The Capacitated Team Orienteering Problem: a hybrid Simulated Annealing and Iterated Local Search Approach

Aldy GUNAWAN[1], Jiahui ZHU[2], Kien Ming NG[2]

aldygunawan@smu.edu.sg, e0032017@u.nus.edu, isenkm@nus.edu.sg
[1]School of Computing and Information Systems, Singapore Management University, Singapore
[2]Industrial Systems Engineering and Management Department, National University of Singapore, Singapore

1. Introduction

Orienteering Problem (OP) is an NP-hard vehicle routing problem that combines two classical combinatorial optimization problems, the Traveling Salesman Problem (TSP) and the Knapsack Problem (Vansteenwegen et al., 2011). The objective of the problem is to select the most profitable combination of customers from a list of potential customers given that the selected customers do not violate the time constraints (Gunawan et al. 2016). OP was first introduced by Golden et al. (1987) and since then, the problem has received a considerable amount of attention by researchers in the past few decades. The survey papers published by Vansteenwegen et al. (2011) and Gunawan et al. (2016) provided an extensive summary of the range of research works done on OP and its variant from the time of introduction of the OP to as recent as 2016.

Team Orienteering Problem (TOP) is one main variant of the original OP. The objective of TOP is to select the most profitable combination of customers for a fleet of vehicles from a list of potential customers, given that the selected customers do not violate the time constraints. The concept of TOP was first introduced by Butt and Cavalier (1994), who named it the Multiple Tour Maximum Collection Problem, and Chao et al. (1996) came up with the term TOP, which is widely used by researchers around the world currently.

In this paper, we focus on another variant of the OP, namely the Capacitated Team Orienteering Problem. CTOP considers each vehicle to have a limited capacity and each customer is associated with a demand for capacity. The objective of the CTOP is to optimize the profit generated for the fleet of vehicles by choosing customers with the consideration of each customer's demand and profit. The concept was first introduced by Archetti et al. (2009) and it has great practical usage in the logistics industry. Many of the logistics companies are now facing international competition, which forces them to cut costs in order to survive in this competitive market. One of the ways to cut cost is to maximise the number of goods each vehicle can hold. However, the survey papers conducted by Vansteenwegen et al. (2011) and Gunawan et al. (2016) showed that only few research works have been done on this particular topic. Therefore, this paper aims to contribute to the research on CTOP by providing a heuristic approach that can generate a good quality solution efficiently.

Archetti et al. (2009) proposed an exact approach and three heuristic approaches to solve this problem. The exact approach is based on the branch-and-price algorithm. The three heuristic approaches comprise of Variable Neighbourhood Search (VNS) algorithm and Tabu search algorithms. A novel Bi-level Filter-and-Fan method is proposed by Tarantilis

(2013). The proposed method consists of three components: a greedy parallel insertion-based construction heuristic to generate an initial feasible solution; a new Tabu Search based local search to identify a local optimal solution, and a novel filter-and-fan search to explore larger combined neighbourhoods and generate multiple search trajectories in an effort to overcome local optimality. The algorithm was able to match and improve some of the best reported results with competitive computational time. Luo et al. (2013) introduced an approach using an adaptive Ejection Pool (EP) with toggle-rule diversification. The proposed algorithm maintains the current solution in two parts: the first part consists of the selected customers and the second part consists of all the potential customers that are currently not selected. The potential customers are arranged based on their value, with the first one being the most valuable customer. Priority is given to the first potential customer if a replacement is to be made between the selected customer and potential customers. Another heuristic algorithm proposed is the Adaptive Iterative Destruction/Construction Heuristic (AIDCH) (Ben-Said et al., 2016). This algorithm starts with an adaptive construction phase based on the Best Insertion Algorithm, followed by an adaptive diversification phase with local search methodologies. A recent research work on CTOP was published by Gunawan et al. (2019), which proposed a heuristic algorithm based on the Iterated Local Search (ILS). This algorithm comprises of 4 main modules: initial solution, local search, perturbation and acceptance. The algorithm produced promising results as compared to other heuristic algorithms proposed previously.

2. Proposed Algorithm

We propose a heuristic which is inspired by the Simulated Annealing and Iterated Local Search (SAILS) algorithm (Gunawan et al., 2017). The entire algorithm is illustrated in Figure 1. The Simulated Annealing (SA) portion of the proposed algorithm is further modified by adapting the SA process proposed by Lin and Yu (2015) with a few minor adjustments. The first adjustment is done after generating the initial solution, with the addition of local search first before entering the looping process. The addition of an extra local search right after initial solution improves the efficiency of the algorithm by starting off the looping process with a much better initial condition (Gunawan et al., 2019). The second adjustment is done at the generation of the solution based on a previous solution through exploring the neighbourhood. The original SA process introduced by Lin and Yu (2015) has 3 types of iterators for exploring the neighbourhood, namely `Swap`, `Insert` and `Reverse`, each with a probability of 1/3 being chosen. Since Insertion will be performed exhaustively in the local search step, `Insert` is removed from this step for the proposed algorithm. More details of the operators will be explained below.

The Random Walk acceptance criterion (Vansteenwegen, 2014) is adapted for the proposed algorithm. This acceptance criterion provides a good balance between intensification and diversification when searching for solutions. Local search operators adapted from ILS (Gunawan et al., 2017) ensure that only solutions that are better than the current solution are kept, leading to search intensification. Random neighbourhood search with SA process allows the algorithm to explore neighbouring solution and have chance to escape local optima, leading to search diversification.

The algorithm starts off by first generating the initial solution X. The current temperature T is also set to the initial temperature $Tmax$. The algorithm then performs a round of local search on X to improve the initial solution, and the best-found solution $F(Z)$ is updated to $F(X)$. Upon completion of the neighbourhood search, a new solution, $Y,$ is found and the objective value of Y, $F(Y)$, is compared against the objective value of X, $F(X)$. If $F(Y)$ is better or equal to $F(X)$, then X is replaced by Y. However, if $F(Y)$ is worse than $F(X)$, another random number r between 0 and 1 is generated and compared against $e^{\frac{F(Y)-F(X)}{T}}$, where X is replaced by Y if $r < e^{\frac{F(Y)-F(X)}{T}}$. Furthermore, if $F(Y)$ is better than $F(Z)$, Z is also replaced by Y. The neighbourhood search repeats itself until $I = Imax$.

Next, the temperature T is reduced with the formula $T = T \times \alpha$, where α is the cooling ratio. Local search is performed on Z to further improve the solution. Now, the algorithm performs a check to see if $F(Z)$ is improved after the local search. If $F(Z)$ is improved, then $N = 0$, $I = 0$, and the algorithm begins another round of neighbourhood search with $X = Z$. If $F(Z)$ is not improved, the non-improved count, N, increases by one and is compared against $Nmax$. If $N < Nmax$, $I = 0$ and the algorithm begins another round of neighbourhood search with $X = Z$. The algorithm terminates when $N = Nmax$.

```
1: Generate Initial Solution
2: F(Z) ← Apply Local Search
3: Set I = 0, N = 0, T = Tmax, F(Z) = F(X*)
4: while (N < Nmax) do
5:    while (I < Imax) do
6:       F(Y) ← Neighborhood Search
7:       I++
8:       if (F(Y) > F(X*))
9:          F(X*) ← F(Y)
10:      else
11:         Generate r ~ U(0,1)
12:         if (r < exp((F(Y) – F(X*)) / T))
13:            F(X*) = F(Y)
14:         else
15:            return to step 5
16:         if (F(X*) > F(Z))
17:            F(Z) = F(X*)
18:         else
19:            Return to step 5
20:   T = T × α
21:   F(Z*) ← Local Search
22:   if (F(Z*) > F(Z))
23:      N = 0
24:   else
25:      N = N + 1

26:   I = 0; return to step 4
```

Figure 1. Proposed Algorithm

In order to generate the initial solution, we implement the simple insertion heuristic (Luo et al., 2013). This method first ranks all the customers based on their potential value. The ranked customers are then inserted one by one into the available vehicles starting from the highest value customers. The process stops when no more customers can be added into any of the vehicles. Since service time is not included into the calculation of the cost for the given benchmark instances, the value calculation for the given instance would be:

$$value = \frac{profit}{demand} \tag{1}$$

Six different local search operators, as shown in Table 1, were adapted from Gunawan et al. (2017). All the six operators are executed in sequence given in Table 1 for every call of the neighbourhood search. `Swap1` selects the vehicle with the least remaining travel time. All possible combinations of exchanging positions between two different customers are performed. `Swap1` is considered successfully executed only if the exchange increases the travel time of the chosen vehicle. `Swap2` is similar to `Swap1`, with the exception of selecting two vehicles with the least remaining travel times. All possible combinations of exchanging positions of customers between two vehicles are performed. `Swap2` is considered successfully executed only if the exchange increases the total travel time from both vehicles. Both operations terminate when all possible combinations of exchange are performed.

`2-Opt` is executed by first selecting the vehicle with the least remaining travel time. All possible combinations of selecting two different customers are performed, and the sequences of customers between the two selected customers are reversed. `Move` reallocates customers from one vehicle to another, with the objective of reducing total remaining time for all vehicles. The reallocation of customers should not violate any constraints and the operation terminates when all customers tried reallocating to all locations in all vehicles.

All the above-mentioned operators do not change the objective function value. They modify the current solution in order to increase the total remaining travel time. This may provide more opportunities for the next two operators, namely `Insert` and `Replace`, to improve the objective function value by adding or replacing customers from the group of unassigned customers. `Insert` rearranges all customers that are not assigned to the vehicles, based on their values in ascending order. Each unassigned customer would now be inserted into the vehicles without violating any constraints. If there are multiple insertion locations available for this unassigned customer, the location with the least addition of total traveling time will be chosen. `Replace` replaces customers assigned to vehicles with customers that have not been selected. The vehicle with the most remaining travel time is chosen for this operation. All unassigned customers are rearranged based on their values. The highest-valued unassigned customer is then selected to replace any customer in the vehicle that has a value lower than that of the unassigned customer without violating any constraints. If there are multiple potential customers in the vehicle that can be replaced by the unassigned customer, replace the customer that will result in the least addition of travel time. Each time after a successful replacement, rearrangement of the unassigned customers will be done and the highest-valued unassigned customer is chosen for the next replacement.

Table 1. Local Search operators

Operator	Definition
Swap1	Exchange two customers within one vehicle
Swap2	Exchange two customers between two vehicles
2-Opt	Reverse the sequence of certain customers within a vehicle
Move	Move one customer from one vehicle to another vehicle
Insert	Insert or add customers to a vehicle
Replace	Replace one customer in a vehicle with another customer that has not been selected

3. Computational Results

The proposed algorithm was implemented in C++ programming language and the computational runs were performed on a CPU with MacOS, Intel Core i5 2.7 GHz Dual-Core processor and 8 GB of RAM. Benchmark instances from Tarantilis (2012) were used to test the proposed algorithm.

The comparison of the results from the proposed solution against other well-known algorithms are presented below. The results of the-state-of-the-art algorithms were used to compare with the results generated from the proposed algorithm. Here, Variable Neighbourhood Search (VNS) and Tabu Search (feasible) (TSf) correspond to the algorithms proposed by Archetti et al. (2009); Bi-level Filter-and-Fan Fast (BiF&F-f) corresponds to the algorithm proposed by Tarantilis et al. (2012); Iterated Local Search (ILS) corresponds to the algorithm proposed by Gunawan et al. (2019). It is observed that keeping $Imax$ at 3000, $Nmax$ at 10, $Tmax$ at 1 and α at 0.99 provides the opportunity to find most of the best-known solutions (BK) at relatively short run time. Due to space constraints, only a summary of the result comparisons is presented in this section. Note that "p" refers to the average objective function value and "$t(s)$" refers to the computation time (in seconds). $\%d$ refers to the percentage difference between "p" for proposed algorithm and "p" for best known solution (BK).

Table 2. Computational Results

Instance	BK	VNS			TSf			BiF&F-f			ILS			SA_ILS		
	p	*p*	*t*(*s*)	%*d*	*p*	*t*(*s*)	%*d*	*p*	*t*(*s*)	%*d*	*p*	*t*(*s*)	%*d*	*p*	*t*(*s*)	%*d*
Set 1	1814.2	1814.2	**0.0**	0.00	1814.0	43.3	0.02	**1824.0**	0.2	**-0.34**	1814.0	19.0	0.01	**1824.0**	21.4	-0.32
Set 2	295.2	**294.9**	667.1	0.07	295.0	505.9	0.06	295.1	**7.9**	**0.03**	292.5	30.0	0.88	293.1	24.1	0.56
Set 3	728.4	728.4	1028.5	**0.00**	726.2	388.0	0.29	**728.5**	**6.0**	**-0.02**	727.9	128.4	0.08	725.5	55.6	0.39
Average	945.9	945.8	565.2	0.03	945.0	312.4	0.12	**949.0**	**4.7**	**-0.11**	944.8	59.2	0.32	947.0	33.7	0.21

As seen from the table above, the proposed algorithm is comparable against the other algorithms in terms of quality and computation time. Furthermore, it is also able to improve one best-known solution from Set 1 of the benchmark instances. The proposed algorithm's computation time is also shorter as compared to the algorithms VNS, TSf and ILS. However, it is worth mentioning that BiF&F-f algorithm is far more superior than all other algorithms in both the results and computation time.

4. Conclusion

In this paper, Simulated Annealing with Iterated Local Search (ILS) metaheuristic algorithm is proposed to solve the Capacitated Team Orienteering Problem (CTOP). It combines Simulated Annealing process with ILS operators to enhance the algorithm's diversification and intensification. The algorithm is then used to solve benchmark instances and the results are compared against other state-of-the-art algorithms. The proposed algorithm displayed good performance that is comparable with other state-of-the-art algorithms in both the results and computation time. Furthermore, the relatively simple algorithm structure, along with few user-defined parameters, indicates the applicability of the proposed approach towards solving other variants of the Orienteering Problems, as well as the real-life Orienteering Problems. The results are still preliminary with some future research directions, such as developing more local search operators and a more rigorous tuning procedure using statistical tests.

Reference

Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society, 60*(6), 831-842.

Ben-Said, A., El-Hajj, R., & Moukrim, A. (2016). An adaptive heuristic for the capacitated team orienteering problem. *IFAC-PapersOnLine, 49*(12), 1662-1666.

Butt, S. E., & Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers and Operations Research, 21*(1), 101-111.

Chao, I., Golden, B. L., & Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research, 88*(3), 464-474.

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL), 34*(3), 307-318.

Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research, 255*(2), 315-332.

Gunawan, A., Lau, H. C., Vansteenwegen, P., & Lu, K. (2017). Well-tuned algorithms for the team orienteering problem with time windows. *Journal of the Operational Research Society, 68*(8), 861-876.

Gunawan, Aldy; Ng, Kien Ming; Yu, Vincent F.; Adiprasetyo, Gordy; and Lau, Hoong Chuin. The capacitated team orienteering problem. (2019). Proceedings of the 9th International Conference on Industrial Engineering and Operations Management Bangkok, Thailand, March 5-7, 2019. 1630-1638.

Lin, S., & Yu, V. F. (2015). A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows. *Applied Soft Computing, 37*, 632-642.

Luo, Z., Cheang, B., Lim, A., & Zhu, W. (2013). An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. *European Journal of Operational Research, 229*(3), 673-682.

Tarantilis, C. D., Stavropoulou, F., & Repoussis, P. P. (2013). The capacitated team orienteering problem: A bi-level filter-and-fan method. *European Journal of Operational Research, 224*(1), 65-78. doi:10.1016/j.ejor.2012.07.032

Vansteenwegen, P., Souffriau, W., & Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research, 209*(1), 1-10.