The Capacitated Team Orienteering Problem

Aldy Gunawan[1], Kien Ming Ng[2], Vincent F. Yu[3], Gordy Adiprasetyo[2] and Hoong Chuin Lau[1]
[1]School of Information Systems
Singapore Management University
Singapore, 178902
aldygunawan@smu.edu.sg, hclau@smu.edu.sg

[2]Department of Industrial Systems Engineering & Management
National University of Singapore
Singapore, 117576
isenkm@nus.edu.sg, gordy@u.nus.edu

[3]Department of Industrial Management
National Taiwan University of Science and Technology
Taipei 106, Taiwan
vincent@mail.ntust.edu.tw

Abstract

This paper focuses on a recent variant of the Orienteering Problem (OP), namely the Capacitated Team OP (CTOP) which arises in the logistics industry. In this problem, each node is associated with a demand that needs to be satisfied and a score that need to be collected. Given a set of homogeneous fleet of vehicles, the objective is to find a path for each vehicle in order to maximize the total collected score, without violating the capacity and time budget. We propose an Iterated Local Search (ILS) algorithm for solving the CTOP. Two strategies, either accepting a new solution as long as it improves the quality of the solutions or accepting a new solution as long as there is no constraint violation, are implemented. For solving difficult instances, we simplify the move operator of local search in order to reduce the computational time. Instead of exploring all possible nodes in all paths to be moved, we only focus on nodes in the path with the least remaining amount of time. Computational experiments on benchmark instances illustrate that the algorithm can generate solutions within 1% and 4% from the current best known solution for small and large instances, respectively.

Keywords
Orienteering Problem, Iterated Local Search, Capacitated Team Orienteering Problem

1. Introduction

Orienteering Problem (OP) is an NP-hard routing problem in which the objective is to find a subset of nodes to be visited by a vehicle, and in what order, such that the total score collected from the visited nodes is maximized while complying to limited the time budget constraint (Vansteenwegen et al. 2011). The OP is first introduced by Tsiligirides (1984). According to the latest survey paper on OP by Gunawan et al. (2016), numerous extended variants of OP, each with its own unique constraints, have been researched on extensively in the past few years. This work focuses on a particular variant of OP called the Capacitated Team Orienteering Problem (CTOP).

CTOP is highly relevant to real-world problems, especially in the logistics industry. In this variant of OP, each node is associated with a demand and a profit. Given a set of homogeneous fleet of vehicles, the main objective is to determine a route for each available vehicle that maximizes the total score (profit) while complying to capacity and time budget constraints that each vehicle has (Archetti et al. 2009). It is assumed that each node can only be visited once by one vehicle. All vehicles start and end at the same node, which is the depot. In the context of the OP, each

vehicle also refers to a particular path. Team OP is the extension of the OP by considering multiple vehicles or paths. Since the OP has been proven to be NP-hard (Golden et al. 1987), it is unlikely that the CTOP can also be solved optimally within polynomial time. Therefore, it is interesting to propose fast algorithms to solve the problem, especially when dealing with larger instances.

Archetti et al. (2013a) introduce a branch-and-price algorithm in order to deal with benchmark instances. Several unsolved benchmark instances can be optimally solved. Tarantilis et al. (2013) propose a hierarchical bi-level search framework, namely a Bi-level Filter-and-Fan method. Experimental results show the efficiency and effectiveness of a Bi-level Filter-and-Fan method in solving the benchmark instances. Gunawan et al. (2016) summarize different variants of the CTOP, such as the CTOP with Incomplete Service (Archetti et al. 2013b), the Split Delivery CTOP (Archetti 2014) and the Split Delivery CTOP with Minimum Delivery Amounts (Wang et al. 2014).

In this paper, we propose an algorithm which is based on the Iterated Local Search (ILS) algorithm for solving the CTOP. Iterated Local Search (Luo et al. 2013) is a simple but effective metaheuristic. ILS generates a sequence of solutions generated by an embedded heuristic, leading to far better results than if one were to use repeated random trials of that heuristic. We consider four basic modules of ILS: generating the initial solution, local search, the perturbation strategy and the acceptance criterion. ILS has been used for solving other variants of the OP, such as OP with Time Windows (OPTW) (Gunawan et al. 2015), Team OP with Time Windows (TOPTW) (Vansteenwegen et al. 2009, Gunawan et al. 2017) and OP with soft Time Windows (OPSTW) (Aghezzaf and Fahim 2016). The proposed ILS uses different types of strategies for implementing operators of local search for different types of benchmark CTOP instances (Archetti et al. 2009). Benchmark instances can be accessed from http://tarantilis.dmst.aueb.gr/docs/.

There are two sets of benchmark instances, Archetti's instances (Archetti et al. 2009) and Tarantilis's instances (Tarantilis et al. 2013). For Archetti's instances, we implement two different strategies when accepting the operators, either the operator is able to improve the quality of the solution or it only increases the remaining time budget that can be used for other nodes. For Tarantilis's instances, we simplify the move operator. Instead of exploring all possible moves from all nodes, we only consider of moving nodes in the path with the least remaining time budget. The results are also compared with the current best known solutions (Tarantilis et al. 2013). Computational experiments illustrate the efficiency and effectiveness of the proposed approach. The paper is organized as follows. Section 2 presents the CTOP mathematical model. In Section 3, we describe the proposed ILS algorithm. Section 4 is devoted to the experimental results and analysis. Finally, Section 5 concludes and provides some ideas for future works.

2. The Capacitated Team Orienteering Problem

The latest CTOP model was presented by Tarantilis et al. (2012). We modify the model by replacing the non-linear subtour elimination constraints. Let $G = (N, A)$ be an undirected graph with the set A of arcs and the set N of nodes. $N = \{0, 1, \ldots, |N|\}$ where 0 denotes the depot. N_c represents a set of nodes that can be served, i.e. node 1 to n or $N_c = N \backslash \{0\}$.

Parameters of the model:
d_i : known demand for node i, $i \in N\backslash\{0\}$
p_i: profit associated with node i
c_{ij}: traveling cost incurred for traveling from node i to j
K: a number of vehicles used to serve the nodes
Q_k: capacity of vehicle k
T_{max}: maximum allowable operation time budget of a vehicle
t_{ij} : time needed to travel from node i to j
s_i : service time needed on node i

Decision variables of the model:

$$x_{ij}^k : \begin{cases} 1 \text{ if vehicle } k \text{ travels from node } i \text{ to } j \\ 0 \text{ otherwise} \end{cases}$$

$$y_i^k : \begin{cases} 1 \text{ if node } i \text{ is served by vehicle } k \\ 0 \text{ otherwise} \end{cases}$$

u_i^k the position of visited node i by vehicle k in the path

The mathematical programming model:

$$\max \sum_{i \in N_c} \sum_{k \in K} p_i y_i^k \tag{1}$$

$$s.t.$$

$$\sum_{k \in K} \sum_{j \in N_c} x_{0j}^k = \sum_{k \in K} \sum_{i \in N_c} x_{i0}^k = K \tag{2}$$

$$\sum_{k \in K} y_i^k \leq 1 \qquad \forall i \in N_c \tag{3}$$

$$\sum_{j \in N_c} x_{ij}^k = \sum_{j \in N_c} x_{ji}^k = y_i^k \qquad \forall i \in N_c, \forall k \in K \tag{4}$$

$$\sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k + \sum_{i \in N_c} s_i y_i^k \leq T_{max} \qquad \forall k \in K \tag{5}$$

$$\sum_{i \in N_c} d_i y_i^k \leq Q_k \qquad \forall k \in K \tag{6}$$

$$1 \leq u_{ik} \leq N - 1 \qquad \forall i \in N, \forall k \in K \tag{7}$$

$$u_i^k - u_j^k + 1 \leq (N-2)\left(1 - x_{ij}^k\right) \qquad \forall i, j \in N, \forall k \in K \tag{8}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i, j \in N, \forall k \in K \tag{9}$$

$$y_i^k \in \{0, 1\} \qquad \forall i \in N, \forall k \in K \tag{10}$$

The Objective function (1) maximizes the collected profit. Constraints (2) ensure that K vehicles leave and enter the depot. Constraints (3) ensure that all nodes are served by at most 1 vehicle, this prevents split-delivery cases. Constraint (4) ensures the route connectivity, e.g. same vehicle enters and leaves a given node. Constraints (5) limits that operation of a certain vehicle k is within time budget. Constraints (6) ensures that demand of nodes in a trip by certain vehicle k does not exceed its capacity. Constraints (7) and (8) are sub-tour elimination constraints. Constraints (9) and (10) impose binary restrictions on variables x_{ij}^k and y_i^k.

3. Iterated Local Search

This section provides the description of our proposed algorithm, Iterated Local Search (ILS). We first introduce a construction heuristic for generating an initial solution. The initial solution is further improved by ILS.

3.1 Construction Heuristic

The initial solution is generated by an insertion heuristic which is adopted from the work of Luo et al. (2013). Basically, nodes are sorted based on a certain valuation, and one by one will be inserted into the available paths/vehicles until no further insertion is feasible. In this paper, we define and rank nodes based on two different criteria: equations 11 and 12 for Archetti's and Tarantilis's instances, respectively.

$$ratio_i = \frac{p_i}{d_i} \tag{11}$$

$$ratio_i = \frac{(p_i)^2}{c_{i0}+s_i} \tag{12}$$

3.2 Iterated Local Search (ILS)

In order to improve the initial solution S_0, we implement a metaheuristic based on Iterated Local Search (ILS). Let S^* be the best found solution so far at a particular iteration. For the first iteration of ILS, S^* equals to S_0.

The main idea of ILS is to explore the solution space by generating and evaluating the neighbors of S_0. LOCALSEARCH1 is applied in order to generate the best neighborhood. Here, we run different operators consecutively. More details will be explained below. The first improving neighbor replaces S_0. If a stagnation condition is met, a perturbation strategy on S_0 is then applied. The idea of this strategy is taken from the one proposed by Vansteenwegen et al. (2009). We implement two different LOCALSEARCH: LOCALSEARCH1 and LOCALSEARCH2. First, LOCALSEARCH1 is applied to S_0 as many times as possible, limited by the parameter LOWERLIMIT, followed by applying LOCALSEARCH2 for (MAXITER - LOWERLIMIT) times. The outline of the ILS algorithm is presented in Algorithm 1 (Figure 1 - left side). There are two different benchmark instances, namely Archetti's instances and Tarantilis's instances. Note that this algorithm is applied to Archetti's instances. We propose another algorithm which is used to solve Tarantilis's instances, as shown in Algorithm 2 (Figure 1- right side). More details about both instances will be explained in Section 4.1.

Algorithm 1 ILS

```
S0 ← INITIAL SOLUTION
S0 ← LOCALSEARCH1
S* ← S0
NOIMPR ← 0
while ITER < MAXITER do
    S0 ← PERTURBATION
    if ITER ≤ LOWERLIMIT then
        S0 ← LOCALSEARCH2
    else
        S0 ← LOCALSEARCH1
    end if
    if S0 better than S* then
        S* ← S0
        NOIMPR ← 0
    else
        NOIMPR ← NOIMPR + 1
    end if
    if (NOIMPR = MAXNOIMPROV) then
        S0 ← S*
    end if
end while
return S*
```

Algorithm 2 ILS

```
S0 ← INITIAL SOLUTION
S0 ← LOCALSEARCH3
S* ← S0
NOIMPR ← 0
while ITER < MAXITER do
    S0 ← PERTURBATION
    S0 ← LOCALSEARCH3
    if S0 better than S* then
        S* ← S0
        NOIMPR ← 0
    else
        NOIMPR ← NOIMPR + 1
    end if
    if (NOIMPR = MAXNOIMPROV) then
        S0 ← S*
    end if
end while
return S*
```

Figure 1. Algorithm 1 and Algorithm 2

Both LOCALSEARCH1 and LOCALSEARCH2 use a set of operators which are run consecutively, as listed in Table 1. SWAP1 examines all possible combinations of selecting two different nodes within one vehicle with the least remaining time budget. It is only executed only if it is able to increase the remaining time budget and there is no constraint violation. SWAP2 is similar to SWAP1 with the difference of selecting two vehicles. In 2-OPT, we select one vehicle with the lowest remaining time budget. All possible combinations of selecting two different nodes are enumerated and the sequence of scheduled nodes is reversed as long as there is no constraint violation and it is able to increase the remaining time budget of the selected vehicle, as shown in Figure 2.

MOVE is started by selected the first scheduled node from the first vehicle. We examine all possible insertion of other nodes from other vehicles. As long as it is able to increase the total remaining time budget of both vehicles, the movement is accepted. Figure 3 illustrates an example of MOVE. The last two operators, INSERT and REPLACE, focus

on increasing the objective function value – the total collected score. INSERT is done by considering unscheduled nodes that can be inserted to any vehicle. Unscheduled nodes are sorted in descending order of their ratios (Equation (1)). The insertion is accepted to the position that gives us the least additional total travel time. REPLACE is started by selecting a vehicle with the highest remaining time budget, followed by selecting one unscheduled node with the highest score. The selected node will replace a node with a lower score.

Table 1. LOCALSEARCH operations

Operations	Descriptions
SWAP1	Exchange two nodes within one vehicle
SWAP2	Exchange two nodes between two vehicles
2-OPT	Reverse the sequence of certain nodes within one vehicle
MOVE	Move one node from one vehicle to another vehicle
INSERT	Insert nodes into a vehicle
REPLACE	Replace one scheduled node with one unscheduled node

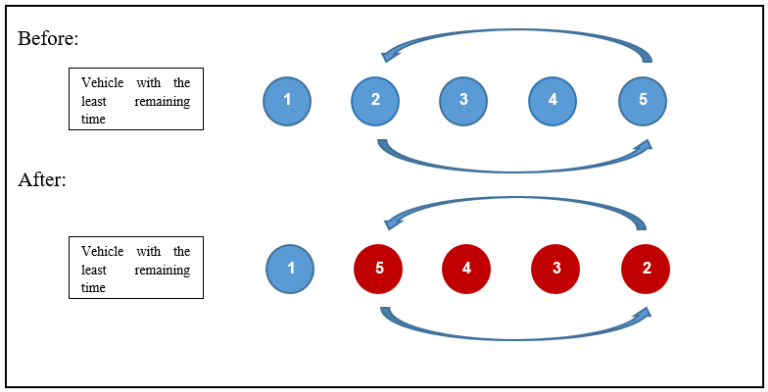

Figure 2. 2-OPT

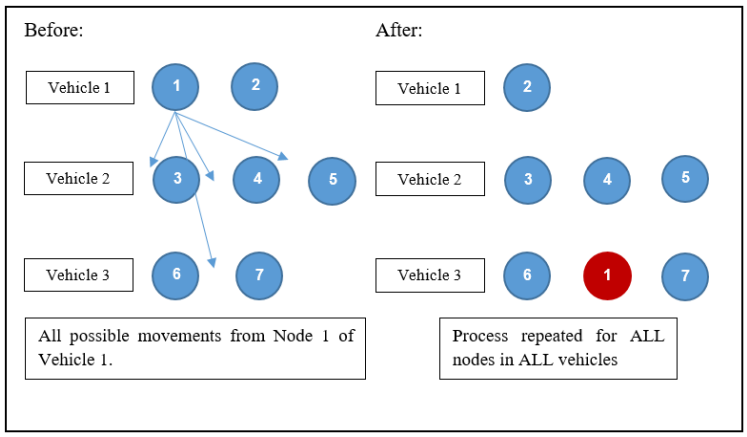

Figure 3. MOVE

The main difference between both LOCALSEARCH1 and LOCALSEARCH2 lies on applying SWAP2 and INSERT. In LOCALSEARCH2, both will be accepted if they do not violate the constraints and able to improve the quality of solutions, e.g. increase the remaining time budget or increase the total score. In LOCALSEARCH1, as long as there is no constraint violation, both operators can be accepted. In other words, LOCALSEARCH1 is a relaxed version of LOCALSEARCH2.

We use SHAKE as the perturbation strategy. All nodes in all routes (except the depot or node 0) are considered as one sequence. We remove one or more nodes, based on two parameters: POS and LENGTH. POS determines the position

of the first node to be removed while LENGTH determines how many consecutive nodes to be removed. At each iteration, POS is increased by the value of LENGTH, while LENGTH itself is increased by one. If LENGTH reaches a certain value which is a predefined value, it will be set to one. If the value of POS is more than the number of nodes, the value is deducted by the number of nodes in the sequence.

Algorithm 2 is proposed for solving Tarantilis's instances. In general, the main idea is the same with the one of Algorithm 1. Here, we apply another version of LOCALSEARCH, namely LOCALSEARCH3. The idea is to simplify MOVE operator since it takes a large computational time. In LOCALSEARCH1 and LOCALSEARCH2, it attempts to move every single node to every single other possible feasible position. For Tarantilis's instances, we only move nodes in the vehicle with the least remaining time budget to other possible positions in other vehicles.

Acceptance criterion is important to find a good balance between diversification and intensification. Although some ILS operators are different for solving both benchmark instances, same acceptance criterion is applied to them. We implement the idea of "random walk" acceptance criterion that ensures a good balance between diversification and intensification of the search. In this method, search would continue from the current found solution until a certain number of non-improving solution MAXNOIMPROV is reached, hence leading to more diversification. After that, only solutions that are better than the best found solution are kept, hence leading to intensification.

4. Computational Experiments

In Section 4.1, we first explain how we setup the experiments. We also present the benchmark CTOP instances and their best known solutions for comparison purpose. In Section 3.2, we describe the parameter values used and present the results obtained.

4.1 Computational Setup and Benchmark Instances

The algorithm is coded in C++ and all experiments are executed on an Intel Core i7-4790 3.60 GHz processor CPU with 32 GB of RAM running on Microsoft Windows operating system. Two sets of benchmark instances, Archetti's instances (Archetti et al. 2009) and Tarantilis's instances (Tarantilis et al. 2013), are used.

As shown in Table 2, Archetti's instances are further categorized into three subsets. The first subset contains the instances generated from the Capacitated VRP instances (Christofides et al. 1979) that allow all customers or nodes to be served. The second subset modifies vehicle sizes, capacity and route duration limits. The third subset alters the vehicle size. Tarantilis's instances (as shown in Table 3), which are adopted from the large scale Period VRP instances of Pirkwieser and Raidl (Pirkwieser et al. 2010), are divided into three subsets with similar characteristics. Benchmark CTOP instances are summarized in Table 2.

Table 2. Archetti's Instances

Subset	Number of instances	Number of nodes	Number of vehicles	Time budget
1	10	51 – 200	10 – 20	160 – 1040
2	90	51 – 200	2 – 4	50 – 100
3	30	51 – 200	2 – 4	140 – 200

Table 3. Tarantilis's Instances

Subset	Number of instances	Number of nodes	Number of vehicles	Time budget
1	10	337 – 577	14 – 24	660 – 720
2	90	337 – 577	6 – 8	100 – 400
3	30	337 – 577	6 – 8	660 – 720

4.2 Computational Results

The parameter values are defined as follows: MAXITER = 1000, LOWERLIMIT = 250 and MAXNOIMPROV = 10. Tables 1 summarizes results obtained by ILS for both sets of instances. Each instance is run five times. The gap values are calculated by comparing our results with the best known solutions. Table 4 summarizes the results of Archetti's instances. It does not improve any current best known solution, partly because most of the solutions are proven to be optimal. The gap values are calculated by comparing our results with the best known solutions. The gap values are less than 0.50 %. In Subset 1 of Archetti et al. instances, both can obtain the best known solutions except for one instance. For Subset 2 with 90 instances, ILS(R1) obtains 70 best known solutions, while ILS(R2) obtains 80 best known solutions, though at the expense of computation time. Finally, ILS can get 24 best known solutions for Subset 3. For the Archetti et al. instances, we conclude that the performance of ILS is thus promising.

Table 4. Archetti's Instances results

Instances	Average Objective Function Value		Gap (%)	Average CPU time (seconds)	
	Best Known	ILS		Best Known	ILS
Subset 1	1814.20	1814.00	0.01	0.22	0.17
Subset 2	295.21	294.22	0.30	19.52	15.39
Subset 3	728.40	728.40	0.00	3.59	3.11

For Tarantilis et al. instances, ILS also performs well at the expense of computation time. But the results are still worse than those of best known solutions, with the gap values being up to 3.09%. However, ILS is able to improve one best known solution of instance 14 of Subset 2, from 509 to 602. Our ILS has some limitations especially pertaining to the large computation time.

Table 5. Tarantilis's Instances results

Instances	Average Objective Function Value		Gap (%)	Average CPU time (seconds)	
	Best Known	ILS		Best Known	ILS
Subset 1	5647.20	5643.50	0.06	63.26	62.11
Subset 2	1268.87	1229.63	3.09	3061.30	3102.69
Subset 3	3032.50	2985.77	1.54	10978.65	10711.66

5. Conclusions

In this paper, we focus on solving the Capacitated Team Orienteering Problem (CTOP). The CTOP is a variant of the TOP where each node is associated with a demand and a score. The main objective is to determine a path for each available vehicle in order to maximize the total score, without violating the capacity and time budget of each vehicle. Iterated Local Search (ILS) metaheuristic is proposed in order to solve the benchmark instances of CTOP.

The proposed ILS uses different types of strategies for implementing operators of local search for different types of benchmark CTOP instances. For Archetti's instances, we implement two different strategies when accepting the operators, either the operator is able to improve the quality of the solution or it only increases the remaining time budget that can be used for other nodes. For Tarantilis's instances, the move operator is modified. Instead of exploring all possible moves from all nodes, we only consider of moving nodes in the path with the least remaining time budget.

The performance of the proposed algorithm was promising. It is able to generate solutions within 1% from the current best known solution for small instances, and within 4% for large instances, giving an average of 2.20% for all instances. Since the chosen acceptance criterion has a critical influence on the performance of the proposed algorithm, a possible improvement of the algorithm could involve also considering worst solutions during the search which is used in Simulated Annealing. Some possible extensions of local search operators would be explore as future works. Since the OP and its variants, such as the Arc Orienteering Problem and the Team OP with Soft Time Windows, have attracted more attention in recent years, the proposed algorithm can be potentially tailored to solve them as well.

References

Aghezzaf, B., and Fahim, H., Iterated local search algorithm for solving the orienteering problem with soft time windows, *Springerplus* vol. 5, no. 1, pp. 1781-1781, 2016.

Archetti, C., Bianchessi, N., and Speranza, M.G., The capacitated team orienteering problem with incomplete service, *Optimization Letters*, vol. 7, no. 7, pp. 1405-417, 2013.

Archetti, C., Bianchessi, N., and Speranza, M.G., Optimal solutions for routing problems with profits, *Discrete Applied Mathematics*, vol. 161, no. 4-5, pp. 547–557, 2013.

Archetti, C., Bianchessi, N., and Speranza, M.G., The split delivery capacitated team orienteering problem, *Networks*, vol. 63, no. 1, pp. 16-33, 2014.

Archetti, C., Feillet, D., Hertz, A., and Speranza, M.G., The capacitated team orienteering and profitable tour problems, *Journal of the Operational Research Society*, vol. 60, no. 6, pp. 831-842, 2009.

Golden, B., Levy, L., and Vohra, R., The orienteering problem, *Naval Research Logistics*, vol. 34, no. 3, pp. 307-318, 1987.

Gunawan, A., Lau, H.C., and Lu, K., An iterated local search algorithm for solving the orienteering problem with time windows. In: G. Ochoa, F. Chicano (eds.), *Proceedings of the 15th European Conference on Evolutionary Computation in Combinatorial Optimisation*, Copenhagen, Denmark, April 8-10, 2015, Lecture Notes in Computer Science, vol. 9026, pp. 61–73, Springer Verlag, Berlin, Germany.

Gunawan, A., Lau, H.C., and Vansteenwegen, P., Orienteering problem: a survey of recent variants, solution approaches and applications, *European Journal of Operational Research*, vol. 255, no. 2, pp. 315-332, 2016.

Gunawan, A., Lau, H.C., Vansteenwegen, P., and Lu, K., Well-tuned algorithms for the team orienteering problem with time windows, *Journal of the Operational Research Society*, vol. 68, no. 8, pp. 861-876, 2017.

Lourenco, H., Martin, O., and Stutzle, T., Iterated local search. In: *Handbook of metaheuristics*, pp. 320–353. Springer, 2003.

Luo, Z., Cheang, B., Lim, A., and Zhu, W., An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem, *European Journal of Operational Research*, vol. 229, no. 3, pp. 673 - 682, 2013.

Tarantilis, C.D., Stavropoulou, F., and Repoussis, P.P., The capacitated team orienteering problem: a bilevel filter-and-fan method, *European Journal of Operational Research*, vol. 224, no. 1, pp. 65-78, 2013.

Tsiligirides, T., Heuristic methods applied to orienteering, *Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797-809, 1984.

Vansteenwegen, P., Souffriau, W., and Van Oudheusden, D., The orienteering problem: a survey, *European Journal of Operational Research*, vol. 209, no. 1, pp. 1-10, 2011.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D., Iterated local search for the team orienteering problem with time windows, *Computers and Operations Research*, vol. 36, no. 12, pp. 3281-3290, 2009.

Wang, X., Golden, B.L., and Gulczynski, D., A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery amounts, *Optimization Letters*, vol. 8, no. 8, pp. 2349-2356, 2014.

Biography / Biographies

Aldy GUNAWAN is a visiting assistant professor of Information Systems at the Singapore Management University. He works in between the areas of Operations Research and Artificial Intelligence. He received his Ph.D in Industrial and Systems Engineering from the National University of Singapore. His main research interests include operations research, algorithm design and data analytics which relate to metaheuristics, algorithm configuration, design of experiments, combinatorial optimization and automated planning/scheduling. His past studies have been published in the top conferences and journals in Operations Research, such as AAMAS, ECAI, JORS, EJOR, Computers and IE and AOR. He is serving as the program co-chair of the 8st Multidisciplinary International Scheduling Conference in 2017. He has been serving as a committee member of the Operational Research Society of Singapore.

Kien Ming NG is currently an Associate Professor at the Department of Industrial Systems Engineering & Management, National University of Singapore (NUS). He received his B.Sc. (Hons.) degree in Mathematics from NUS in 1994, and he subsequently obtained an M.S. degree in Engineering-Economic Systems & Operations Research in 1999 and a Ph.D. degree in Management Science and Engineering (Operations Research) in 2002, both from Stanford University. His research interests are in discrete and nonlinear optimization problems arising from the industry and military. Some of the problems that he has worked on include defence logistics, routing and scheduling problems, and his focus is on developing efficient optimization algorithms and techniques to solve such problems. He is the Managing Editor of the Asia-Pacific Journal of Operational Research.

Vincent F. YU is a Professor of Industrial Management at the National Taiwan University of Science and Technology (NTUST). Dr. Yu receives his Ph.D. degree in Industrial & Operations Engineering from the University of Michigan, Ann Arbor. His research interests include operations research, soft computing, and logistics/supply chain management. Dr. Yu is currently the president of POMS Taiwan. He is also a board member of the International Federation of Logistics and SCM Systems and the Operations Research Society of Taiwan. Dr. Yu was a Decision Technology Manager of a large trucking company in the US for about five years. He has collaborated with many leading companies in Taiwan such as DHL Supply Chain, Foxconn, and Kerry TJ Logistics. Dr. Yu has received NTUST Excellent Research Award three times (2014-2016, 2016-2018, 2018-2020).

Gordy ADIPRASETYO is currently working at a logistic company in Singapore. He obtained his bachelor degree from the Department of Industrial Systems Engineering & Management, National University of Singapore (NUS). His research interests are related include operations research and logistics.

Hoong Chuin LAU is a Professor of Information Systems and Director of the Fujitsu-SMU Urban Computing and Engineering Corporate Lab at the Singapore Management University (SMU). Working at the interface of Operations Research and Artificial Intelligence, he is interested in combining data analytics and optimization for decision-making in business. He was awarded the Lee Kwan Yew Fellowship for research excellence in 2008. He currently serves on the editorial board of the IEEE Transactions on Automation Science and Engineering, and the Web Intelligence Journal. Throughout his 18 years of full-time university lecturing experience, he taught a number of undergraduate and graduate courses, including Design and Analysis of Algorithms, Combinatorial Graph Algorithms, Computer as an Analysis Tool, Computational Thinking, Decision Analytics and Optimization, Advanced Topics in Intelligent Systems, and Enterprise Analytics for Decision Support.