Optimizing Routes for Medicine Distribution Using Team Ant Colony System

Renan Costa Alencar[1[0000−0003−0028−558X]] and
Carmelo J. A. Bastos-Filho[1[0000−0002−0924−5341]]

Polytechnic University of Pernambuco, University of Pernambuco, Recife, Brazil
{rca2,carmelofilho}@ecomp.poli.br

Abstract. Distributing medicine and hospital supplies is considered a complex and hard problem to solve. Within hospital logistics, that problem is associated with the Multiple Traveling Salesman Problem (MTSP) and the Knapsack Problem (KP). MTSP problems aim to minimize the total displacement of the salesmen, with a constraint that the paths must begin and end in the depot and all intermediate nodes should be visited only once. In the order hand, KP instances aim to maximize the capacity of a sack through previously selected objects. Those problems can be solved using the Team Ant Colony Optimization (TACO), a variation of the Ant Colony System (ACS) algorithm, based on ant colony behavior. In preliminary results, the above approach was promising for two scenarios: minimizing the largest route, uniformly allocating the workload to all salesmen, and minimizing the total cost of routes, i.e., the sum of all route costs of individual salesmen. However, these objectives are concurrent. This work proposes the use of swarm optimization algorithms as global optimizers to obtain better results than those from previous findings. TACO algorithm uses those algorithms as global optimizers to adjust its parameters and consequently to improve the results for those objectives already mentioned. The results for the use of global optimizers were promising for the optimization of the objectives tackled by TACO.

Keywords: Multiple Traveling Salesman Problem, · Swarm Intelligence · Hospital Logistics.

1 Introduction

The Travelling Salesman Problem (TSP) and the Knapsack Problem (KP) are two of the most studied combinatorial optimization problems so far. They stand out from other combinatorial problems for being easy to design but hard to solve. Hence, the TSP and KP belong to the set of NP-complete problems. Although they are tough to be solved even separately, there are some real-world problems that can be mapped as a combination of them, resulting in a complex task.

There is a TSP variation, which multiple salesmen engage in building a solution, the so-called Multiple Salesmen Problem (MTSP). According to Bektas [3], there is a variety of real-life problems, which are considered as MTSP, e.g.

Routing Vehicle Problems (PRVs) with solution constraints. As stated in the MTSP definition, there are $m > 1$ salesmen located initially in the same city which defines the depot. The other cities of the instance are defined as intermediate nodes. The MTSP aims to minimize the sum of route lengths with a constraint, which every route should begin and end at the depot node, and all intermediate nodes should visit only once. Besides, the MTSP problem has another constraint which there should be at least an intermediate node but the depot for each salesman route.

The KP is a combinatorial problem, which allocates space in a knapsack in advance according to an object selection. Hence, the total value of all chosen objects is maximized in the knapsack. Martello and Toth [12] state that KP is a very often problem which appears in business, e.g., economic planning, and industry as cargo loading, cutting stock, and bin packing problems. They define the KP problem as an n-object vector of binary variables $x_i(i = 1, \ldots, n)$, in which the object i has a weight w_i and the knapsack has a capacity M. If a fraction x_i; $0 \leq x_i \leq 1$, is placed in the knapsack, then a profit, $g_i x_i$, is earned. The KP aims to find a combination of objects that maximizes the total profit from all chosen objects in the knapsack. While the knapsack's capacity is M, the total weight of all chosen objects to be at most M.

Although having those two problems associated is not unusual, they can appear in complex scenarios. For instance, the medicine distribution at big hospital centers can be viewed as an MTSP-KP instance. Both separation (KP) and distribution (MTSP) have a high number of combinations and, even separated, those tasks are hard to be optimized, requiring sophisticated tools for tackling them. The hospital logistics can ensure patient's safety; however, it is one of the most challenging problems faced by hospital managers, especially in Brazil, due to meeting the organizational needs in a fast, accurate and efficient way. Furthermore, the financial resources addressed to hospital logistics should be implemented efficiently due to the low budget of Brazilian public hospitals.

Some approaches, based on mathematical methods and evolutionary algorithms, are used to optimize MTSP and KP instances. We can cite the following examples Genetic Algorithm – GA [8], Ant Colony Optimization - ACO [6] and Artificial Bee Colony - ABC [9]. These approaches aim to solve problems of steel production [14], cigarette distribution [11], service orders [1], sensor network routing [16], among others.

For the best of our knowledge, there are no reports in the literature regarding the use of global optimization processes to improve the performance of meta-heuristics deployed to solve the MTSP-KP problem. Thus, there is still room for optimizing MTSP-KP parameters with meta-heuristic algorithms aiming to achieve specific goals, especially when one needs to balance the length of the routes and the number of deliveries per agent simultaneously. Then, this work proposes a methodology to optimize MTSP parameters through global population-based optimizers, based on swarm intelligence based algorithms. We use as a case study a medicine distribution process with multiple routes.

The remainder of this paper is structured as follows. Section 2 introduces basic concepts of some swarm intelligence-based algorithms to solve combinatorial problems. Section 3 describes the related works involving meta-heuristic algorithms to solve MTSP instances. Sections 4 and 5 depict the proposed model and the problem instance to be optimized. Section 6 outlines the scenarios and their settings to minimize the MTSP instance. Section 7 presents the optimized results making a comparison among the global population-based optimizers. Finally, Section 8 highlights some conclusions about this and future works.

2 Background

2.1 Team Ant Colony Optimization

The Ant Colony Optimization (TACO), proposed by Vallivaara [15], is based on the Ant Colony System (ACS) to solve MTSP instances. This basic generalization is made by replacing N ACS ants, which build solutions for TSP, with N teams of m members. An ant team represents a salesman in building the MTSP solution, and each team has its taboo list.

All ants of every team are placed at the depot at the beginning of the route construction. To distribute the workload, an ant k with the shortest partial route chooses its next city j, at any moment of the building process, according to the Transition State Rule (TSR) equation, as shown in (1).

$$j = \begin{cases} \text{argmax}_{l \in J_k} \{\tau_{il}[\eta_{il}]^{\beta}\}, & \text{if } q \leq q_0 \\ \text{J}, & \text{otherwise;} \end{cases} \tag{1}$$

After choosing the next city, it is checked if another ant l could add the chosen city to its route and end up with better total route length. If so, that ant can make its move first not choosing j. This checkpoint avoids the algorithm to force non-optimal solutions.

TACO has several parameters which are responsible for the its behavior while building solutions. The initial probability q_0 determines whether the ants' initialization has only deterministic or random choices ($0 \leq q_0 \leq 1$). The pheromone parameters α and β define the weight of the pheromone trail and the visibility, respectively, in the choice of the next node by the ant. The parameter ξ controls the pheromone persistence when the Pheromone Update Rule (PUR) takes place locally, just after an ant moves from one city to another, that is, it includes one more edge on its route. Likewise, ρ regulates pheromone persistence for global PUR, i.e., at the end of each cycle of the algorithm.

2.2 Particle Swarm Optimization

Kennedy and Eberhart [10] proposed the Particle Swarm Optimization (PSO) method based on bird flocking. PSO is suitable for the optimization of continuous variables in a high-dimensional search space and presents high precision. It performs searching via a swarm of particles through an iteration process. Each particle moves towards its previous best (P_{best}) position and the global best (G_{best}) position in the swarm to achieve the optimal solution.

The solution represents the particle position in the search space, a vector xi. For each step, the particles have their positions according to their velocity vector v_i. The velocity clamping, an upper bound for the velocity parameter avoids particles flying out the search space. Likewise, the "constriction coefficient" strategy, proposed by Clerc and Kennedy [5], constricts the velocities through the dynamic swarm analysis.

2.3 Fish School Search

Bastos-Filho et al. [2] developed a population-based search algorithm inspired by fish swimming behavior, which expands and contracts while looking for food. The Fish School Search (FSS) algorithm considers the individual and collective fish movements. This optimization algorithm does not present the same exploitation capability of the PSO, but it has the capability to find good solutions in a search space with many local minima.

Each fish, in n-dimensional location, represents a feasible solution for the problem. Its success is measured by its weight, a cumulative account of how successful the search for each fish in the school has been. The fishes not only store information about their weight but also position in the search space.

FSS consists of moving and feeding operators. On the individual movement, each fish randomly moves towards a position in its neighborhood looking for promising regions. After moving to new positions, all fishes have their weights updated. The weight update is determined by the individual movement success, which is computed through the fitness of current and new positions. After feeding all fishes, the collective-instinctive movement takes place. All fishes move towards an influence vector. Those fishes that improved their fitness in the current iteration generate this vector.

At the end of the current iteration, the school contracts or expands according to the volitive-collective movement operator. The school's contraction results in an exploitation search whereas its expansion make the school explore the search area avoiding local minima. Thus, the volitive operator computes the school's barycenter. This last operator gives to the FSS the capability to self-adjust the granularity of the search along the optimization process.

3 Related Works

Although there is not an approach related to medicine distribution until the present moment, there is a set of computational problems, which are transversal to one tackled in this paper. Those problems have been studied to improve commercial production and distribution under one or more objective constraints.

Somhom, Modares and Enkawa [13] use a Competition-Based Neural Network (CBNN) to minimize the longest route in an MTSP (based on the TSPLIB) instance with a single depot closed routes. Tang et al. [14] use a Modified Genetic Algorithm (MGA) to improve production scheduling of hot rolling from an iron and steel industry in China. Carter and Ragsdale [4] use a Genetic Algorithm (GA) with a new chromosome and MTSP operators. Wang et al. [16] use the Ant Colony System (ACS) to group and route sensor nodes from a wireless multimedia network with a limited time interval as a constraint. Vallivaara [15] proposes the Team Ant Colony Optimization, based on ACS rules to manage routes with multiple robots in a hospital environment. The problem constraint is to minimize the longest route and the sum of routes. Liu et al. [11] use the ACS and the Max-Min Ant System to solve the distribution of cigarettes in a Chinese company. Barbosa and Kashiwabara [1] use the Single Team Ant Colony System (STACS), based on TACO, to route the service orders in an energy distribution company.

4 Proposed Model

The proposed model consists of using a base algorithm to compute the best solutions for an MTSP instance while having an global optimizer algorithm to seek the best parameters sets for the base algorithm. In this work, the Team Ant Colony Optimization (TACO) is used as the base algorithm. Moreover, the Particle Swarm Optimization (PSO) and the Fish School Search take part in the optimization of the values for TACO parameters.

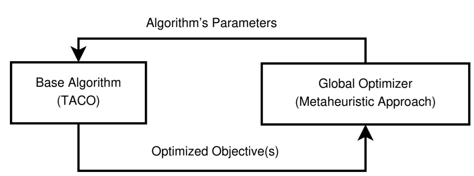

Fig. 1. Proposed Model for optimizing the TACO

As shown in Figure 1, the Global Optimizer (GO) starts to generate a set of values for the TACO's parameters. In this case, the optimized parameters from TACO are α (pheromone relevance), β (relevance of pheromone's visibility), ξ (pheromone persistence, local update) and ρ (pheromone persistence, global update). Then, TACO builds a set of best solutions based on the optimized parameters. Next, TACO returns to GO the set of best solutions, which are evaluated to the meta-heuristic algorithm. As the iterations happen, the GO keeps a record of the best set of parameters, which have been found so far. The execution finishes when the GO reaches the limit of iterations.

5 Problem Instance

A Medicine Distribution Center (MDC) at a public hospital in Brazil usually executes the orders manually, both packaging the orders and building the routes. It does not matter how long it would take for the deliverymen to do that. Due to the considerable time variation for answering the orders, it is difficult to determine the delivery capacity for each agent in order to minimize individual costs of the built routes.

That real-life problem highlights a set of objectives, which can be optimized. One of those objectives is to reduce the total sum of the routes without worrying about the work balance between the deliverymen. Another objective is to have balanced routes at the end of order execution, which also results in more orders being executed at the same time interval, i.e., in this case the target is to avoid a partially inoperative deliveryman because it has a route significantly lower than the others.

Figure 2 highlights 16 pharmacies and an MDC, represented as $V1$, of a real hospital environment from Brazil represented as a graph with costs associated with distances for the deliveries. The MDC is the depot where the deliverymen start and finish their routes. A matrix, called cost matrix, contains the data of each pharmacy like id, latitude, longitude and its distance. The matrix helps to calculate the route cost.

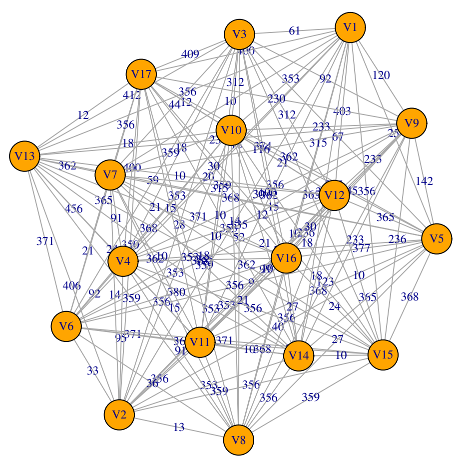

Fig. 2. Graph Model

Likewise, another matrix, called the data matrix, contains the delivery orders for each node in the built model. Each order contains ids of the deliveryman and pharmacy, the initial and final distances, when the deliveryman left the MDC and the duration of the order. The data matrix helps to simulate a day of ordering in the depot. Also, there are four deliverymen to deliver the orders to the pharmacies.

As stated in Section 4, a global optimizer, based on swarm intelligence, is also used to improve the TACO results by optimizing its parameters. FSS and PSO are taken as a global optimizer of the TACO parameters. The standard values for the set of used parameters are: number of deliverymen $M = 4$, the number of teams $N = 10$, initial probability $q_0 = 0.5$, pheromone relevance $\alpha = 0.5$, visibility relevance $\beta = 1.0$, pheromone persistence for local update $\xi = 0.1$, pheromone persistence for global update $\rho = 0.1$. Those default values were obtained by a parameter test. The stop criterion is 1000 iterations for each independent run. The global optimizers optimize the subset of parameters $P = \{\alpha, \beta, \xi, \rho\}$ in a range $0 \leq P \leq 2$.

6 Experiments

The experiments presented in this section aims to show the effectiveness of the developed methodology applied to the real problem. In the first scenario, the algorithms are configured to minimize the total cost of the solutions, without considering the distribution of the workload between the teams, as in the general description of the MTSP. In the second scenario, the algorithms minimize the cost of the largest individual route of the solutions, aiming the construction of solutions formed by routes with equal costs between the deliverymen, as in the MTSP with workload balance.

All Experiments were executed on a MacBook Pro (13-inch, Late 2011) with 2.4GHz Intel Core i5 CPU, 16GB of RAM (1333MHz DDR3) and macOS High Sierra (version 10.13.4) operating system. We used the database presented in Section 4. Then, we performed 30 independent runs of the algorithms for each experiment. TACO algorithm was coded in Java based on the proposed algorithm by Vallivaara [15]. The FSS is based on Bastos-Filho et al. [2] version. The single objective PSO were taken from the jMetal framework [7].

TACO takes part in the experiment as a base algorithm. Before starting the experiments of the proposed model, a base test has to be taken to prove the effectiveness and robustness of the chosen algorithm. The default values for both scenarios are stated in Section 5. The FSS as a GO is executed with a stop criterion of 1000 iterations per run, and it has 30 independent runs. The values of FSS parameters were taken from [2]. Similarly, the PSO has the same values of stop criterion and independent runs. The values of PSO parameter remained the same as in [7].

7 Results

The first set of experiments was carried out in order to minimize the Total Cost of Routes (TCR), i.e., the sum of each team's route. By comparing the solutions, the solution, which has the lowest total cost, is considered the best one. This scenario can be applied to real life when we aim to reduce the total amount of the deliverymen routes instead of prioritizing the work balance.

The second set of experiments aimed to minimize the Longest Route (LR) of the solutions keeping the same values of the parameters in the first scenario. That case is suitable for real situations when we prioritize the balance among the individual routes (work balance) rather than the total sum of routes.

7.1 Experiments Carried Out without Global Optimizers

Table 1 shows the average execution time of the algorithm for the instance. This value was obtained from the average of the time spent to execute the 1000 iterations with 30 independent runs of the algorithm, for one working day.

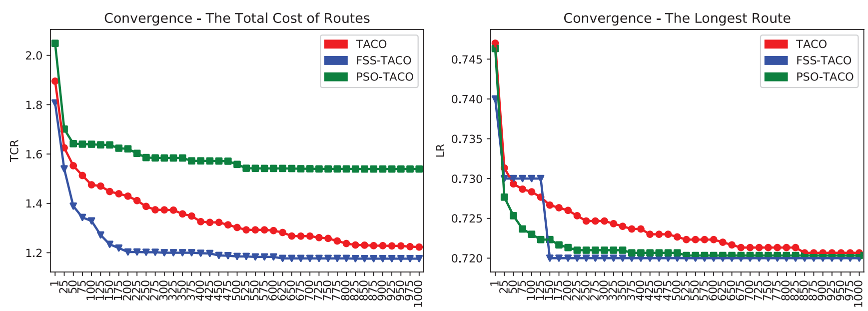

Fig. 3. Comparison among the three approaches when minimizing TCR and LR

TACO presented satisfactory results for the two MTSP variations: minimizing the total cost of the solution and minimizing the cost of the largest individual solution route (Table 1). The small values of the standard deviations in the two tables confirm the robustness of the algorithm when generating solutions with costs close to the average of the 30 executions. Although LR is being minimized, TCR varies throughout the iterations. The same situation happens when TACO minimizes TCR. Bearing this in mind, there is still room to make improvements in the results of both objectives.

7.2 Experiments Carried Out with Global Optimizers

The experiments carried out with global optimizer showed better results for both scenarios, as displayed in Table 1. FSS has a better improvement of minimizing the longest route in comparison to the base algorithm. This result is due to the FSS's capacity to exploit the search space. Besides, FSS has a lower standard deviation which corroborates to its robustness.

PSO also had better results comparing to the TACO approach without global optimizers (see Table 1). Its standard deviation also shows the robustness and effectiveness when optimizing TCR and LR.

7.3 Comparison

All the previous results are compiled in Table 1 for a better understanding of them. Comparing the three approaches, FSS got better results when minimizing both TCR and LR. It also had the lowest result with a standard deviation close to zero. As seen in Figure 3, in both scenarios, FSS converged earlier than PSO and the base-algorithm. Results of PSO were better than base-algorithm results and converged earlier than TACO as expected.

Table 1. Comparison of the Results with Average of 30 Independent Runs

Approach	**Minimizing TCR**				**Minimizing LR**			
	TCR	S.D.	LR	S.D.	TCR	S.D.	LR	S.D.
TACO	1.8	0.25	0.721	0.003	1.223	0.050	0.756	0.012
FSS	1.0797	0.002	0.747	0	1.944	0.335	0.719	0.001
PSO	1.115	0.032	0.750	0.003	2.048	0.333	0.720	0.005

Figure 4 shows the standard deviation for both TCR and LR. When minimizing the TCR, we can notice that the results from TACO varies more than FSS and PSO. In the other hand, PSO has the shortest variation of its results. When minimizing the LR, it is noteworthy that for all approaches (TACO, FSS and PSO) there is a small or no variation among their results.

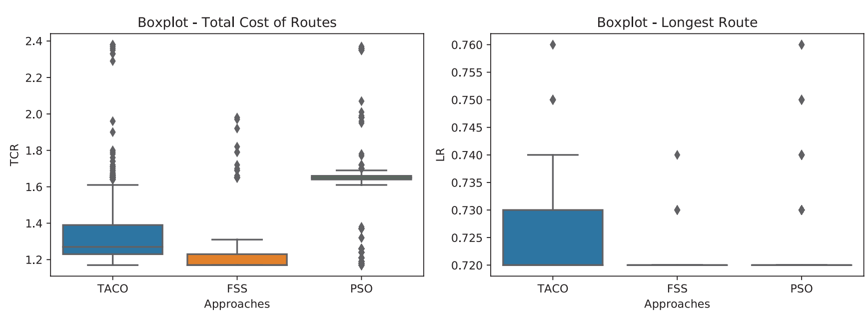

Fig. 4. Standard Deviation of TCR and LR minimizations

8 Conclusion

The proposed model described was efficient in the distribution of work orders among the deliverymen and in the creation of optimized routes to carry out the services. However, some questions need to be analyzed, such as multi-objective optimization of the total cost of the routes and the longest route and optimization of the deliverymen's knapsacks.

For the creation of solutions optimized for MTSP in this work, an algorithm based on the ACO metaheuristic was implemented and associated with an global optimizer. The FSS and PSO were responsible for optimizing the TACO parameters in order to improve the results.

As shown in Section 6, FSS had the best results for both scenarios with the lowest minimization results for the Total Cost of Routes (TCR) and the Longest Route (LR) with standard deviations around zero. In those two cases, FSS as a global optimizer converged earlier than the other two approaches.

Another approach to the MTSP problem is to optimize both objectives, TCR and LR. To achieve this goal, a multi-objective optimization algorithm called Multi-Objective Fish School Search (MOFSS) will take part in the minimization process as a global optimizer. Then, a new comparison will be made among MOFSS and the global optimizers used in this work.

References

1. Barbosa, D.F., Silla Jr, C.N., Kashiwabara, A.Y.: Aplicaçao da otimizaçao por colônia de formigas ao problema de múltiplos caixeiros viajantes no atendimento de ordens de serviço nas empresas de distribuiçao de energia elétrica. Anais do XI Simpósio Brasileiro de Sistemas de Informaç ao pp. 23–30 (2015)
2. Bastos Filho, C.J., de Lima Neto, F.B., Lins, A.J., Nascimento, A.I., Lima, M.P.: A novel search algorithm based on fish school behavior. In: Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on. pp. 2646–2651. IEEE (2008)
3. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega **34**(3), 209–219 (2006)
4. Carter, A.E., Ragsdale, C.T.: A new approach to solving the multiple traveling salesperson problem using genetic algorithms. European journal of operational research **175**(1), 246–257 (2006)
5. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE transactions on Evolutionary Computation **6**(1), 58–73 (2002)
6. Dorigo, M., de Oca, M.A.M., Engelbrecht, A.: Particle swarm optimization. Scholarpedia **3**(11), 1486 (2008)
7. Durillo, J.J., Nebro, A.J.: jmetal: A java framework for multi-objective optimization. Advances in Engineering Software **42**(10), 760–771 (2011)
8. Holland, J.H.: Genetic algorithms. Scientific american **267**(1), 66–73 (1992)
9. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005)
10. Kennedy, J., Eberhart, R.C.: The particle swarm: social adaptation in information-processing systems. In: New ideas in optimization. pp. 379–388. McGraw-Hill Ltd., UK (1999)
11. Liu, W., Li, S., Zhao, F., Zheng, A.: An ant colony optimization algorithm for the multiple traveling salesmen problem. In: Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on. pp. 1533–1537. IEEE (2009)

12. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Inc., New York, NY, USA (1990)
13. Somhom, S., Modares, A., Enkawa, T.: Competition-based neural network for the multiple travelling salesmen problem with minmax objective. Computers & Operations Research **26**(4), 395–407 (1999)
14. Tang, L., Liu, J., Rong, A., Yang, Z.: A multiple traveling salesman problem model for hot rolling scheduling in shanghai baoshan iron & steel complex. European Journal of Operational Research **124**(2), 267–282 (2000)
15. Vallivaara, I.: A team ant colony optimization algorithm for the multiple travelling salesmen problem with minmax objective. In: Proceedings of the 27th IASTED International Conference on Modelling, Identification and Control. pp. 387–392. ACTA Press (2008)
16. Wang, X., Wang, S., Bi, D., Ding, L.: Hierarchical wireless multimedia sensor networks for collaborative hybrid semi-supervised classifier learning. Sensors **7**(11), 2693–2722 (2007)