Proceedings of the 48th ORSSA Annual Conference
pp. 106–113
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3

The Travelling Thief Problem for advancing combinatorial optimisation

KM Malan*

Abstract

When benchmarking the performance of combinatorial optimisation algorithms or teaching students about combinatorial optimisation, it is common practice to consider self-contained classical problems such as the *Travelling Salesman Problem* or the *Quadratic Assignment Problem*. Although these classical problems occur in many real-world problems, they are seldom found in isolation. Instead, real-world problems are more commonly composed of different sub-problems that interact with each other in non-trivial ways. The *Travelling Thief Problem* (TPP) was recently introduced with the aim of providing a benchmark problem that more closely matches some of the complexity evident in real-world problems. This paper provides a survey of the research emanating from the introduction of this problem. It is argued that the introduction of this problem is advancing the field of combinatorial optimisation in positive ways that is directing research to be more focused on addressing some of the challenges that arise when solving real-world problems.

Key words: Combinatorial optimisation, travelling thief problem, multiple interdependent components.

1 Introduction

The *Travelling Thief Problem* (TTP) was introduced in 2013 by Bonyadi *et al.* [3] to address the need for more realistic problems for benchmarking metaheuristics. The TTP is a combination of two classical combinatorial problems: the *Travelling Salesman Problem* (TSP) and the *Knapsack Problem* (KP). Both the TSP and the KP are NP-hard problems, making it infeasible to find the optimal solution for large instances. Heuristic and meta-heuristic approaches have therefore been widely used for finding approximate solutions to these individual problems in a reasonable time.

Real world problems frequently consist of sub-problems that are interdependent. This implies that finding the best solutions of the sub-problems in isolation is not helpful, because the overall quality depends on the interaction between the sub-problems [3]. The TTP captures this complexity by combining two sub-problems that depend on each other for constructing feasible solutions of high quality. This is confirmed by Mei *et al.* [17]

*Department of Decision Sciences, University of South Africa, PO Box 392, UNISA, 0003, Republic of South Africa, email: malankm@unisa.ac.za

when they show that the computational complexity of the search required for TTP is much higher than the complexity required for the individual TSP and KP sub-components of the problem.

The parameters of the TTP can be used to control the level of interdependence between the two sub-problems. Instances of the TTP can be designed to have simple dependencies that can be decomposed and solved more easily, but instances can also be designed to have tighter dependencies so that they are not easily decomposable [5]. There are different ways of modelling the problem and choosing the most appropriate model depends on the aims of the researcher.

This paper provides an overview of research in the TTP: proposed models of the problem (§2), problem instances and characteristics (§3), and algorithmic solutions (§4). Finally, §5 discusses the ways in which the introduction of the TTP is advancing research in discrete optimisation and argues for the value in the TTP as a case study for tuition in combinatorial optimisation.

2 The travelling thief problem

The original formulation of the TTP is as follows [3]. A thief carrying a knapsack of maximum capacity, W, visits each of n cities exactly once, picking at most m items at cities to fill his knapsack. The thief starts and ends at city 1, but may only pick up items on the first visit to city 1. Each item I_k has a value p_k and a weight w_k and is available at a subset of cities: $A_k \subseteq \{1, \ldots, n\}$, but the thief is only able to pick one of each item. For example, $A_5 = \{1, 10, 30\}$ indicates that item 5 is available at cities 1, 10 and 30. If the thief decides to pick item 5, he can only do so at one of these three cities. A matrix $D = d_{ij}$ defines the distances d_{ij} between cities i and j. A solution to the TTP is in the form of two vectors:

- a tour $\overline{x} = (x_1, \ldots, x_n)$, where x_i is the index of a city, and
- a picking plan $\overline{z} = (z_1, \ldots, z_m), z_k \in \{0 \cup A_k\}$, indicating from which city item I_k should be picked (0 implies the item is not picked at all).

The tour element, $\overline{x}$, is the same as a solution in the traditional TSP. However, the picking plan, $\overline{z}$, differs from the traditional KP in being an m-length vector of city index values, in contrast to the m-length binary vector of KP.

Four versions of the TTP model are described below.

Model 1: TTP$_1$

TTP$_1$ [3] has a single objective to maximise the benefit which equals the value of the items in the knapsack minus a knapsack rental cost of R per time unit. The thief travels at a current velocity, v_c, which is related to the current weight of the knapsack, W_c, and is defined as

$$v_c = v_{max} - W_c \left(\frac{v_{max} - v_{min}}{W} \right), \tag{1}$$

where v_{max} and v_{min} are the maximum and minimum velocities of the thief. When the knapsack is empty ($W_c = 0$), the thief travels at the maximum velocity and when the knapsack is full ($W_c = W$), the thief travels at the minimum velocity.

The time that a thief takes to travel between two cities is the distance between the two cities divided by the current velocity of the thief, and the total time for the thief to travel the full tour $\overline{x}$ with picking plan $\overline{z}$ is defined as

$$t(\overline{x}, \overline{z}) = \sum_{i=1}^{n-1}(t_{x_i,x_{i+1}}) + t_{x_n,x_1}, \tag{2}$$

where

$$t_{x_i,x_j} = \frac{d_{ij}}{v_c}.$$

Given the above, the single-objective function for TTP_1 is to maximise

$$f(\overline{x}, \overline{z}) = g(\overline{z}) - R \times t(\overline{x}, \overline{z}), \tag{3}$$

where $g(\overline{z})$ is the total value of the picked items, subject to the knapsack capacity, W.

Notice the interdependence between the sub-problems: the selected tour affects the time travelled via the distances (as in the usual TSP), but the picking plan also affects the time travelled via the velocity of the thief. One picking plan may result in a higher value than another, but could also result in higher rental by slowing down the thief.

Model 2: TTP_2

TTP_2 [3] is formulated as a bi-objective problem: maximisation of the value and minimisation of travel time. The thief travels at a velocity as defined for TTP_1, however, the value of picked items reduce over time while they are being carried in the knapsack. Each item is multiplied by a factor smaller than 1, equal to $\delta^{\lceil \frac{T_k}{C} \rceil}$, where δ is a proportion of value per time unit of travel, T_k is the total time that item k is carried in the knapsack from the time it is picked up to the end of the tour, and C is a constant.

The objective for TTP_2 is to simultaneously maximise the value in the knapsack, $g(\overline{x}, \overline{z})$, and minimise time, $t(\overline{x}, \overline{z})$. Notice that in this formulation, both the value and the time depend on the tour and the picking schedule, because the chosen tour affects the time each item is in the knapsack and the weight of the items affects the speed of the thief.

Model 3: BO-TTP

Wu *et al.* [28] propose a second bi-objective formulation that involves maximising the combined value function as in equation 3 while simultaneously minimising the accumulated weight. This has the effect of treating the constraint as a second objective.

Model 4: MTTP

Chand and Wagner [6] propose a formulation called the *Multiple TTP* (MTTP) with multiple thieves travelling to cities with the aim of maximising the collective profit of the group. Added to the original formulation, the MTTP includes p thieves and each thief can visit a set of n_p cities ($n_p \leq n$). Unlike the original TTP, visiting all cities is not required. All thieves start and end at city one and can visit the same city, but each item can only be picked up by one thief. This modified formulation of the TTP is more complex as it involves multiple tours and packing plans.

3 Travelling thief problem instances and characteristics

When the TTP was originally introduced, Bonyadi *et al.* [3] proposed a procedure for generating problem instances and later provided a set of forty-five TTP instances [4]. At the same time, Polyakovskiy *et al.* [21] established a large benchmark suite of 9 720 different TTP instances to cover a wide range of features and different levels of correlation between the sub-components. This suite has been the basis for a number of TTP competitions. The first competition at the 2014 Congress on Evolutionary Computation focused on the single-objective formulation of the problem, TTP_1, while the most recent competition at the 2019 Genetic and Evolutionary Computation Conference is focused on the bi-objective formulation, TTP_2.

A few studies have focused on understanding the complexity and nature of the TTP problem. Wu *et al.* [26] investigated the effect of the knapsack renting rate on problem difficulty and showed how this parameter could be manipulated to create hard problem instances for evolutionary algorithms. Polyakovskiy and Neumann [22] showed that a simpler version of the TTP with a fixed route is NP-hard for both the constrained and unconstrained cases. Wagner *et al.* [25] investigated the link between basic features of the TTP and algorithm performance and found that the most important features for distinguishing algorithm performance were the knapsack capacity and the renting rate.

El Yafrani *et al.* [10] performed a fitness landscape analysis based on a full enumeration of the search space of small TTP instances. They used local optimal networks [20] to model the global structure of search space using the Cartesian product between the neighbourhoods of the two sub-problems as the basis for the definition of neighbourhood. Their analysis showed that there is a direct correlation between lower knapsack capacity and the problem hardness.

4 Algorithmic approaches to solving the TTP

Most of the techniques reported in the literature to solving the TTP are approximate heuristic or metaheuristic approaches. One study by Wu *et al.* [27] proposed three exact techniques to solving the TTP, using dynamic programming, branch and bound search, and constraint programming. These exact techniques are useful for evaluating the success of approximate techniques, but are only applicable to small TTP instances.

The simplest heuristic approach to solving the TTP is to tackle the two sub-problems sequentially, by first solving the TSP component of the problem — using an approach such as the *Chained Lin-Kernighan* (CLK) [1] — and then using a heuristic to choose a picking plan for the given TSP solution. A number of studies have used this approach, differing only in the heuristics for the KP sub-problem using the route found by CLK [4, 12, 13, 21]. Mei *et al.* [17] extended this two-stage approach to use a population of tours generated by CLK and later proposed the evolution of the gain and picking heuristic functions using genetic programming [18]. Wagner [24] deviated from the practice of solving the TSP component using CLK by using ant colony optimisation for finding tours, followed by heuristics for the KP sub-component and further local search on the tours. He found that this approach was able to find better quality solutions by considering longer tours, but only for instances with up to 250 cities and 2 000 items.

Faulkner *et al.* [12] performed a comparative analysis of eleven different heuristic approaches on seventy-two instances of the TTP and concluded that there is no single best heuristic approach to solving the TTP. Given the fact that there is no single best heuristic approach to solving TTP (known as *performance complementarity* [14]), Martins *et al.* [16] and El Yafrani *et al.* [11] proposed hyper-heuristic approaches to solving the TTP that select the best combination of heuristics for a particular problem instance.

An alternative to the two-stage sequential heuristic approach includes some mechanism for influence between the two sub-problems. Bonyadi *et al.* [4] proposed CoSolver, that solves the two sub-problems in parallel, but includes revision of solutions through "negotiation" between the two solvers. El Yafrani and Ahiod [7] proposed a modification to CoSolver utilising local search metaheuristics (a hill climber and simulated annealing) and Nieto-Fuentes *et al.* [19] proposed a further improvement to CoSolver by using guided local search to escape local optima in the tour optimisation stage. A different approach to combining the sub-problems was proposed by El Yafrani & Ahiod [9]. They used the usual two-stage approach of first finding a CLK tour followed by a heuristic-based picking plan, but then implemented a local search using a joined neighbourhood formed by the Cartesian product of the neighbourhoods of the two sub-problems. Although their approach gave good results for small instances, the high computational complexity resulting from the Cartesian product neighbourhood is a limitation of the approach.

All of the above approaches involved solving the first, single-objective model, TTP_1. In contrast, El Yafrani *et al.* [8] investigated solving the bi-objective model, TTP_2, using the NSGA-II framework to search for a Pareto set of solutions. Blank *et al.* [2] proposed three approaches to the bi-objective TTP_2: two deterministic approaches and the third using NSGA-II. They showed that the NSGA-II approach provided the best performance of the algorithms. Wu *et al.* [28] proposed a solution to the second bi-objective formulation, BO-TTP, using an evolutionary approach to constructing tours and a dynamic programming approach to solving the picking plan.

In the case of the third model, the MTTP, only the authors that introduced the formulation, Chand & Wagner [6], proposed a number of heuristics for solving the MTTP.

5 Advancing the field of combinatorial optimisation

Considering the algorithmic solutions to the TTP proposed in the literature, all proposed solutions use a combination of different algorithmic approaches. Most solutions use established heuristics that have been in use for decades in combination with newly designed heuristics, local search and/or established metaheuristics. The performance complementarity [14] of different algorithmic approaches to the TTP is clear in the success of algorithm selection in solving the problem, as investigated by Wagner *et al.* [25]. In the context of a complex problem with interdependent subcomponents, introducing a "new" metaheuristic inspired by some arbitrary phenomenon in nature [23] would be counterproductive. The challenge is rather in finding efficient and effective ways of combining established techniques.

Another challenge for researchers that has become evident through the TTP is the need for new ways of understanding search spaces and neighbourhood of these interdependent subspaces. Fitness landscape analysis is commonly used in the evolutionary computation community to understand complex problems and guide the choice of appropriate algorithms [15]. The notion of neighbourhood is central to the concept of fitness landscapes. Preliminary work in the landscape analysis of TTP [10] defined neighbourhood based on the Cartesian product between the neighbourhoods of the two sub-problems. Alternative notions of neighbourhood are needed for landscape analysis of the TTP to be computationally feasible.

The multi-faceted nature of the TTP makes it an ideal case study for educational modules in combinatorial optimisation. The TSP and KP sub-components can be introduced individually at first, covering permutation-based and binary search spaces. Small instances can be used to introduce exact solvers, moving to heuristic solvers and metaheuristics for larger instances. Using a problem like TTP will convey the essential message to students that there is value in different approaches and it is in the combination of approaches that we are able to develop good solutions to complex problems.

6 Conclusion

There has been a flurry of research activity in the evolutionary computation community dedicated to the newly introduced *Travelling Thief Problem* (TTP). It is argued that this benchmark problem has had a positive effect on research in combinatorial optimisation, resulting in researchers focusing on the combination of different approaches from classical heuristic methods to newer evolutionary and swarm-based approaches. The problem has also highlighted a new direction for research in fitness landscape analysis to accommodate these complex interdependent sub-spaces. Furthermore, the problem provides an excellent case study for tuition in combinatorial optimisation.

References

[1] Applegate D, Cook W & Rohe A, 2003, *Chained Lin-Kernighan for large traveling salesman problems*, INFORMS Journal on Computing, **15(1)**, pp. 82–92.

[2] Blank J, Deb K & Mostaghim S, 2017, *Solving the bi-objective traveling thief problem with multi-objective evolutionary algorithms*, Proceedings of the 9th International Conference on Evolutionary Multi-Criterion Optimization, Münster, pp. 46–60.

[3] Bonyadi MR, Michalewicz Z & Barone L, 2013, *The travelling thief problem: the first step in the transition from theoretical problems to realistic problems*, Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancún, pp. 1037–1044.

[4] Bonyadi MR, Michalewicz Z, Przybylek MR & Wierzbicki A, 2014, *Socially inspired algorithms for the travelling thief problem*, Proceedings of the 2014 Genetic and Evolutionary Computation Conference, Vancouver, pp. 421–428.

[5] Bonyadi MR, Michalewicz Z, Wagner M & Neumann F, 2019, *Evolutionary computation for multicomponent problems: Opportunities and future directions*, pp. 13–30 in Datta S & Davim J (Eds.), *Optimization in Industry: Present practices and future scopes*, Springer, Cham.

[6] Chand S & Wagner M, 2016, *Fast heuristics for the multiple traveling thieves problem*, Proceedings of the 2016 Genetic and Evolutionary Computation Conference, Denver (CO), pp. 293–300.

[7] El Yafrani M & Ahiod B, 2016, *Population-based vs. single-solution heuristics for the travelling thief problem*, Proceedings of the 2016 Genetic and Evolutionary Computation Conference, Denver (CO), pp. 317–324.

[8] El Yafrani M, Chand S, Neumann A, Ahiod B & Wagner M, 2017, *Multi-objectiveness in the single-objective traveling thief problem*, Proceedings of the 2017 Genetic and Evolutionary Computation Conference Companion, Berlin, pp. 107–108.

[9] El Yafrani M & Ahiod B, 2017, *A local search based approach for solving the travelling thief problem: The pros and cons*, Applied Soft Computing, **52**, pp. 795–804.

[10] El Yafrani M, Martins MSR, Krari ME, Wagner M, Delgado MRBS, Ahiod B, & Lüders R, 2018, *A fitness landscape analysis of the travelling thief problem*, Proceedings of the 2018 Genetic and Evolutionary Computation Conference, Kyoto, pp. 277–284.

[11] El Yafrani M, Martins MSR, Wagner M, Ahiod B, Delgado MRBS & Lüders R, 2018, *A hyperheuristic approach based on low-level heuristics for the travelling thief problem*, Genetic Programming and Evolvable Machines, **19(1-2)**, pp. 121–150.

[12] Faulkner M, Polyakovskiy S, Schultz T & Wagner M, 2015, *Approximate approaches to the traveling thief problem*, Proceedings of the 2015 Genetic and Evolutionary Computation Conference, Madrid, pp. 385–392.

[13] Gupta BC & Prakash VP, 2015, *Greedy heuristics for the travelling thief problem*, Proceedings of 39th National Systems Conference, Noida, pp. 1–5.

[14] Kerschke P, Hoos HH, Neumann F & Trautmann H, 2019, *Automated algorithm selection: Survey and perspectives*, Evolutionary Computation Journal, **27(1)**, pp. 3–45.

[15] Malan KM & Engelbrecht AP, 2013, *A survey of techniques for characterising fitness landscapes and some possible ways forward*, Information Sciences, **241**, pp. 148–163.

[16] Martins MSR, El Yafrani M, Delgado MRBS, Wagner M, Ahiod B & Lüders R, 2017, *HSEDA: a heuristic selection approach based on estimation of distribution algorithm for the travelling thief problem*, Proceedings of the 2017 Genetic and Evolutionary Computation Conference, Berlin, pp. 361–368.

[17] Mei Y, Li X & Yao X, 2014, *Improving efficiency of heuristics for the large scale traveling thief problem*, Proceedings of SEAL 2014: Simulated Evolution And Learning, Dunedin, pp. 631–643.

[18] Mei Y, Li X, Salim F & Yao X, 2015, *Heuristic evolution with genetic programming for traveling thief problem*, Proceedings of the 2015 IEEE Congress on Evolutionary Computation, Sendai, pp. 2753–2760.

[19] Nieto-Fuentes R, Segura C & Valdez SI, 2018, *A guided local search approach for the travelling thief problem*, Proceedings of the 2018 IEEE Congress on Evolutionary Computation, Rio de Janeiro, pp. 1–8.

[20] Ochoa G, Verel S, Daolio F & Tomassini M, 2014, *Local optima networks: A new model of combinatorial fitness landscapes*, pp. 233–262 in Richter H & Engelbrecht A (Eds.), *Recent advances in the theory and application of fitness landscapes*, Springer, Heidelberg.

[21] Polyakovskiy S, Bonyadi MR, Wagner M, Michalewicz Z & Neumann F, 2014, *A comprehensive benchmark set and heuristics for the traveling thief problem*, Proceedings of the 2014 Genetic and Evolutionary Computation Conference, Vancouver, pp. 477–484.

[22] Polyakovskiy S & Neumann F, 2017, *The packing while traveling problem*, European Journal of Operational Research, **258(2)**, pp. 424–439.

[23] Sörensen K, 2015, *Metaheuristics – the metaphor exposed*, International Transactions in Operational Research, **22**, pp. 3–18.

[24] Wagner M, 2016, *Stealing items more efficiently with ants: A swarm intelligence approach to the travelling thief problem*, Proceedings of the 10^{th} International Conference on Swarm Intelligence, Brussels, pp. 273–281.

[25] Wagner M, Lindauer M, Misir M, Nallaperuma S & Hutter F, 2018, *A case study of algorithm selection for the traveling thief problem*, Journal of Heuristics, **24(3)**, pp. 295–320.

[26] Wu J, Polyakovskiy S & Neumann F, 2016, *On the impact of the renting rate for the unconstrained nonlinear knapsack problem*, Proceedings of the 2016 Genetic and Evolutionary Computation Conference, Denver (CO), pp. 413–419.

[27] Wu J, Wagner M, Polyakovskiy S & Neumann F, 2017, *Exact approaches for the travelling thief problem*, Proceedings of Simulated Evolution and Learning, Shenzen, pp. 110–121.

[28] Wu J, Polyakovskiy S, Wagner M & Neumann F, 2018, *Evolutionary computation plus dynamic programming for the bi-objective travelling thief problem*, Proceedings of the 2018 Genetic and Evolutionary Computation Conference, Kyoto, pp. 777–784.