create offspring.
Step 4: Apply Mutation to offspring.
Step 5: Select parents and offspring to form the new population for the next generation.
Step 6: If termination condition is met finish, otherwise go to step 2.

5.5 Advantages:

It can solve every optimization problem which can be described with the chromosome encoding. It solves problems with multiple solutions. Since the genetic algorithm execution technique is not dependent on the error surface, we can solve multi-dimensional, non-differential, non-continuous, and even non-parametrical problems. Structural genetic algorithm gives us the possibility to solve the solution structure and solution parameter problems at the same time by means of genetic algorithm. Genetic algorithm is a method which is very easy to understand and it practically does not demand the knowledge of mathematics.

5.6 Disadvantages:

There is no absolute assurance that a genetic algorithm will find a global optimum. It happens very often when the populations have a lot of subjects. Like other artificial intelligence techniques, the genetic algorithm cannot assure constant optimization response times. Even more, the difference between the shortest and the longest optimization response time is much larger than with conventional gradient methods. Genetic algorithm applications in controls which are performed in real time are limited because of random solutions and convergence, in other words this means that the entire population is improving, but this could not be said for an individual within this population.

VI. Tabu search

Tabu search, created by Fred W. Glover in 1986 and formalized in 1989, is a metaheuristic search method employing local search methods used for optimization. Local take a potential solution to a problem and check its immediate neighbors in the hope of finding an improved solution. The word tabu comes from Tongan, a language of Polynesia, used by the aborigines of Tonga to indicate things that cannot be touched because they are consecrated. TS is a metaheuristic algorithm that can be used for solving combinatorial optimization problems. Current applications of TS span the areas of resource planning, telecommunications, VLSI design, financial analysis, scheduling, space planning, energy distribution, molecular engineering, logistics, pattern classification, flexible manufacturing, waste management, mineral exploration, biomedical analysis, environmental conservation and scores of others.

6.1 What is Tabu search?

Tabu search is a metaheuristic algorithm that can be used for solving combinatorial optimization problems that is problems where an optimal ordering and selection of options is desired.The method is based on procedures designed to cross boundaries of feasibility or local optimality, instead of treating them as barriers. Three main strategies:

- **Forbidding strategy**: control what enters the tabu list
- **Freeing strategy**: control what exits the tabu
- **Freeing strategy**: control what exits the tabu list and when
- **Short-term strategy**: manage interplay between the forbidding strategy and freeing strategy to select trial solutions

6.3 Parameters of Tabu Search

- Local search procedure
- Neighborhood structure
- Aspiration conditions
- Form of tabu moves
- Addition of a tabu move
- Maximum size of tabu list
- Stopping rule

6.4 Tabu Search Steps

The process of Tabu Search 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.

3.1: Generate candidate solution x" from current solution x" from current solution x.

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 Step3.

VII. Simulated Annealing Algorithm

A Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space. It is often used when the search space is. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to alternatives such as descent. The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Both are attributes of the material that depend on its thermodynamic free energy. Heating and cooling the material affects both the temperature and the thermodynamic free energy. SA has been successfully adapted to give approximate solutions for the SP. SA is basically a randomized local search algorithm allowing moves with negative gain. A baseline implementation of SA for the TSP is presented in. They use 2-opt moves to find neighboring solutions. Not surprisingly the resulting tours are comparable to those of a normal 2-opt algorithm. Better results can be obtained by increasing the running time of the SA algorithm, showing results comparable to the LK algorithm. Due to the 2-opt neighborhood, this particular implementation takes $O(n2)$ with a large constant of proportionality. Some speed-ups are necessary to make larger instances feasible to run, and also to make it competitive to other existing approximation algorithms. The first thing to improve is the 2-opt neighborhood. By keeping neighborhood lists as described in section 4.2, one can cut down this part dramatically. The process of SA Search can be represented as follows:

Step 1: Compute randomly next position.

Step 2: Determine the difference between the next position and current position, call this different delta.

Step 3: If delta < 0, the assign the next position to the current position .

Step 4: If delta > 0, then compute the probability of accepting the random next position .

Step 5: 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 .

Step 7: Loop to step 1 until temperature is not greater than epsilon

VIII. Simulated Annealing, Genetic algorithms and tabu search

Genetic algorithms,Simulate Annealing and tabu search have a number of considerable differences. They also have some common bonds, often unrecognized. We walk around the nature of the connections between the methods, and show that a variety of opportunities exist for creating hybrid approaches to take advantage of their opposite features. Tabu search has pioneered the systematic exploration of memory functions in search processes, while genetic algorithms have pioneered the implementation of methods that exploit the idea of combining solutions. There is also another approach, related to both of these, that is frequently overlooked. The procedure called scatter search, whose origins overlap with those of tabu search also proposes mechanisms for combining solutions, with useful features that offer a bridge between tabu search and genetic algorithms. Recent generalizations of scatter search concepts, embodied in notions of structured combinations and path relinking, have produced effective strategies that provide a further basis for integrating GA and TS approaches. A prominent TS component called strategic oscillation is susceptible to exploitation by GA processes as a means of creating useful degrees of diversity and of allowing effective transitions between feasible and infeasible regions. The independent success of genetic algorithms and tabu search in a variety of applications suggests that each has features that are valuable for solving complex problems. The thesis of this paper is that the study of methods that may be created from their union can provide useful benefits in diverse settings.

IX. Conclusion and Future work

In this paper, we applied Genetic algorithm (GA), tabu search (TS), and simulated annealing (SA) as Meta-heuristic algorithms for solving the TSP. This research is dedicated to compare the relative percentage deviation of these solution qualities from the best known quality solution which is introduced in TSP. The results show that GA, TS, and SA algorithms have effectively demonstrated the ability to solve TSP optimization problems. The computational results show that genetic algorithm has a better solution quality than the other Meta-heuristic algorithms for solving TSP problems. Tabu search algorithm has a faster execution time than the other Meta-heuristic algorithms for solving TSP problems.

In future research, comparisons between Meta-heuristic algorithms for more different types, different sizes of TSP instances and different algorithms can be conducted. Also apply Meta-heuristic algorithms to solve other combinatorial problems such as container terminals problems.

Reference:

1. D.S. Johnson and L.A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local optimization",November 20, 1995.
2. D.S. Johnson and L.A. McGeoch, "Experimental Analysis of Heuristics for the STSP", *The Traveling Salesman Problem and its Variations*, Gutin and Punnen (eds), Kluwer Academic Publishers, 2002,pp. 369-443.
3. M.L Fredman, D.S. Johnson, L.A. McGeoch, G. Ostheimer, "Data Structures For Traveling Salesmen", *J. ALGORITHMS 18*, 1995, pp. 432-479.
4. D.S. Johnson, L.A. McGeoch, E.E. Rothberg, "Asymptotic Experimental Analysis for the Held-Karp Traveling Salesman Bound" *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 1996, pp. 341-350.
5. K. Helsgaun, "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic", Department of Computer Science, Roskilde University.
6. D. Applegate, W. Cook and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems",July 27, 2000.
7. W. Zhang, "Depth-First Branch-and-Bound versus Local Search: A Case Study", Information Sciences Institute and Computer Science Department University of Southern California.
8. M. Dorigo, L.M. Gambardella, "Ant Colonies for the Traveling Salesman Problem", Universit Libre de Bruxelles, Belgium, 1996.
9. S. Arora, "Polynomial Time Approximation Schemes for Euclidian Traveling Salesman and Other Geometric Problems", *Journal of the ACM*, Vol. 45, No. 5, September 1998, pp. 753-782.
10. Abeer M.Mohamound,"A Comparative study of Metaheuristic Algorithm for solving Quadratic Assignment Problem",Journal of ACCA,vol 5,No 1,2014.
11. Gerald Paul, "Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem ", Operations Research Letters 38 (2010) 577–581, 2010.
12. John Silberholz and Bruce Golden, "Comparison of Meta-heuristic " Handbook of Meta-heuristic algorithms International Series in perations Research & Management Science Volume 146, pp 625-640,2010.

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.