Greedy Based Population Seeding Technique in Ant Colony Optimization Algorithm

KalaiPriyanThirugnanasambandam
Research Scholar
Department of CSE
Pondicherry University
kalaip27@gmail.com

Rajakumar Ramalingam
Research Scholar
Department of CSE
Pondicherry University
ramukshare@gmail.com

Vengattaraman Thirumal
Assistant Professor
Department of CSE,
Pondicherry University
vengattaraman.t@gmail.com

Sujatha Pothula
Assistant Professor
Department of CSE
Pondicherry University
spothula@gmail.com

Abstract—**In this paper, a new population seeding technique has been introduced in ACO algorithm. The method used for seeding technique is Greedy method. Along with the proposed method, a meta-heuristic algorithm Ant Colony Optimization is explained along with its applications on various problems. A detailed explanation of how the problem can fit into ACO algorithm is given. A tabular column of how the ACO has been modified to solve the respective problem has been given out.**

Keywords—Ant Colony Optimization, Pheromone, SPSP, TSP, ATSP, TSSP, FSSP, VRPTW, RDF chain Queries, RAP.

I. INTRODUCTION

Ant Colony Optimization [9-11] is one of the leading soft computing methods, which leads on discrete problem solving techniques. Discrete problems are the one which comes under combinatorial optimization. In discrete problem we have a finite set of inputs and need to optimize it in a way directed by its objective function. Ant colony optimization is inspired from the foraging behavior of ants. It works in cooperative manner with the help of pheromone referred as stigmergy (indirect communication). The paper [12, 13] gives knowledge of how ACO works and its application in various problems. It describes how the problem has been mapped with the ACO concept so that it could be easier to optimize using ACO.The paper has been organized as follows. Section 2 gives an introduction about ACO. Section 3 describes the problems which were optimized by ACO.

II. ANT COLONY OPTIMIZATION

Ant colony optimization is a constructive method in finding the optimal solutions. It constructs the solutions in an iterative manner. Two forms are used in this ACO. Pheromone table updation and probability calculation are the two phases that makes ACO a high meta-heuristic optimization algorithm.

A. Pheromone Table Updation

The pheromone table gets updated each time when it chooses a node based on the probability. This change in pheromone table leads to find an optimal solution for a given problem.

$$\tau_{ij} = (1-\rho).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

B. Probability Calculation

Probability is the decision variable which promotes a given node to participate in the solution set. This probability calculation includes visibility, pheromone rate, alpha and beta values for tuning the heuristic involvement over ACO. The calculation of pheromone rate is as follows. The detailed description about ACO Algorithm is portrayed in fig.1.

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l\in\aleph_i^k}[\tau_{il}(t)]^{\alpha}.[\eta_{il}]^{\beta}}$$

C. Pseudo-code of Ant Colony Optimization

Input: An Instance *P* of ACO problem model *P = (S, f, Ω)*
InitializePheromoneValues(*T*)
S_{bs} ← NULL
while termination conditions not met **do**
§$_{iter}$ ← Ø
For j = 1,………, n_a **do**
S ←ConstructSolution(*T*)
If S is a valid solution **then**
S ←LocalSearch(S)
{optional}
If (f(s) < f(S_{bs})) or (S_{bs}=NULL)
then S_{bs} ← S
§$_{iter}$ ← §$_{iter}$U {S}
end if
end for
ApplyPheromoneUpdate(T, §$_{iter,}$ S_{bs})
End while
Output: The best-so-far solution S_{bs}

Figure 1: ACO Algorithm

III. ACO AND OTHER APPLICATIONS

A. ACO on RDF Chain Queries

Basically there were a lot of computational research have been done in the field of optimization on RDF query search. The main cause the optimization is to reduce the search time and the collected queries should be sorted according to their relevance to the given query.

Ant colony optimization for RDF chain queries for decision support, Alexander Hogenboom, et al. [1] have suggested a new way to improve the data accessing speed faster than before. The previous algorithms did not give reasonable results and also the process is static in a dynamic environment where the data changes dynamically. This would be one of the major problems in the most dynamic scenario (Web) where the datum keeps updated or upgraded. The proposed work resolves scheduling problem and sequential ordering. And the way he achieved the aim is by applying the ACO foraging tech of ant the path has been developed in a shorter format and whenever the data changes its source address the ACO adapts accordingly. With this kind of idea the run time changes can also be adapted according to the situation and this is one of the most essential things in web search. To achieve the result, parameters as the cost of each joins and the number of joins were taken into account for resulting a particular query. The way author have transformed the ACO into the suggested system is by taking the joins as the vertices of a graph where the ant travels along the path from vertex to another vertex and the path the ant chosen will be the flow of join and the cost will be calculated in such a way that in which path that the ant travels will be taken into account and the problem will be resolved in such a manner. The pros and cons of this paper: **Pros:** It handles the change over the data source and latency differences. **Cons:** For more join operations a lot of combinations of possible solutions are computed. (Just for 20 join operations 720,218 possible candidate solutions are formed). The final set of computational results have also been made and these results has been compared with the existing 2 phase optimization approach and Genetic algorithm approach and the results shown with drastic improvement.

B. ACO in Software Project Scheduling Problem

In the field of software development there were many problems that are facing on how to estimate the duration of the software process to product delivery. One of those problems is Software Project Scheduling Problem. This is also being solved by the Ant Colony Optimization Technique. This is one of the emerging problems in the existing informative world. This given approach has solved the problems in SPSP.

Jing Xiao, et al. [2], have suggested a way for solving the software project scheduling problems using the ant colony optimization (SPSP). The need of this technique is to make the project scheduling crispier by adding more parameters and the parameters were chosen in such a manner that each and every parameter gives the additional weightage to the problem. Using the suggested idea of this paper we can reduce the project scheduling. This idea has been suggested since the previous work does not give accurate timing because, the previous works on this SPSP does not include the cost associated with the employees. The employee's several skills also taken into account in this SPSP solving using ACO. This will give the project scheduling results crispier than the existing algorithms which didn't concentrate much on these parameters. It has been done by considering the employee dedication, Number of skills present in each employee, utilization of employee skills completely during the regular working hours. The parameters taken into account for solving the SPSP using ACO are the number of tasks, skills required number of employees, Vertex set, arc set, workload per task, etc. The technique handled here would be Ant Colony System. The transformation of the SPSP problem according to the suggested idea of solving SPSP using ACO is the employee's dedication is taken in format of vertices with edges. The dedication of the employees is scattered in the form of vertices of a graph. The employees are placed as columns. - The ant travel in such a way it chooses one set of dedication from each employee and finally finishes the task. This paper has taken the Genetic Algorithm as the comparison data set. The pros and cons of this paper, Pros: The Results are accurate since we didn't leave any of the parameter and because of giving weight-age for the parameters which are in need. Cons: Only the employee dedication has been calculated on the whole Project Scheduling. For this alone they are taking a lot of effort. When comes on calculating the whole SPSP the computational time would be more. Finally, the algorithm have been implemented and the results have been compared with existing Genetic Algorithm approach and the results shows that this Ant colony System produce better results on SPSP problem than GA.

C. ACO on Traffic Signal Timing Problem

Traffic Signal timing problem would be considered as the one of the most depressed state of problem which makes a vast damage in the fast this world. The timing that a person waits in a signal, the number of stops they used to stop the vehicle at peak time were all been the problems in Traffic Signal Timing.

Jiajia He, et al. [3], suggested an idea for optimizing the traffic signal timing using Ant Colony Optimization. The need of this technique is to handle the traffic flow in rush hours and free hours. This idea has been suggested because, the time delay is more when there is less flow of traffic and the number of stops also more and the traffic capacity is less or constant even in rush hours which can be resolved with some heuristics and with some mathematical calculations. It has been done by optimizing the signal timing, by reducing the time delay and number of stops in free hours, and by improving the traffic capacity in Rush hours. The parameters taken in this paper are saturation flow, time delay, number of stops, traffic capacity and the weighting coefficients of all these parameters. These parameters will result the solutions in a very sharp result on how to handle the 3 kind of situations as mentioned above. The transformation of the Traffic signal problem which needs to be solved using ACO to the suggested idea is the ants are taken as vehicle that crosses the signal. The pheromone that it secrets along the path will be considered as the path or optimal solution to handle the traffic signal at various situations like What to do when there is low traffic? And how to optimize the time at Large flow of traffic? A new formula has been derived along with the coefficients of those parameters and also the parameters are taken into account. Implement the formula and the simulation with some numerical values has been done. The technique they have used for handling all the constraints is Ant Colony Algorithm. The pros and cons of this paper are Pros: The evaporation of pheromone has been considered as one of the idea and the iteration has been done in some other paths which may lead to optimal solution. Cons: If the pheromone updating has not been done in optimal solution which convergence to optimality that drops down may lead to non-optimal solution. The implementation has been made with a simulator and the result have been compared with the existing Genetic Algorithm Technique and also the Webster technology and it is proved that the Ant Colony Algorithm provide best results than others.

D. ACO on Vehicular Routing Problem

Vehicular Routing Problem is a kind of thing where the problem occurs in a scenario like serving to the customers with the same number of vehicles as customers. In this paper they have solved the problems related to vehicular routing problem using Improved Ant Colony Optimization which is also called Hybrid Ant Colony Optimization.

Iulei Ding, et al. [4] have suggested a new idea for solving the vehicular routing problem with time windows using an improved ant colony optimization (VRPTW). The need of this technique is to improve the vehicular routing problem which needs an appropriate way to travel via the vehicles. This idea has been suggested because the previous algorithms often get trapped in local optima and premature convergence and low search speed which makes the problem not to meet the maximum iteration number and produce local optimum result as best result. It has been done by adjusting the pheromone to avoid local optima, the disaster operator to search the solution space completely, to minimize the candidate list in order to reduce the computational time and finally the use of Solving Algorithm and Interchange Algorithm in order to improve the convergence speed. The parameters that were taken into account to solve the VRPTW are the number of customers, density of pheromones, density of visibility, and by taking the candidate list as $1/4^{th}$ of the total customers. These parameters will help the proposed ACO technique to result the given problem in an optimistic manner. The VRPTW problem which needs to be solved using improved ACO has been transformed to the suggested idea (i.e.) representing the problem in order to solve it with the help of artificial ants, each constraint in VRPTW is encoded with a formula, each parameter is initialized with a numerical value and the evaluation is made using a format, ants have been taken as vehicles, Customers have been taken as Nodes, and the distance between 2 customers will be considered as the travelling time from one city to other. The given problem has been implemented using the Hybrid Ant Colony Technique. The pros and cons of this paper, Pros: The Pheromone density value set dynamically avoids the local optima. Cons: Using these many algorithms increase the computational time heavily which gives very minimal reduction of errors compared with previous algorithm. Finally, the result has been made and it has been compared with the existing techniques and the proposed technique yields better result when compared with the other things. The modifications and adaptions for solving problems on ACO is tabulated in Table I.

TABLE I
Modifications on ACO – Adaptation for solving the problems

Authors	Problem	ACO variant	Modification
Alexander Hogenboom, et. al. [1]	RDF Chain Queries	Ant Colony Optimization	No modifications were incorporated
Jing Xiao et. al. [2]	Software Project Scheduling Problem	Ant Colony System for Software Project Scheduling (ACS-SPSP)	Incorporates 6 heuristics information in ACO. It diverts the ACO from choosing the nodes not only by the pheromone table but also using the proposed heuristics values
Jiajia He et. al. [3]	Traffic Signal Timing Optimization	Ant Colony Optimization	No modifications were incorporated
Iulei Ding et. al. [4]	Vehicular Routing Problem with time windows	Improved Ant Colony Optimization	Modified the original ACO by imposing dynamic evaporation rate ($1-\rho$)

IV. PROPOSED METHODOLOGY

Although the existing conventional ACO and its variants performs well in multiple domains there exists still an unstable behavior of working. For example, if the initial population leads all the solutions with worst fitness function then at the first iteration of ACO the pheromone rate will be updated on those edges. If this may lead to next generation then the possibility of getting a global optimal solution is uncertain.

In order to eradicate this uncertainty, the proposed methodology incorporates a seeding method along with the random population generation.

The proposed methodology incorporates a greedy technique to generate a single solution in population initialization. A solution for initial population using greedy technique uses distance matrix to build it. Greedy algorithm works with the form of choosing a best node based on problem dependent criteria. On incorporating this kind of solution among the random population, a solution which satisfies a firm fitness value can be obtained. And also this method will be useful when considering edge selection during the next iteration.

Working methodology of greedy algorithm has been described below in fig.2.

A. Greedy Algorithm

Optimization algorithms often go through a sequence of steps to determine optimal solutions. Greedy algorithm has been developed in order to eradicate the exhaust search from a given search space. To be practical the greedy algorithm will not give a best solution for large scale problems. Still this algorithm is often used in wide range of domains for time saving process. The general greedy technique chooses the best edge from the given node and further proceeds from the next node in same process.

Greedy Algorithm

Greedy-selection (S, Dis[])
$N \leftarrow length(S)$
$A \leftarrow S_1$
$j \leftarrow 1$
$fori$ =2 to N-1
$if\left(D\left[A_j, S_i\right] > D\left[A_j, S_{i+1}\right]\right)$
$A_{j+1} \leftarrow S_{i+1}$
end

Figure 2: Greedy Algorithm

B. Greedy Algorithm based Population Seeding Technique in ACO

Based on the above greedy algorithm the seeding technique of ant colony optimization technique can be improvised with the help of greedy search algorithm. Greedy based population seeding technique in ant colony optimization is shown in fig.3.

Greedy Algorithm in ACO

```
Population Initialization
    Define Random Population Rate = 99%
    Greedy Method Population = 1%
    for i=1 to 99%
      randperm(N)
    end
    for I = 100
      Greedy Method using above alg. Fig(2)
    end
```

Figure 3: Greedy Algorithmin ACO

The reason for not having more than 1 greedy based solution in initial population is that because a greedy search may result in similar solution when the starting and ending nodes are same. A classical example of those kind of problem is Travelling Saleman Problem (TSP).

V. CONCLUSION

This paper presents an overview of ACO along with a proposed population seeding technique called Greedy based population seeding technique. It started with an introduction about the working then a pseudo code of ACO has been given. Later on a detailed explanation of how ACO adapt with the given problem has been discussed in detail. Finally, a detailed description of what modifications made in the original ACO algorithm has been given. This paper will lead the researchers to apply other domain problems on ACO.

REFERENCES

[1] Alexander Hogenboom, Flavius Frasincar, UzayKaymak, “Ant colony optimization for RDF chain queries for decision support” Proc. Expert Systems with Applications of Science Direct.

[2] Jing Xiao, Xian-Ting Ao, Yong Tang, “Solving software project scheduling problems with ant colony optimization” Proc. Computers & Operations Research 40 Science Direct, pp 33–46, 2013.

[3] Jiajia He, ZaienHou, “Ant colony algorithm for traffic signal timing optimization” Proc. Advances in Engineering Software 43 Science Direct, pp. 14–18, 2012.

[4] iulei Ding, Xiangpei Hu, Lijun Sun, Yunzeng Wang, “An improved ant colony optimization and its application to vehicle routing problem with time windows” Neuro computing Science Direct.

[5] Jie Bai, Gen-Ke Yang, Yu-Wang Chen, Li-Sheng Hua, Chang-Chun Pan, “A model induced max-min ant colony optimization for asymmetric traveling salesman problem”, Applied Soft Computing, Science Direct, 2012.

[6] Gaifang Dong, William W. Guo, Kevin Tickle, “Solving the traveling salesman problem using cooperative genetic ant systems”, Expert Systems with Applications 39, Elsevier Journal, v2012. pp. 5006–5011.

[7] R.F. Tavares Neto, M. GodinhoFilho, “An ant colony optimization approach to a permutational flowshop scheduling problem with outsourcing allowed”, Computers & Operations Research 38, Elsevier Journal, 2011.,pp. 1286–1293,

[8] Zhengxing Huang, Xuang Lu, HuilongDuan,” A Task Operation Model for Resource Allocation Optimization in Business Process Management”, IEEE Transactions on Systems, man, and cybernetics- part a: systems and humans, vol 42, no. 5, september 2012.

[9]Dorigo, M., & Gambardella, L. M, ”Ant colony system: a cooperative learning approach to the traveling salesman problem.” IEEE Transactions on Evolutionary Computation, 1, 1997, 53–66.

[10] Stützle, T., and M. Dorigo. "The ant colony optimization metaheuristic: Algorithms, applications, and advances." Handbook of metaheuristics (2002): 251-285.

[11] Parpinelli, Rafael S., Heitor S. Lopes, and Alex Alves Freitas. "Data mining with an ant colony optimization algorithm." IEEE transactions on evolutionary computation 6.4 (2002): 321-332.

[12] Liang, Yun-Chia, and Alice E. Smith. "An ant colony optimization algorithm for the redundancy allocation problem (RAP)." IEEE Transactions on reliability 53.3 (2004): 417-423.

[13] Al Salami, Nada MA. "Ant colony optimization algorithm." UbiCC Journal 4.3 (2009): 823-826.