A Hybrid Ant Colony Optimization Algorithm using MapReduce for Arc Routing Problem

Gajanan Aochar
M.E. Student
Dr. D Y Patil School of Engg and Tech
Savitribai Phule Pune University, Pune

Roshni Ade
Assistant Professor
Dr. D Y Patil School of Engg and Tech
Savitribai Phule Pune University, Pune

ABSTRACT
Ant colony optimisation (ACO) could be a comparatively new random heuristic approach for determination optimisation issues. Furthermore, This paper extend these implementations with two local search methods and we compare two heuristics that guide the HACO algorithms. However, relatively few results on the runtime analysis of the ACO on the TSP are available. Moreover, we experiment with two different pheromone update strategies. In order to demonstrate this we present an ACO implementation for the travelling salesman problem it requires a larger number of ants and iterations which consume more time. The influence of the parameters controlling the relative importance of pheromone trail versus visibility is also analyzed, and their choice is shown to have an impact on the expected runtime. The main application of the ACO algorithm lies in the field of combinatorial optimization, and the traveling salesman problem (TSP) is the first benchmark problem to which the HACO algorithm has been applied.

Keywords
Hybrid Ant colony optimization (HACO), Ant colony optimization(ACO), Capacitated vehicle routing problem (CVRP), Multi-depot vehicle routing problem (MDVRP), combinatorial optimization, mimetic algorithms

1. INTRODUCTION
This downside naturally arises in several industrial sectors, i.e., the routing of street sweepers [1], snow ploys [2], manage refuse assortment vehicles [3] or gritting (salting) trucks [4], the look of mail delivery [5] or school bus service [6], and also the examination of electrical power lines [7], gas pipelines [8], or oil pipelines [9]. In this paper, a replacement ECARP is planned that considers these aspects that we tend to view as essential with regard to sensible requirements: With the event of service-oriented governments [11], every office is accountable to produce a timely service to the general public. At identical time, a robust competition necessitates a versatile and dynamic company model that facilitates agile delivery. what is more, the service design (e.g., infrastructure) ought to be optimized to satisfy the changing requirements of the purchasers. Whereas the normal service mission is "can do it" (i.e.,give the optimum service among the existing service architecture), the present service mission is "do it best" (i.e., give the optimum service among Associate in Nursing optimized service architecture). In terms of the ECARP, both the service programming (a set of auto routes) and therefore the service architecture (a configuration of depots and vehicles) ought to be optimized at the same time to enhance the standard of service. For this reason, each the utmost total service time and therefore the fixed investment price (which is applied to optimize the service architecture) square measure thought of during this paper. Ant colony improvement (ACO) that has gained quality in recent years attributable to varied success stories [12]–[15]. For this reason, a unique hybrid ACO algorithmic program (HACOA) is planned to tackle the new and extended version of ECARP. The remainder of this paper is organized as follows. Section II reviews some recent work associated with the CARP. Section III formulates this ECARP. The HACOA is made public in Section IV. Section V analyzes the performance of the algorithm, and Section VI offers some final remarks as well as directions for future work.

2. LITERATURE SURVEY
Here are some papers that will be referred and the technique used in that paper or discussion held in that paper will be used for our knowledge. **[4]**M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the travelling salesman problem this paper introduces the premire colony system (ACS), a distributed formula that's applied to the voice problem (TSP). In the ACS, a collection of cooperating agents referred to as ants cooperate to seek out smart solutions to TSP's. Ants collaborate victimization associate indirect kind of communication mediate by a secretion they deposit on the perimeters of the TSP graph whereas building solutions. We study the ACS by running experiments to know its operation. The results show that the ACS outperforms different nature-inspired algorithms like simulated hardening and organic process computation, and that we conclude comparison ACS-3-opt, a version of the ACS increased with an area search procedure, to a number of the simplest playing algorithms for interchangeable and asymmetric TSP's.

[4]M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," An analogy with the method hymenopteran colonies perform has prompt the definition of a brand new process paradigm, that we tend to decision hymenopteran system (AS). we tend to propose it as a viable new approach to random combinatorial improvement. the most characteristics of this model ar feedback, distributed computation, and also the use of a constructive greedy heuristic. feedback accounts for speedy discovery of fine solutions, distributed computation avoids premature convergence, and also the greedy heuristic helps realize acceptable solutions within the early stages of the search method. we tend to apply the projected methodology to the classical roadman drawback (TSP), and report simulation results. we tend to additionally discuss parameter choice and also the early setups of the model, and compare it with tabu search and simulated tempering victimization TSP.To demonstrate the lustiness of the approach, we tend to show however the hymenopteran system (AS) is applied to different improvement issues just like the uneven roadman, the quadratic assignment and also the job-shop programming. Finally we tend to discuss the salient characteristics-global

organisation revision, distributed communication and probabilistic transitions of the AS.

[3] C. Blum, "Ant colony optimization: Introduction and recent trends," Ant colony improvement may be a technique for improvement that was introduced within the early 1990's. The inspiring supply of hymenopterous insect colony improvement is that the forage behavior of real hymenopterous insect colonies. This behavior is exploited in artificial hymenopterous insect colonies for the search of approximate solutions to distinct improvement issues, to continuous improvement issues, and to special issues in telecommunications, like routing and cargo equalization. First, we have a tendency to contend with the biological inspiration of hymenopterous insect colony improvement algorithms. we have a tendency to show however this biological inspiration may be transferred into AN rule for distinct improvement. Then, we outline ant colony improvement in additional general terms within the context of distinct improvement, and gift a number of the days best-performing hymenopterous insect colony improvement variants. When summarizing some necessary theoretical results, we have a tendency to demonstrate however hymenopterous insect colony optimization may be applied to continuous improvement issues. Finally, we offer samples of a motivating recent analysis direction: The interbreeding with a lot of classical techniques from computing and research. 2005 Elsevier B.V. All rights reserved

3. IMPLEMENTATION DETAILS

***A)* Architecture Optimization Content :** architecture is to decide the number of depots required and their positions. MapReduce, which is usually designed for data-intensive problem, provides a two-phase (map and reduce) divide-and conquer processing strategy. The principle of MapReduce framework illustrates that in the map phase no data exchange between map tasks can be performed. Reduce is the only way to aggregate outputs from different map tasks. This restriction ensures the simplicity and reliability of the model. However, at the same time, it also requires developers to design the partition and aggregation of the problem carefully.

1) Compute the Least Number of Depots: Since all required arcs should be serviced within the maximal service time T and vehicle routes must start and end at the same depot, the least number of depots is decided by the maximum travel distance of trucks, $\beta = S \times T$. The distance between node $i(1 \le i \le NV)$ and the depots can be defined as

$$dis(i) = \min_{1 \le j \le N_{V,d_j=1}} SPL(i,j)$$

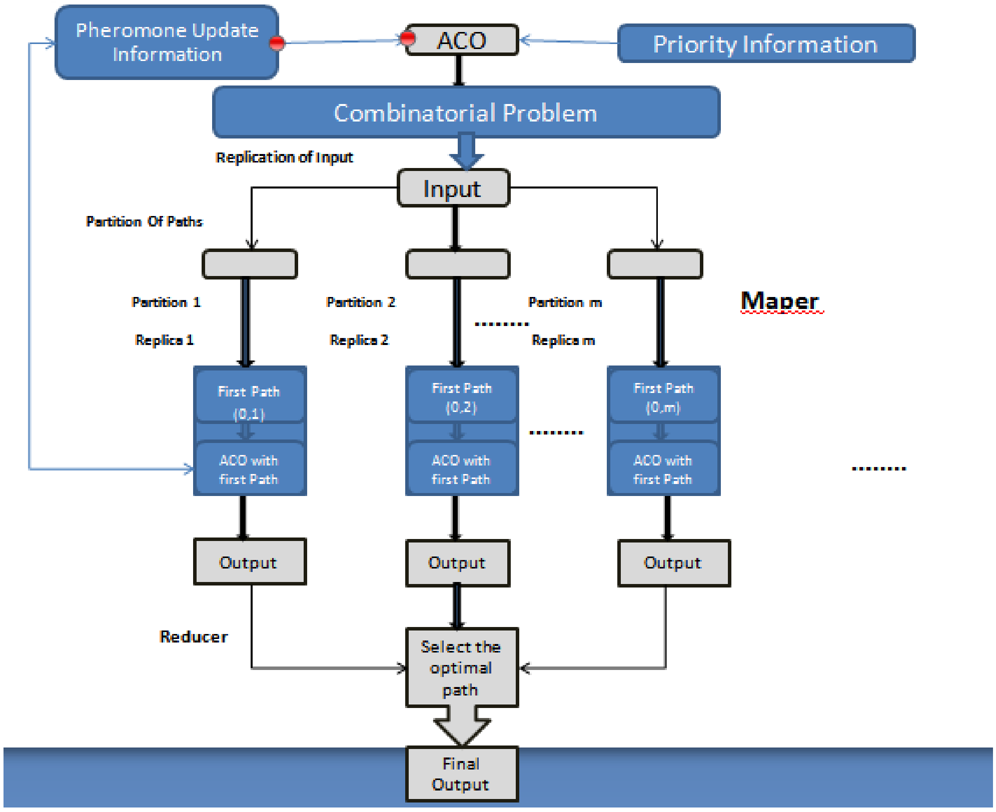

Figure1 System Architecture

2) Heuristic info Initialization: The depot choice info, used to improve the standard of the design improvement, is the accumulative information (refined iteratively) of assignment a given node to be a depot. associate array DSI with size Sagebrush State is employed for the depot choice info, where DSI(i) denotes the amount of times node i used to be selected as a depot among the near-optimal solutions obtained.

3) Evaluate the Fitness of Each Depot Set: The fitness of each set of depots is computed using the Extended Random Path- Scanning (ERPS) heuristic [40]. When constructing a tour,ERPS appends the most promising arc to the existing sequence of arcs until the capacity Q or service time T is exhausted.

B) Information Initialization: In ACO, the synthetic ants can deposit secretions on the paths traversed. kind of like the secretion definition delineated earlier during this paper, this extends the construct of secretion deposition, creating use of arc cluster info and arc priority information. In HACOA, the synthetic ants can construct feasible solutions supported this info, reinforcing (via deposition) the knowledge on the travelled methods. Arc Cluster Information: This info wont to assign required arcs to existing clusters is extracted from the simplest solutions found throughout the search.

C) Dynamic Parameter Adjustment ***:*** In order to boost the performance of HACOA and decrease the impact of initial parameter decisions on the results obtained, parameters ar dynamically adjusted per the performance of various parameter combos. If the most effective solution found to date improves when associate degree iteration, then this iteration are going to be referred to as a prospering iteration. the amount of successful iterations beneath a given parameter combination will be thought to be its improvement performance. within the data formatting phase, several parameter combos ar generated following the orthogonal style [57], and therefore the improvement performance of each parameter combination is initialized together. In the beginning of every iteration, one parameter combination are going to be selected willy-nilly per its improvement performance.

3.1 System Features:

1) Architecture optimization: It obtains near-optimal sets of depots (the range and position of depots are set in this step).

2) Data initialization: It initializes the data (e.g., arc cluster or arc priority information).

3) Dynamic parameter adjustment: It dynamically selects a specific combination of parameters consistent with their past optimization performances.

4) Possible answer construction: It constructs a bunch of solutions exploitation ACO. This step makes use of the heuristic information obtained from the most effective solutions found to date.

5) Data updating: the data is updated exploitation information extracted from the near-optimal solutions obtained. Steps 3)–5) are perennial iteratively till a stopping criterion is satisfied. within the following, we have a tendency to define every part well.

Algorithm Used:

Existing Algorithms:

Iterative heuristic algorithm-IHA

Algorithm 1:

Generate initial solution S1

BestSolution=S1

While Termination Criterion is not fulfilled **do**

S2=ConstructionPhase(S1)

If Solution S2 fulfills the acceptance criterion **then**

S1=S2

else

S1=BestSolution

Apply perturbation in solution S1

If S2 has better (or equal) width than BestSolution **then**

BestSolution=S2

return BestSolution

Algorithm 2:

Begin

Initialization: Set parameters, initialize pheromone values, choose randomly an initial ant solution x;

while (termination-condition does not hold) **do**

Construct an ant solution y;

Selection: If f(y) < f(x),x:=y;

Update the pheromone values with respect to x;

Endwhile

End

4. MATHEMATICAL MODEL

For Filtering Rules:

1) Input

Filtering Rules are customizable by the user. User can have authority to decide what contents should be blocked or displayed on his wall by using Filtering rules. For specify a Filtering rules user profile as well as user social relationship will be considered.

FR= {Pheromone, Exploitation, Trail, Inverse, Evaporation, Exponent, Elite}

FR is dependent on following factors

- Pheromone
- Exploitation
- Trail
- Inverse
- Evaporation
- Exponent
- Elite

Pherormone: chemical produce by ant

Exploit: Probability for exploiting best edge

Initial: Starting point

Epoch: current number of epochs

2) Process

FM= {Pheromone, Exploitation, Trail, Inverse, Evaporation, Exponent, Elite }

Here, most existing parallelized ACO works lie in implementing ACO with traditional parallel programming models. M. Manfred et al. implemented a parallel ACO to the TSP in a multi-machine environment with MPI. Basically, there are two information exchange strategies in parallel ACO: synchronous and asynchronous. In the synchronous strategy, all execution units run synchronously, and the pheromone is updated after all units complete one iteration. In the asynchronous model, each execution unit s run independently and the wait-free pheromone update is performed. Craus and Rudeanu also implemented a MPI-based parallel ACO algorithm with a one-master-multi-slave architecture. In, checkpoint is used to update pheromone asynchronously. Compared with traditional models like MPI, parallelization with MapReduce has less attention in the academia. Huang and Lin implemented a genetic algorithm with MapReduce to the job scheduling problem and has a reasonable output [9]. The approach in [9] requires multiple MapReduce iterations, which drop the performance of the algorithm. The same problem also exists in [5], which implements parallel ACO with MapReduce to the TSP. Experimental result in [5] illustrates that the one-iteration approach has a far better performance than the multi-iteration approach

3) Output
PFM= {BestSolution}

- BestSolution get from number of epoch.

In general, more than a filtering rule can apply to the same user.

4.1 Result of Practical Work

4.1.1 Performance Metrics

The performance of our proposed topology are evaluated by various parameters described below : Precision: It defines refinement in a measurement of the result.

Recall: it is used to find out total recall of true values from the result.

Accuracy: It shows accuracy of sending message between HACOA algorithm without using MapReduce Framework and with using MapReduce Framework.

4.1.2 Graphical Analysis

Precision:
It will show and give the comparative precision result between Hybrid Ant Colony Optimization Algorithm using MapReduce Framework and without using MapReduce Framework in terms of time in milliseconds.

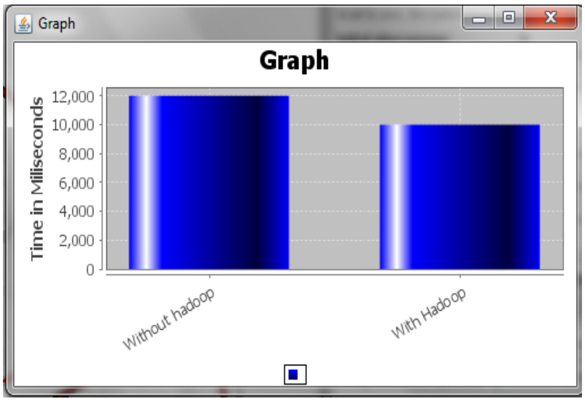

Figure 2. Time performance with and without Hadoop

5. CONCLUSION

The main contributions of this paper could also be summarized as follows: 1) actuated by sensible necessities, AN ECARP is proposed that considers each the overall service time and stuck investment prices. 2) A HACOA is planned to cope with the increased quality of the ECARP. Domain-specific data, arc cluster data and arc priority data, is continuously extracted from the solutions obtained throughout the search and is employed to guide the next improvement process. The dynamic parameter adjustment perceptibly improves the performance of the rule and reduces the sensitivity of initial parameter decisions. Finally, an area improvement step supported two-opt considerably improves the performance of HACOA. 3) In depth experimental study on eighty seven downside instances has been meted out to guage the performance of HACOA. The results indicate the prevalence of HACOA over other problem-specific heuristics. Future research both the service architecture and service scheduling should be optimized synchronously using metaheuristic.

6. REFERENCES

[1] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 53–66, Apr. 1997.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 26, no. 1, pp. 29–41, Feb. 1996.

[3] C. Blum, “Ant colony optimization: Introduction and recent trends,” Phys. Life Rev., vol. 2, no. 4, pp. 353–373, Dec. 2005.

[4] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey,” Theor. Comput. Sci., vol. 344, no. 2/3, pp. 243–278, Nov. 2005.

[5] B. Bullnheimer, R. Hartl, and C. Strauss, “A new rank-based version of the Ant System: A computational study,” Central Eur. J. Opera. Res. Econ., vol. 7, no. 1, pp. 25–38, 1999.

[6] T. Stutzle and H. H. Hoos, “Max-min ant system,” Future Gener. Comput. Syst., vol. 16, no. 8, pp. 889–914, 2000.

[7] C. Blum and M. Dorigo, “The hyper-cube framework for ant colony optimization,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 34, no. 2, pp. 1161–1172, Apr. 2004.

[8] Y. W. Leung and F. Wang, “An orthogonal genetic algorithm with quantization for global numerical optimisation,” IEEE Trans. Evol. Comput., vol. 5, no. 1, pp. 41–53, Feb. 2001.

[9] S. Lin and B. W. Kemighan, “An effective heuristic algorithm for the traveling salesman problem,” Oper. Res., vol. 21, no. 2, pp. 498–516, Mar./Apr. 1973.

[10] M. Kiuchi, Y. Shinano, R. Hirabayashi, and Y. Saruwatari, “An exact algorithm for the capacitated arc routing problem using parallel branch and bound method,” in Proc. Abstr. Spring Nat. Conf. Oper. Res. Soc. Jpn., 1995, pp. 28–29.

[11] J. M. Belenguer and E. Benavent, “A cutting plane algorithm for the capacitated arc routing problem,” Comput. Oper. Res., vol. 30, no. 5, pp. 705–728, Apr. 2003.

[12] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” J. Mach. Learn. Res., vol. 7, pp. 1–30, 2006.

[13] J. He, X. Yao, and J. Li, “A comparative study of three evolutionary algorithms incorporating different amount of domain knowledge for node covering problems,” IEEE Trans. System.

[14] H. G. Beyer, H. P. Schwefel, and I. Wegener, "How to analyze evolutionary algorithms," Theor. Comput. Sci., vol. 287, no. 1, pp. 101–130,2002.

[15] C. Witt, "Worst-case and average-case approximations by simple randomized search heuristic," in Proc. 22nd Annu. Symp. Theor. Aspects Comput. Sci. (STACS), LNCS vol. 3404. Stuttgart, Germany, Feb. 2005, pp. 44–56.

[16] D. Sudholt, "Crossover is provably essential for the Ising model on trees," in Proc. Genetic Evol. Comput. Conf. (GECCO '05), Washington D.C., Jun. 2005, pp. 1161–1167.

[17] J. He, X. Yao, and J. Li, "A comparative study of three evolutionary algorithms incorporating different amount of domain knowledge for node covering problems," IEEE Trans. Syst., Man Cybern., Part C, vol. 35, no. 2, pp. 266–271, 2005.

[18] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt, "On improving approximate solutions by evolutionary algorithms," in Proc. Congr. Evol. Comput., Piscataway, NJ: IEEE Press, 2007, pp. 2614–2621.

[19] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt, "Approximating covering problems by randomized search heuristics using multiobjective models," in Proc. Genetic Evol. Comput. Conf. (GECCO), London, U.K., Jul. 2007, pp. 797–804.

[20] F. Neumann and C. Witt, "Runtime analysis of a simple ant colony optimization algorithm," in Proc. 17th Int. Symp. Algorithms Comput. (ISAAC), LNCS vol. 4288. Kolkata, India, Berlin, Germany: Springer- Verlag, Dec. 2006, pp. 618–627.

[21] W. J. Gutjahr, "First steps to the runtime complexity analysis of ant colony optimization," Comput. Oper. Res., vol. 35, no. 9, pp. 2711–2727, 2008.

[22] B. Doerr and D. Johannsen, "Refined runtime analysis of a basic ant colony optimization algorithm," in Proc. IEEE Congr. Evol. Comput, Piscataway, NJ: IEEE Press, 2007, pp. 501–507.

[23] B. Doerr, F. Neumann, D. Sudholt, and C. Witt, "On the runtime analysis of the 1-ANT ACO algorithm," in Proc. Genetic Evol. Comput. Conf. (GECCO), London, U.K.: ACM, 2007, pp. 33–40.

[24] W. J. Gutjahr and G. Sebastiani, "Runtime analysis of ant colony optimization with best-so-far reinforcement," Methodology Comput. Appl. Probab., vol. 10, no. 3, pp. 409–433, 2008.

[25] F. Neumann, D. Sudholt, and C. Witt, "Analysis of different MMAS ACO algorithms on unimodal functions and plateaus," Swarm Intell., vol. 3, no. 1, pp. 35–68, 2009.