ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

ANALYSIS THE PERFORMANCE OF MINING BASED ANT COLONY OPTIMIZATION WITH ANT COLONY OPTIMIZATION

G.SRINIVASU

DEPT. OF MATHEMATICS, RSR ENGINEERING COLLEGE, KADANUTHALA, SPSR NELLORE DIST., A.P., INDIA.
E mail: gandrakota_srinu@yahoo.co.in

INTRODUCTION

The Web mining plays an important role of knowledge discovery. Therefore, this study intends to propose a novel framework of mining which clusters the data first and then followed by association rules mining. In addition to sharing and applying the knowledge in the community, knowledge discovery has become an important issue in the knowledge economic era. Ant Optimization, in the mathematical sense, is the process of finding solutions to a problem such that one or many objectives are minimized or maximized. The novel method efficiently combines the rectangle packing method with ACO and improves the scheduling results by dynamically choosing the test-access-mechanism widths for cores and changing the testing orders. The method is also combined in ant clustering algorithm to discover mining patterns (data clusters) and a linear genetic programming approach to analyze the visitor trends. The visitor entry should be converted in to objectives. These objectives problems are minimized and maximized solution in the mining based ant colony optimization. Deterministic algorithms usually have set execution schedules and are fairly exhaustive search methods. Non-deterministic algorithms use randomness and prove useful for problems where it may not be possible to execute a deterministic algorithm due to the size on the value entries, or nature of the problem search space. In these cases a deterministic algorithm may take days or months to find an optimal solution,

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

whereas a non-deterministic algorithm can usually find an approximate but still near-optimal solution in a matter of minutes or seconds. The Mining-based Ant Colony Optimization (MACO) algorithm is a recently developed ant inspired algorithm which, unlike traditional ACO algorithms, maintains finite visitors of solutions as well as pheromone information. It has been demonstrated to be an efficient optimization algorithm when applied to a range of difficult single-objective, multi-objective and dynamic problem instances. In this paper a review of existing ACO algorithms is offered and an identification of common features is used in the development of a Mining ACO framework. An empirical analysis of these novel implementations is presented using a variety of single and multiple objective continuous function and combinatorial optimization problems on the based visitor entries. These optimization problems have been chosen since they demonstrate the advantages and disadvantages of adding MACO algorithm. To conclude, two of these new MACO algorithms are applied to a real-world optimization problem.

As humans in an increasingly busy world we are confronted with optimization problems on a daily basis. We are often striving to solve these problems more efficiently and effectively, regardless of whether we are consciously aware of it or not. The simple task of commuting between our homes and workplace can be treated as an optimization problem since we may seek to minimize the time taken (objective value) to perform this commute. Furthermore there is a finite (albeit large) set of decision variables we can choose from such as roads, bus/train routes and walking paths. Yet while this simple and sometimes mundane task may seem trivial for us as humans to solve, it can prove to be quite difficult for a computer. Although real world optimization problems come in a variety of different forms, optimization algorithm practitioners usually treat these

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

depending on the interplay between problem variables. Consequently, two very popular and well studied problem classes are function optimization, and combinatorial optimization. The first class is concerned with selecting values for a finite set of problem variables, defined along either a discrete or continuous range. The later class includes problems were not only the value but also the ordering of variables is important, e.g. Travelling Salesman Problem, Shortest Path Problem, University Timetabling Problem, etc. Methods used to solve optimization problems are usually defined into one of two categories: deterministic and non-deterministic algorithms.

Deterministic algorithms are usually well de-fined and understood since their deterministic nature allows for more accurate analysis and estimation of performance. Non-deterministic (stochastic) algorithms are not always understood, and hence performance estimations for these algorithms are usually given as confidence measures. Non-deterministic algorithms are useful for problems where it may not be possible to execute a deterministic algorithm due to the size, or nature of the problem's search space. Such problems are usually denoted as NP-hard or NP-complete. In these cases a deterministic algorithm may take days or months to find an optimal solution, whereas a non-deterministic algorithm can usually find an approximate but hopefully still near-optimal solution in a matter of minutes or seconds. Ant Colony Optimization (ACO) is a non-deterministic algorithm class that is based on the foraging

Behaviors' of Argentine ants. ACO algorithm aim to mimic (and exploit) the behaviors of real ant colonies in order to solve optimization problems. ACO algorithms belong to the class of constructive heuristic algorithms which work by building solutions to a given optimization problem one solution component at a time according to a defined set of rules (Mining Algorithm). In other words these

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

algorithms start with an 'empty' solution and add solution components one at a time until a complete solution is built. ACO algorithms are characterized by this solution construction and by their use of past solutions in manipulating an artificial 'pheromone'. This pheromone is a numeric value which is associated to every unique solution component. It reflects the estimated utility of each unique solution component. These pheromone values are used to bias solution construction by influencing the probability of a solution component being added to a growing solution based on the magnitude of the pheromone value. The ability of ACO algorithms to solve more difficult artificial problem instances is important for researchers, as these difficult artificial problems are often close approximations of industrial (real-world) applications. However, as the complexity of the problem increases, the optimization performance of many standard ACO algorithms will often decrease. To address this decrease in performance, practitioners often make augmentations to standard ACO algorithms in an attempt to increase their performance on specific problem, usually through the introduction of complex operations and extra problem specific parameters. These modifications generally adjust the type or amount of problem specific information that the algorithm has access to, or the algorithm behavior to balance the amount of computation time spent:

1. Searching for solutions radically different from those already found (exploration).
2. Exploiting information learnt through previously evaluated solutions (exploitation).

Exploration and exploitation somewhat define the difference between the various available algorithms which comprise the field of computational intelligence. It is the (often dynamic) balance between these behaviors that can define specific algorithms suitability on a given problem.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

1.1 Travelling Salesman Problem (TSP)

The Travelling Salesman Problem (TSP) can be described as: Given a set of n cities (vertices) and weights for each pair of cities, find a round-trip of minimal total weight that visits every city exactly once. The total number of feasible solutions of a symmetric TSP, which is a TSP for which the weight connecting any two cities is the same regardless of direction of travel, is that of (2.2.1). Given the development of many very good heuristics for the TSP (Nearest Neighbor, Lin-Kernighan) the size of the search space required to locate the optimal solution and thus the search complexity is much lower than the size of the feasible solution space. Even given these heuristics though, the problem is still classified as NP-hard.

$$\textit{Total Solutions} = \frac{(n-1)!}{2}$$

A suite of standard TSP instances is available from an online reference, TSPLIB, and includes a variety of datasets ranging in size and representation. TSP datasets are usually represented in a standard coordinate data system such as 2-dimensional Euclidean, or geographical (using latitude and longitude). This representation is mostly chosen given the nature of the dataset itself, e.g. Burma14 is a dataset which represents 14 cities in Burma using geographical coordinates. Using an appropriate formula the distances connecting each vertex is calculated and stored in a matrix .

City	X	Y
1	1	49
2	14	26
3	-20	43

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

4	30	32
5	27	-44

Table1: 1 city TSP represented using 2D Euclidean coordinates

Examples of the TSP can be found in more practical scenarios than just the original context of a travelling salesman. Printed Circuit Board manufacture often requires the drilling of a number of holes and/or placement of a number of components (Equation.5.1). To minimize the time required to drill/place we can interpret the locations as 'cities' and minimize the problem to minimize.

City (to/from)	1	2	3	4	5
1	0.00	26.42	21.84	33.62	96.57
2	26.42	0.00	38.01	17.09	71.20
3	21.84	38.01	0.00	51.20	98.88
4	33.62	17.09	51.20	0.00	76.06
5	96.57	71.20	98.88	76.06	0.00

Table 2: 5 city TSP edge weight Matrix

the time required to process each PCB. Solving this problem can save a manufacturer much time due to the large number of PCBs processed. Since hundreds of thousands of identical PCBs may be manufactured in a single batch, a small saving in time to manufacture a single PCB may scale to a large saving in the total processing time.

1.2 Pheromone mapping

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

The pheromone mapping means by which solution components are able to be ranked and selected based on past usefulness. The pheromone mapping connects pheromone values from a pheromone map (usually a matrix structure) to specific solution components. The assumption usually being that if a prior solution is good then at least some of its parts (solution components) should also be good and therefore a remixing of these components with other good components may lead to an optimal or near-optimal solution. A first step in defining an ACO algorithm is to define the pheromone mapping. The problem domain will dictate how the pheromone mapping should be defined. In applying an ACO algorithm to a combinatorial optimization problem such as the travelling salesman problem (TSP) it is not of interest which specific components are included, as any feasible solution will include every city once (and only once), it is the order of these components which is important in finding an optimal solution. For the TSP the transition points or edges between the specific components can be assigned a specific pheromone value in order to reflect which order of cities works the best. That is, that if a solution included an edge connecting city to city b and the solution is good then this should be reflected in the pheromone level on this specific edge and the other edges included in the solution.

(a) MAX-MIN Ant Systems

The MAX-MIN Ant Systems algorithm (MMAS) improved the Ant System algorithm by introducing pheromone thresholds to counter premature convergence observed in AS. This thresholding is achieved through the introduction of upper and lower pheromone bounds, t_{min} and t_{max}. The AS global pheromone update is modified to that of ACS and as in ACS only the global best solution is used to apply this update. Pheromone decay is the same as that of AS. When applying

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

update and decay any individual pheromone value is restricted to the range [t_{min}, t_{max}]. Guidelines for determining the values of *t*min and *t*max are outlined in. In early works on MMAS the standard AS random proportional rule is used to determine transition probabilities; however in later works some researchers opt to use the ACS pseudo-random proportional rule instead. MMAS employs a pheromone re-initialisation scheme which upon detection of convergence initialises all pheromone values to *t*max, while retaining the best solution found so far.

(b) DACO for Multi-objective Problems

When applied to multi-objective problems DACO maintains a different pheromone matrix for each objective. For each iteration of the algorithm, where iteration refers to every artificial ant creating a complete solution, a random ant is selected from the population (Q) along with its k closest neighbours2 to form a Mining Technique on Sub - Population P. At any time Q will contain the complete set of non-dominated solutions found to date. The ants in P are then used to update the individual pheromone matrix for each objective. When available a separate heuristic matrix is used for each objective, e.g. in the case of the TSP these heuristic matrices are simply the corresponding edge weights for each individually defined TSP. DACO uses an average-rank-weight method to weight the importance of each objective. These weightings (w) are used to bias the solution construction towards satisfying specific objectives over others. Briefly, the average-rank-weight method measures how well each solution in P satisfies each individual objective. Objectives which are better satisfied by the solutions in P relative to the entire population Q are given a higher rank and a subsequently larger weighting. This approach was shown to be among the state-of-the-art ACO

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

approaches for the multiobjective TSP in. It was conjectured that the good performance of the algorithm can be attributed to the algorithm's ability to target specific areas of the approximate Pareto front for improvement. This is possible since the algorithm is able select few solutions from a larger population to create a temporary pheromone map.

(c) Genetic Algorithm (GA) Similarities

The similarities between ACO algorithms and GA are known and have been previously discussed in numerous studies. This research highlights two particular classes of GA known as Probabilistic Model Based Genetic Algorithms (PMBGA) and Estimation of Distribution Algorithms (EDA) as the most similar. These methods, developed mostly throughout the mid to late 90s, are extensions of the canonical GA and share many similarities with ACO algorithms since they also use probabilistic models to rank solution components. Most comparisons centre on differences between traditional ACO algorithms (such as Ant Systems) and a typical PMBGA, the Mining Based Incremental Learning (MBIL) algorithm. Of importance here though is the similarity of MACO to algorithms such as PBIL since MACO algorithms have a more population centric focus which may mean that MACO algorithms are now the closest ant-inspired algorithms to the PMBGA family. In this section the PBIL algorithm is introduced and then a subsequent comparison between PBIL and ACO is offered to determine similarities and points of difference.

The canonical Genetic Algorithm (GA) differs from other EC algorithms by its solution representation and combination of selection, recombination, mutation and replacement. Solutions were normally binary strings of a fixed length governed by the number of problem variables and precision required for each variable. While much of the early GA used binary strings, later work also includes

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

solutions encoded as arrays of floating point numbers (Real Value Genetic Algorithm – RVGA). The GA is usually initialized with a population of random solutions, although some 'seed' the initial population with solutions which are thought to be good.

After initialization the GA loops through the following processes until some termination criteria is met:

1. Selection – Select 'parent' solutions from the population using a fitness proportionate selection mechanism such as tournament selection or biased roulette wheel selection.
2. Recombination – Also known as crossover, components of the selected solutions (parents) are mixed to create new candidate solutions (children).
3. Mutation – Mutation involves targeting a candidate solution's individual solution components and perturbing them to introduce local variation.
4. Evaluation – The candidate solutions are evaluated using a fitness function to determine their utility.
5. Replacement – Candidate solutions are inserted into the population usually replacing older

or less fit solutions.

(d) ACO for Continuous Domains

To date, many ant-inspired approaches for application to CFO problems have been proposed. The Ant Colony Metaphor for Continuous Design Spaces was the first ant-inspired search proposed for CFO. That algorithm starts by placing a 'nest' somewhere in the n-dimensional search space, after which it projects a group of vectors (ants) into the search space around the nest. Over successive iterations it gradually adjusts the direction of these vectors towards promising areas of the search space. Other approaches include ACO for Continuous Domains with

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

Aggregation Pheromones Metaphor (APS), Continuous Interacting Ant Colony (CIAC) and Continuous ACO (CACO). Strictly speaking most of these approaches are ant-inspired but do not fit the criteria of ACO. ACO for Continuous Domains (ACOCD) is an extension of PACO. It maintains a population where every population member represents a single point in the n dimensional search space. Each population member is also assigned a weight **w** which is used for selection purposes. Solution construction is achieved by way of sampling each dimension in turn using a combination of the population's Probability Density Functions to resolve each new point. Newly constructed solutions are evaluated only if they fall within the bounds of the solution space and inserted into the population using the PACO Quality population management strategy, albeit with a large population size. Here, the minimum allowable population size is equal to the number of independent dimensions.

Algorithm 1:ACO for Continuous Domains in Mining (ACOCD)

1: Initialize history with uniform random solutions

2: **while** stopping criterion not met **do**

3: Rank population of visitor entries according to fitness (Fittest member l = 1)

4: **for** i = 1 to k **do**

5: Calculate selection probabilities for population according to equation 2.1

6: **end for**

7: **for** i = 1 to m **do**

8: Select s using biased random selection

9: **for** j = 1 to n **do**

10: s_i^j --- Gaussian distributed random value with mean $\mu_t^j = s_t^j$ and standard deviation μ_t^j according to Mining ACO

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 4, 2022

11: **end for**

12: Evaluate: s_i^t

13: **end for**

14: Select best new solution s0 from all new solutions ($s_0^t, \ldots\ldots\ldots\ldots\ldots\ldots, s_m^t$)

15: i**f** s_j^i is better than worst solution in population **then**

16: Replace worst solution in population with s0

17: **end if**

18: **end while**

OBSERVATION

This process was applied to two known uni-modal and multi-modal functions, a simple linear function with a global optima at the origin and Mining Function, in multiple dimensions. The application of this technique to known functions (Algorithm 1) is used to finding the problem of population entries in the mining methodology can be effort to assist in the correct interpretation of the results of the random sampling of the problem data from making on the cluster and genetic based mining algorithm. All solutions are normalized according to the possible dimensional boundaries and are plotted for comparison on the graph theory on ACO. In all dimensions are presented in order along with the x-axis. While the normalized position of a solution in this dimension is presented on the y-axis, with each solution's values connected with a line. The main aim to finding the problem on the minimize and maximize objectives values on the Ant Colony Optimization. The problem should be greatly solved on the Mining Methodology function like clustering and genetic algorithm. As can be seen from 5 cities on the MACO algorithms all produce good quality solutions as compared to the control algorithms. Of these MACO variants the MACO Quality algorithm produced the best quality results, although the difference is marginal when compared to the TSA

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

Datas and Cities Data in the Ant Colony Optimization Theory on the Mining Methodology.

REFERENCES

[1] J. Montgomery. Alternative representations for the job shop scheduling problem in ant colony optimsation. In M. Randall, H. A. Abbass, and J.Wiles, editors, Progress in Artifical Life, Proceedings of the 3rd Australian Conference on Artificial Life (ACAL07), number 4828 in LNAI, pages 1–12. Springer, 2007.

[2] A. Moraglio and R. Poli. Topological crossover for the permutation representation. In
GECCO'05, Washington, DC, USA, 2005.

[3] S. H. Pourtakdoust and H. Nobahari. An Extension of Ant Colony System to Continuous Optimization Problems. In M. Dorigo, M. Birattari, C. Blum, L. Gambardella, F. Mondada, and T. Stützle, editors, Proceedings of ANTS 2004 – Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence, volume 3172 of Lecture Notes in Computer Science, pages 294–301, Brussels, Belgium, September 2004. Springer Verlag.

[4] M. Randall. Maintaining explicit diversity within individual ant colonies. In Recent Advances in Artificial Life, chapter 17.World Scientific, 2005.

[5] H. Sato, H. E. Aguirre, and K. Tanaka. Local dominance including control of dominance area of solutions in MOEAs. In 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007), pages 310–317, 2007.

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

[6] T. Stützle and H. Hoos. TheMAX _MIN Ant System and local search for combinatorial optimization problems: Towards adaptive tools for global optimization. In Proceedings of the second Metaheuristics International Conference (MIC'97), Sophia-Antipolis, France, 1997.

[7] M. Manfrin, M. Birattari, T. St¨utzle, and M. Dorigo. Parallel ant colony optimization for the traveling salesman problem. Technical Report 2006-007, IRIDIA, March 2006.

[9] B. Scheuermann. Ant Colony Optimization on Runtime Reconfigurable Architectures. PhD thesis, Universit¨at Fredericiana zu Karlsruhe, 2005.

[10] R. K. Ursem. When sharing fails. In Proceedings of the Third Congress on Evolutionary Computation (CEC-2001), pages 873–879, 2001.

[11] K. Socha and M. Dorigo. Ant colony optimization for continuous domains. Technical Report 2005- 037, IRIDIA, December 2005.

[12] Y. Huang, S.M. Reddy, W.-T. Cheng, P. Reuter, N. Mukherjee, C.-C. Tsai, O. Samman, and Y. Zaidan, "Optimal Core Wrapper Width Selection and SOC Test Scheduling Based on 3-D Bin Packing Algorithm," *Proc. ITC*, 2002, pp. 74-82.

[13] F. Glover and G. Kochenberger, *Handbook of Metaheuristics*, Kluwer Academic Publishers, 2003.

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

[14] W. Ying, X. Jianying, Ant colony optimization for multicast routing, in *the 2000 IEEE*
Asia-Pacific Conference on Circuits and Systems, Tianjin, China

[15] B. Wu, Y. Zheng, S. Liu, Z. Shi, CSIM: a document clustering algorithm based on
swarm intelligence, in *Proceedings of the 2002 congress on Evolutionary Computation*,
Honolulu, USA.

[16] T. Stützle and M. Dorigo, A Short Convergence Proof for a Class of ACO Algorithms, *IEEE Transactions on Evolutionary Computation*, 6 (4), 2002 (in press).

[17] T. Stützle and M. Dorigo, Aco algorithms for the quadratic assignment problem, *New Ideas in Optimization*, McGraw-Hill, London, 1999, pp. 3--50.

[18] E. Siegel, B. Denby, S. Le Hégarat-Mascle, Application of ant colony optimizatio to
adaptive routing in a leo telecommunications satellite network, submitted *to IEEE Trasactions on Networks*, july 2000

[19] T. Stützle, A .Grün, S. Linke, M. Rüttger, A comparison of nature inspired heuristic on the traveling salesman problem, In Deb et al, editors, Proceedings of PPSN-VI, *Sixth*
International Conference on Parallel Problem Solving from Nature, volume 1917 of LNCS, pages 661-670, 2000

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 4, 2022

[20] T. Stützle, H. H. Hoos, MAX-MIN ant system, *Future Generation Computer Systems*, Vol. 16 (2000)

[21] V. Maniezzo and A. Carbonaro, A bionomic approach to the capacitated p-median
problem, *Future Generation Computer Systems* 16(8) (2000), 927--935.

[22] V. Maniezzo, A.Carbonaro (2001), Ant Colony Optimization: an overview, in C.Ribeiro (eds.) *Essays and Surveys in Metaheuristics*, Kluwer, pag.21-44.