A New Hybrid Ant Colony Algorithms for The Traveling Thief Problem

Wiem Zouari
COSMOS Laboratory
National School of Computer Science
University of Manouba
Tunisia
wiem.zouari@ensi-uma.tn

Ines Alaya
COSMOS Laboratory
National School of Computer Science
University of Manouba
Tunisia
ines.alaya@ensi-uma.tn

Moncef Tagina
COSMOS Laboratory
National School of Computer Science
University of Manouba
Tunisia
moncef.tagina@ensi-uma.tn

ABSTRACT

The Traveling Thief Problem (TTP) is a new problem recently proposed in the literature. The TTP combines two well-known optimization problems: the knapsack problem (KP) and the travelling salesman problem (TSP). In this paper, new hybrid ant colony algorithms are presented. We study and compare different approaches for solving the TTP. The first approach is a centralized and static metaheuristic, the second is a dynamic metaheuristic and the third is a distributed metaheuristic. The obtained results prove that our algorithms are efficient for instances of TTP.

CCS CONCEPTS

• **Computing methodologies** → **Heuristic function construction**; • **General and reference** → Empirical studies.

KEYWORDS

The Traveling Thief Problem, dynamic metaheuristic, distributed metaheuristic

ACM Reference Format:

Wiem Zouari, Ines Alaya, and Moncef Tagina. 2019. A New Hybrid Ant Colony Algorithms for The Traveling Thief Problem. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3326785

1 INTRODUCTION

Many practical applications can be formulated as a *combinatorial optimization problem*. A *combinatorial optimization problem* generally consists in traversing a search space in order to extract an optimal solution from among a finite set of solutions while maximizing (or minimizing) an objective function. A *combinatorial optimization problem* is *static* when optimization refers to a process of minimizing (maximizing) the costs (benefits) of certain objective functions for a single instance. Unlike the *dynamic optimization problem* whose optimization refers to this process over a period of time. In both cases, the equality and inequality constraints can be applied. On the other hand, an optimization problem is *distributed* when it is subdivided into smaller sub-problems and which can be simultaneously solved in *parallel*. Indeed, a *parallel* program that resolves the distributed problem can be faster than its sequential equivalent. However, *centralized resolution* is still a complicated process in view of the complexity of the problem and the processing time. In this work, we are interested in studying and comparing those different aspects of optimization problems in order to define adequate algorithms with improved performance for the TTP [1].

2 MMACS

MMACS is a *hybrid ant colony optimization algorithm* based on the foraging behavior of ants. MMACS was introduced in [2] and revisited in [3]. This algorithm presents two hybridization levels. The first hybridization consists in integrating the Ant Colony System selection rule in MAX-MIN Ant System [4]. The second level of hybridization is to combine the hybridized ant colony optimization algorithm and an algorithm based on a local search heuristic, then both algorithms are operating sequentially. The pseudo-code of MMACS algorithm is represented by algorithm 1 where τ_{max} is the

Algorithm 1 MMACS pseudo-code

```
Initialize pheromone trails to τ_max
repeat
  repeat
    Select randomly a first item
    Remove from candidates each item that violates resource
    constraints
    while Candidates ≠ ∅ do
      if a randomly chosen q is greater than q_0 then
        Choose item o_j from Candidates with probability P_ij^k
      else
        Choose the best next item
      end if
      Remove from candidates each item that violates resource
      constraints
    end while
    Update S_best
  until maximum number of ants is reached or optimum is
  found
  Update pheromone trails
until maximum number of cycles is reached or optimum is found
Apply a local search algorithm
```

upper bound of the pheromone trails, q is a random variable uniformly distributed in $[0, 1]$, q_0 $(0 \leq q_0 \leq 1)$, P_{ij}^k is the probabilistic

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326785

action choice rule and S_{best} is the best solution found all along the execution. After that, the 2–opt algorithm takes a current solution as input and returns a better accepted solution to the problem, if it exists. The 2–opt algorithm is used once ants have completed their solution construction, thereby improving the solution by approaching the best one or even reaching it. Our proposed 2–opt algorithm can be written as represented by algorithm 2.

Algorithm 2 A 2–opt pseudo-code

Initialize Candidates by observing S_{best}
repeat
  for each item $o_j \in Candidates$ **do**
    for each item $o_i \in S_{best}$ **do**
      S'_{best} = Swap (o_i, o_j)
      if constraints are satisfied by S'_{best} and S'_{best} is better than S_{best} **then**
        Update the best solution
      end if
    end for
  end for
until no improvement is made

3 CMMACS

Our first approach CMMACS to solve TTP is a *static* and a *centralized* methaheuristic that combines a 2–opt–based neighborhood search and the hybrid ant colony algorithm (MMACS). The algorithm uses the 2–opt search for the TSP part to find the tour and the MMACS methaheuristic to solve the picking plan.

4 OMMACS

Our second approach OMMACS is a *dynamic* variation of CMMACS, which was developed to solve TTP over time. Dynamic processing involves processing the input as a stream of data. Thus, the input is provided piece by piece, without all data being available from the start. The best solution is approached at each step of the time.

5 DMMACS

Our third approach DMMACS is a *distributed* version of CMMACS algorithm. DMMACS solves the TTP complex problem by decomposing it into parts, stores the results of parts and computes the global result by combining sub-solutions.

6 EXPERIMENTS

In this section, we study the results of a set of experiments that was carried out to determine the efficacy of the proposed algorithms: CMMACS, OMMACS and both versions of DMMACS, the *serial* and *parallel* one. In order to evaluate the performance of new algorithms, experiments were conducted on TTP instances. Namely, we used a set of ten *bounded strongly correlated instances* `a280` from [5] where the dimension of TSP sub-problem is 280 and the number of items of KP sub-problem is 279.

In order to facilitate the visualization of the algorithms´ performance, the obtained results are represented by curves in Fig 1. In fact, the results show that OMMACS has improved the process of obtaining a better and faster solution for the used TTP instances.

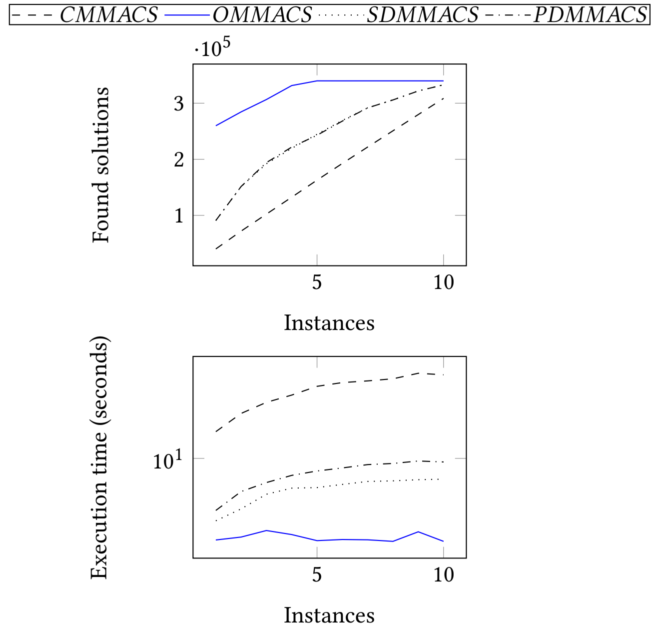

Figure 1: Results of best ten tests.

A possible explanation for the solid performance of OMMACS, mainly regarding the execution time, is that the complexity of building a solution for every step of the time based on a sub local solution should be substantially lower.

7 CONCLUSIONS

The paper presents a comparative study of the proposed hybrid algorithms CMMACS, OMMACS and DMMACS while solving the traveling thief problems. Experiments show that OMMACS turned out to outperform both CMMACS and DMMACS algorithms. It is also noticed from the results that the parallel version of DMMACS takes longer than the serial version. In fact, by using small instances of TTP we are not taking advantage of processing in parallel.

As perspective, it is intended to verify the algorithms´ performance while solving large instances in order to measure the performance of multithread architecture.

REFERENCES

[1] Mohammad Reza Bonyadi, Zbigniew Michalewicz, and Luigi Barone. 2013. The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. In *2013 IEEE Congress on Evolutionary Computation*. IEEE, 1037–1044.
[2] Wiem Zouari, Ines Alaya, and Moncef Tagina. 2017. A Hybrid Ant Colony Algorithm with a Local Search for the Strongly Correlated Knapsack Problem. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 527–533. https://doi.org/10.1109/aiccsa.2017.61
[3] Wiem Zouari, Ines Alaya, and Moncef Tagina. 2018. A Comparative Study of a Hybrid Ant Colony Algorithm MMACS for the Strongly Correlated Knapsack Problem. *Advances in Science, Technology and Engineering Systems Journal* 3, 6 (2018), 1–22. https://doi.org/10.25046/aj030601
[4] Marco Dorigo and Thomas Stützle. 2019. Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics*. Springer, 311–351.
[5] Sergey Polyakovskiy, Mohammad Reza Bonyadi, Markus Wagner, Zbigniew Michalewicz, and Frank Neumann. 2014. A comprehensive benchmark set and heuristics for the traveling thief problem. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 477–484.