Simulated Annealing for the Route Planning of Advisors in the RMUTL Cooperative Education Program

Parida Jewpanya*

Rajamangala University of Technology Lanna Tak, Thailand 63000

Received: 15 March 2019; Revised: 3 May 2019; Accepted: 17 September 2019

Abstract

The Rajamangala University of Technology Lanna (RMUTL) is Thailand's leading university in the field of science and technology. The university focuses on producing graduates who are ready for workplaces through a hands-on learning system. RMUTL incorporates a Cooperative Education Program (CO-OP) in the learning system which aims to gives students an opportunity to receive career training in industries. The students in the final academic year are required to attain this program for a semester to be prepared for the workplace. During the students' internship period, the RMUTL CO-OP has a team of advisors who need to visit those students in the industries at least two times to give the students advice on their projects and working life. The current visiting plan is created based on the planner experience which sometimes ineffective in term of costs. Therefore, in this study, a mixed-integer linear programming model is formulated to help the planner find an optimal route for the advisors. The objective is to minimize the total cost of the advisor's visitation. Moreover, we propose a simulated annealing heuristic (SA) to solve the problem. We generate benchmark instances and solved them by SA. Computational results show the excellent performance of SA in terms of solution quality and computational efficiency.

Keywords: Cooperative education program, simulated annealing heuristic, mixed-integer linear programming

* Corresponding author. E-mail: parida.jewpanya@gmail.com

Industrial Engineering, Faculty of Engineering, Rajamangala University of Technology Lanna Tak

1. Introduction

The Cooperative Education Program (CO-OP) is an internship program that gives students an opportunity to receive career training as they work with professionals in their major fields of study [1]. The students who attend this program are encouraged to build essential knowledge and skills such as teamwork and problem-solving in the real-world experience. Moreover, it supports the development of graduates and making them more employable and adaptable at the workplace [2].

The organization in CO-OP program consists of competitive industry leaders and higher education institutions "cooperating" with each other to provide hands-on work experience to full-time actively enrolled students within a degree-seeking program [3]. Rajamangala University of Technology Lanna (RMUTL) requires students in the final academic year to attain the CO-OP program for four months (one semester). The students are eligible to apply to CO-OP after completing a minimum requirement of the university. The RMUTL CO-OP allows the students to select the industry by themselves under the direction of the department's CO-OP advisors. Then, the RMUTL CO-OP coordinator will contact the chosen industry and process the application. During the student's 'internship' period, the industries are requested to assign students the project related to their field of study. This project offers students the opportunity to gain real-world experience in both theory and practical skill.

The RMUTL CO-OP has a team of advisors who come from different departments and have different expertise. Those advisors are required to visit the students in the industries at least two times during the students' internship period. The purpose of this visit is to give the students advice on their projects and working life. Moreover, the visit can bring out a stronger relationship between university and industries. To arrange this activity, the CO-OP coordinators have a responsibility to plan visiting routes for the advisors in each department by manually created a route base on the planner experience which sometimes ineffective in term of costs and time.

The coordinators desire a good solution to help them make significant improvements on planning this activity. Due to the dynamic of the visiting locations, there is no methodology or repeatable process that available for the coordinator which can be applied directly. Therefore, in this paper, we propose the model for assigning and routing advisors visitation to industries. The proposed model is modified from the vehicle routing problem (VRP) model which can be described as the problem of designing optimal delivery or collection of routes from a depot to some customers subject to side constraints [4, 5]. In using the model, we incorporate guidelines concerning capacity, flow balance, and available travel schedules of advisors. The model's objective is to minimize the total cost.

The combinatorial nature of VRP makes this type of problem an NP-hard problem. Thus, studies with the same intrinsic complexity usually use heuristic and meta-heuristic solution approaches. In this paper, we propose a simulated annealing heuristic that incorporates several neighborhood structures to improve the performance on solving the CO-OP instances. The performance of the proposed SA in solving CO-OP instances is verified by comparing its solution quality and computational time with those obtained by GUROBI. This study provides a decision support tool for the CO-OP coordinators mainly at the operational decision-making level.

This paper organized as follow. Section 2 presents the literature review. Section 3 provides problem statement and mathematical model formulation. Section 4 shows the detailed descriptions of the proposed SA algorithm. The solution representation scheme is also explained. Section 5 shows a comparative analysis of SA and GUROBI results and presents the sensitivity. Finally, Section 6 concludes the paper and presents future research directions.

2. Literature review

The vehicle routing problem (VRP) and its variants play an important role in many distribution and transportation systems, as well as the costs associated with operating vehicles [4, 6, 7]. VRP can be described as the process of determining optimal routes from one depot to some geographically scattered customers, subject to side constraints [8]. One of the classical versions of VRP is the capacitated vehicle routing problem (CVRP), wherein a vehicle must satisfy a certain vehicle capacity restriction [9]. Subsequently, when a total distance or time restriction is also imposed on CVRP, the problem becomes a distance-constrained capacitated vehicle routing problem (DCVRP) [10]. Another special variant of VRP is an open vehicle routing problem (OVRP), in which the route does not require vehicles to return to the depot to avoid adding extra mileage to the compensation model [11, 12].

VRP has been studied extensively over the past few decades by using different approaches [4], including the exact method, heuristic method and metahueristic have been used to solve VRP and its extensions [13]. The basic VRP model, known as the NP-hard problem, uses the exact method to determine the optimal solution; however, this method is considerably difficult and usually requires a long computational time. Therefore, heuristic and evolutionary computing methods have been applied to determine a near-optimal solution in a reasonable amount of time such as: (1) simulated annealing, (2) Tabu search, (3) genetic algorithms, (5) particle swarm optimization, and (6) some recently developed hybrid heuristic algorithms. For further information one may refer to works of Yu, Jewpanya [14], Osman [15], Barbarosoglu and Ozgur [4], Lee, Jung [16], Baker and Ayechew [17], Kachitvichyanukul [18], Berger and Barkaoui [19] and Berger and Barkaoui [20].

In practice several variants of the problem exist because of the diversity of operating rules and constraints encountered in real-life applications. This study focuses on the development of the routing problem, which is motivated by the problem faced in the real case. Although routing problems have been studied and solved using a variety of methods, the proposed problem has several unique requirements.

3. Problem definition and model formulation

This study considers the vehicle routing problem. *D* is a set of a depot that is designated as the starting and end location, that is, *D* = {0}. A set *F* is a set of industries, *F* = {1, 2... | *F* |}. Each industry is allowed to visit at most once. A visiting time t_i is associated with industry $i \in F$. *V* is a set of vehicles, *V*= {1, 2, 3, ..., |V|}. A set of routes is constructed such that all industries will be visited. The starting location and the ending location of the routes are fixed.

Parameters:

t_i	Visiting time at a factory *i*
d_{ij}	Travel distance from location *i* to *j*
h_{ij}	Travel time from location *i* to *j*
VCAP	Number of industries allowed per route
CDIST	Cost per unit distance
CRENT	Renting cost of vehicle per day
CADV	Allowances for an advisor per day
CHOT	Cost of hotel renting per day
N	Number of advisors assigned in each route
W	Working hours per day
M	Big number

Variables:

D_v	Number of days of each route *v*
S_{iv}	Start time of a visit to industry *i* in route *v*
x_{ijv}	1, if a visit to industry *i* is followed by a visit to factory *j* by route *v*; 0, otherwise

The objective of the problem is to determine the number of routes and the best vehicle routes for advisors to visit industries. The sum of the travel costs is minimized consisting of the cost of vehicles, the allowance and hotel renting cost of advisors.

$$Min \sum_{\forall i,j \in F \cup D} \sum_{v \in V} x_{ijv} \cdot d_{ij} \cdot CDIST + \sum_{\forall v \in V} D_v \cdot (CRENT + N \cdot CADV) + \sum_{\forall v \in V} (D_v - 1) \cdot N \cdot CHOT$$

Constraints:

$$\sum_{\forall k \in F \cup D} \sum_{\forall v \in V} x_{k,k,v} = 0 \quad (1)$$

$$\sum_{\forall j \in F, i \neq j} x_{j,i,v} - \sum_{\forall j \in F, i \neq j} x_{i,j,v} = 0 \ , \forall i \in F, \forall v \in V \quad (2)$$

$$S_{jv} \geq S_{iv} + t_i + t'_{ij} - M(1 - x_{ijv}) \quad , \forall i, j \in F, \forall v \in V \quad (3)$$

$$S_{jv} \geq t'_{ij} \cdot x_{ijv}, \forall i \in D, \forall j \in F, \forall v \in V \quad (4)$$

$$\sum_{\forall i \in F \cup D} x_{ijv} \leq 1, \quad \forall v \in V, \forall j \in D \quad (5)$$

$$\sum_{\forall i \in F \cup D} \sum_{\forall v \in V} x_{ijv} = 1, \quad \forall j \in F \quad (6)$$

$$\sum_{\forall i \in F \cup D} \sum_{\forall j \cup F} x_{ijv} \leq VCAP, \quad \forall v \in V \quad (7)$$

$$D_v \cdot W \geq S_{jv} + t'_{ji} \cdot x_{jiv}, \forall i \in D, \forall j \in F, \forall v \in V \quad (8)$$

Constraint (1) guarantees that a vehicle does not travel inside itself. Constraint (2) is the connectivity of each route. Constraint (3) determines the timeline of each route. Constraint (4) makes sure that the start service time at the first visited industry of each route should be greater than the traveling time from depot 0 to *j.* Constraint (5) ensures that vehicle *v* can be used only one time. Constraint (6) provides that an industry is visited at most once. Constraint (7) confirms that the total visited industry does not exceed the number of industries allowed in each route. Constraint (8) determines the number of the day used in each route.

4. Simulated Annealing (SA)

The SA algorithm is a local search-based heuristic algorithm that is capable of avoiding being trapped in a local optimum and can explore a wider area of the search space. SA was introduced by Metropolis, Rosenbluth [21] and popularized by Kirkpatrick, Gelatt [22]. SA has been successfully used to solve several problems such as location routing problem (LRP) [23], vehicle routing problem [14], and team orienteering problem (TOP) [24].

SA typically starts with an initial solution. At each iteration, the algorithm randomly selects a neighborhood move to generate a new solution from the current solution. The neighborhood moves that are normally used in SA are swap, reverse, and insert. In the SA searching process, the new solution is generated by one of these moves. If the new solution is better than the current solution, it replaces the current solution. The search process resumes from the new current solution. However, a small probability is calculated using the Boltzmann function in which a worse solution is accepted as the new current solution.

4.1 Solution representation

The solution representation consists of routes of advisors, that is represented by permutation numbers consisting of $|F|$ industries, which are denoted by the set $\{1, 2, ..., |F|\}$ and N_{dummy} zeros. The parameter N_{dummy} for the route is equal to $|V|-1$. The N_{dummy} is the number of zero added to the solution representation that are used to terminate routes. Figure 1 illustrates the solution representation.

Industries				Dummy zeros		
1	2	...	$\|F\|$	0	0	0

Figure 1 The solution representation

To demonstrate the solution and route construction, we give an example with a small instance consisting of seven industries and four vehicles. The solution representation is shown at Figure 2.

Industries							Dummy zeros		
1	2	3	4	5	6	7	0	0	0

Figure 2 The example of a solution representation

Figure 3 presents an example of a solution. The first vehicle starts from depot. The vehicle visits industries 1 and 2. The industry 2 is followed by a zero; thus, the first route is terminated. The second vehicle visits only industry 6 and followed by a zero; thus, the second route is terminated. The third vehicle services industries 4 and 7. Finally, the fourth vehicle visits industries 3 and 5.

1	2	0	6	0	4	7	0	3	5

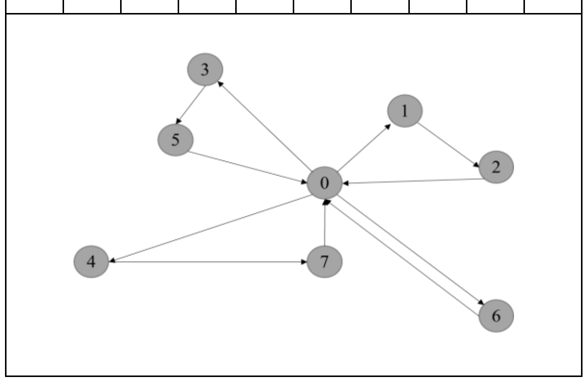


Figure 3 The example of a solution

4.2 Initial solution

An important factor for obtaining an effective initial solution is the effective utilization of vehicle. To generate the initial solution, we try to produce a feasible solution with a greedy strategy. Such a strategy is inspired by the probabilistic nearest neighbor heuristic approach. The steps of this method are described as follows.

***Step* 1.** Gather all the new information for the current period, including data of factory locations. The coordinator determined the amount of time each person can work.

***Step* 2.** Clusters the industries based on a region. For example, the industries that located in the same province are assigned into the same cluster.

***Step* 3.** Assign a vehicle route to each cluster. The order of visiting is arbitrary.

***Step* 4.** Determine the number of days and travel cost of each route.

4.3 Neighborhood moves

The proposed SA algorithm incorporates three neighborhood structures to explore different possibilities of route combination. We independently use all neighborhoods the routes. These neighborhood structures are explained as follows.

Swap move: The swap procedure starts by taking random i^{th} and j^{th} positions of a solution and systematically swapping the numbers in these two positions. For example, the swap procedure is applied to the routes in Figure 3. The 2nd and 4th positions of the solution are randomly selected. The numbers appearing in these positions are 2 and 6, respectively. These numbers are then systematically swapped. After finishing the swap operation, the pickup route is updated to {1, 6, 0, 2, 0, 4, 7, 0, 3, 5}.

Reverse move: The reverse procedure operates by reversing the direction of numbers in the position between two randomly chosen i^{th} and j^{th} positions of a solution. For example, the 3nd and 6th positions of the solution in Figure 3 are chosen. The numbers in the positions between these random positions are 0, 6, 0, 4. The direction of these numbers is then reversed. Here, the solution is updated to {1, 2, 4, 0, 6, 0, 7, 0, 3, 5}.

Insertion move: This insert operator is executed by randomly selecting the i^{th} and j^{th} positions of a solution and then inserting an i^{th} number in front of the j^{th} position. Using the same example with the previous, the 5th and 1st positions are randomly selected from solution in Figure 3. The numbers in these positions are 0 and 1. In this case, 0 is inserted in front of 1. Therefore, the solution is updated to {0, 1, 2, 0, 6, 4, 7, 0, 3, 5}.

4.4 Parameters used

The SA algorithm uses five parameters, namely, I_{iter}, T_0, T_f, K and α, to search for the best solution. I_{iter} denotes the number of iterations that proceed at a particular temperature. T_0 and T_f represent the initial and final temperatures, respectively, and K is the Boltzmann constant used in the calculation of the probability of accepting a worse solution. Lastly, α is the cooling rate coefficient to control the cooling process.

4.5 SA Procedure

Algorithm 1 describes the steps of the SA heuristics. The search mechanism starts with an initial solution and follows the SA searching process to improve it. The SA searching process starts at setting the current temperature T to the initial temperature T_0. The best solution X_{best} and the current solution X are set to be the initial solution. The best objective value is set to be the objective value of solution X. Here, R_t is the probability of choosing neighborhood t, $t \in$ {swap, reverse, insert}. R_t is set to be 1/3.

The process continues to generate a new solution Y from the current solution X by the neighborhood moves in each iteration at a particular temperature. Let Δ be the objective value difference between Y and X. If $\Delta < 0$, then Y is better than X; thus, Y replaces X as the current solution; otherwise, the new neighborhood solution is accepted with a probability calculated by the Boltzmann function, $\exp(-\Delta/KT)$. The next temperature is that the current temperature decreases to αT. The SA algorithm is terminated if the current temperature is below or equal to the final temperature T_f.

Algorithm 1. SA

```
Input: I_iter, T_0, T_f, α , K, InitialSolution;
 Output: Objective;
        I ← 0; T ← T_0; F_best ← obj (X), X ← InitialSolution; X_best ← X;
        R_t ← 1/3 for all t in {swap, reverse, insert};
        While T < T_f;
                I ← 1; N_t ← 0 and O_t ← Ø for all t in {swap, reverse, insert};
                While I ≤ I_iter;
                        r ← random ( 0, 1);
                        If (r ≤ R_swap) then
                        Generate a new solution Y from X by swap move;
                        Else if (R_swap < r ≤ R_swap+ R_reverse) then
                        Generate a new solution Y from X by reverse move;
                        Else Generate a new solution Y from X by insertion move;
                        End if
                Δ ← obj (Y) – obj (X);
                If (Δ ≤ 0) then X ← Y;
                Else r ← random ( 0, 1);
                If (r < exp(−Δ/KT)) then X ← Y; End if
                End if
                If ( obj (X) < F_best ) then X_best ← X; F_best ← obj (X); End if
                I ← I + 1;
                End while
        T = αT;
        End while
```

5. Computational study

5.1 Instances

We generate the CO-OP instances from real geographical distances of industries in Thailand. Those industries have accepted the students from the industrial engineering department of RMUTL, Tak, to attend the CO-OP program. The dataset contains 8 to 100 industries, with a total of 40 instances. The problem parameter values are shown in Table 1.

Table 1 the problem parameter values

Scale	No. of industries	No. of instances	No. of Vehicle	VCAP	CDIST	CRENT	CADV	CHOT	N	W
Small	8	5	3	4, 5, 6, 7, 8	4	1800	240	750	2	12
	11	5	3	6, 7, 8, 9, 11						
	17	5	6	3, 6, 11, 15, 17						
Medium	25	5	6	6, 12, 15, 20, 25						
	30	5	6	6, 10, 15, 20, 30						
	50	5	6	10, 20, 30, 40, 50						
Large	80	5	10	10, 20, 40, 60, 80						
	100	5	10	20, 30, 50, 80, 100						

5.2 Parameter settings

We conduct the experiment to select the best combination of SA parameters. We use the same methodology as reported in Yu, Lin [23], who introduced a simple method for selecting the parameter combination. The method first introduces several values for each parameter. The process then tests the combinations of these parameter values and selects the combination that can obtain the lowest average cost for solving problem instances. In the proposed algorithm, the parameter values tested are as follows:

I_{iter} = 500, 800, and 1000;

T_0 = 3, 5, 7;

T_f = 0.1, 0.01, 0.001;

α = 0.8, 0.9, 0.99;

K = 1, 1/2, 1/3, 1/4.

The results indicate that the best solution quality can be obtained from the parameter values: I_{iter} = 500, T_0, = 5, T_f = 0.001, α = 0.99, K = 1.

5.3 Computational results

5.3.1 The results of the small-scale CO-OP instances

To confirm optimality, we compare the results from the proposed SA with those obtained from GUROBI. GUROBI is a commercial optimization solver for linear programming (LP). The commercial solver was terminated after 4 hours if it could not find an optimal solution. Tables 2 report the best objective values and computational time (CPU) of the small-scale instances, respectively. The respective optimality gap is calculated by equation (9), where *Solution* represents the best solution value of SA. The *BKS* denotes the solutions obtained by GUROBI.

$$Gap\ (\%) = \frac{Solution - BKS}{BKS} \times 100 \qquad (9)$$

Comparative results in Table 2 show that the average gap upon the best-known solution is -0.06% for small-scale instances. As for computational time, SA can obtain the solution in an average 1.82 s. In summary, the proposed SA obtains six optimal solutions same as the GUROBI solutions and five solutions better than GUROBI solutions.

Table 2 Comparison of CO-OP solution for small-scale instances

No.	Instance ID	Objective values		%Gap	Computational time	
		Gurobi	SA		Gurobi	SA
1	S8_4	26460*	26460	0.00	22.08	1.45
2	S8_5	23544*	23544	0.00	23.09	1.16
3	S8_6	23544*	23544	0.00	27.36	1.18
4	S8_7	20580*	20580	0.00	10.84	1.26
5	S8_8	18276*	18276	0.00	9.39	1.19
6	S11_6	27808*	27808	0.00	6173.94	1.36
7	S11_7	27808	27808	0.00	34065.20	1.52
8	S11_8	27256	27256	0.00	42273.10	1.26
9	S11_9	27256	27256	0.00	14428.80	1.64
10	S11_11	21916	21916	0.00	14406.80	1.65
11	S17_6	68188	68068	-0.18	38193.00	3.89
12	S17_11	61600	61468	-0.21	44947.30	2.57
13	S17_13	61516	61368	-0.24	39232.00	2.57
14	S17_15	61428	61368	-0.10	43213.70	2.42
15	S17_17	55800	55720	-0.14	43684.80	2.17
	Average	**36865.33**	**36829.33**	**-0.06**	**21380.76**	**1.82**

5.3.2 The results of the medium and large-scale CO-OP instances

For the medium and large-scale CO-OP instances, GUROBI fails to find the solution within 4 hours. The proposed SA is tested on medium and large-scale CO-OP instances. We evaluate the performance of SA over the initial solution using the percentage of the relative improvement (Improv), which is calculated by equation (10); *SASolution* is the best solution values obtained by SA and *InitialSolution* is the solution obtained from the initial solutions.

$$\text{Improv }(\%) = \frac{InitialSolution - SASolution}{InitialSolution} \times 100 \qquad (10)$$

Tables 3 and 4 display the performance of SA compared to the initial solution in solving the medium and large scale instances, respectively. We offer two observations about the objective values and CPU. First, Table 3 shows that the best objective values of the medium-scale instances obtained by SA are improved from initial solution by an average of 9.25%. For the large-scale instances, Table 4 reports that the best SA solutions are better than the initial solutions by an average of 7.39%. However, as for computational time, the initial solution can obtain the solution faster than SA.

Table 3 Comparison of CO-OP solution for medium-scale instances

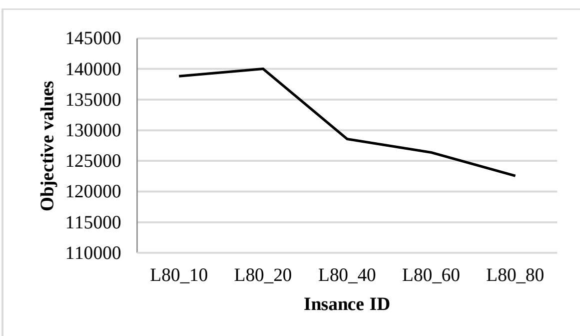
		Objective values			Computational time	
No.	Instance ID	Initial Solution	SA	%Gap	Initial Solution	SA
1	M25_6	75612	62968	16.72	0.58	2.40
2	M25_12	60560	53936	10.94	0.51	2.28
3	M25_15	60500	55044	9.02	0.73	2.14
4	M25_20	60780	51184	15.79	0.66	2.23
5	M25_25	54624	48032	12.07	0.62	2.17
6	M30_6	86936	75948	12.64	0.55	2.56
7	M30_10	67512	65316	3.25	0.69	2.62
8	M30_15	61204	58348	4.67	0.79	2.48
9	M30_20	61620	57004	7.49	0.53	2.27
10	M30_30	55448	51132	7.78	0.93	2.36
11	M50_10	117424	106540	9.27	0.61	3.51
12	M50_20	105968	94228	11.08	0.49	3.53
13	M50_30	91892	88104	4.12	0.96	3.44
14	M50_40	96216	86832	9.75	0.15	3.00
15	M50_50	85728	82120	4.21	0.80	2.97
	average	**76134.93**	**69115.73**	**9.25**	**0.64**	**2.66**

Table 4 Comparison of CO-OP solution for large-scale instances

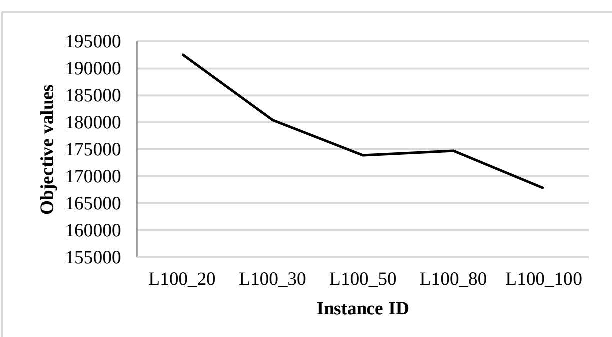
		Objective values			Computational time	
No.	Instance ID	Initial Solution	SA	%Gap	Initial Solution	SA
1	L80_10	176436	138804	21.33	1.62	9.73
2	L80_20	150420	140004	6.92	1.25	9.13
3	L80_40	135076	128556	4.83	1.65	8.78
4	L80_60	135336	126380	6.62	1.58	8.80
5	L80_80	129172	122580	5.10	1.38	8.87
6	L100_20	206736	192624	6.83	1.77	19.31
7	L100_30	196860	180412	8.36	1.85	20.18
8	L100_50	184620	173880	5.82	1.79	20.64
9	L100_80	181768	174728	3.87	1.92	23.00
10	L100_100	175144	167736	4.23	1.85	21.30
	average	**167156.80**	**154570.40**	**7.39**	**1.67**	**14.97**

5.4 Sensitivity analysis

To provide more insight on the problem, we conducted sensitivity analysis on the impact of the number of industries allowed in each route. The large-scale instances are tested with five values of VCAP. The first set of data consists of 80 industries. VCAP equals to 10, 20, 40, 60 and 80 industries are allowed in each route. The second set of data consists of 100 industries. VCAP equals to 20, 30, 50, 80 and 100 industries are allowed in each route. Figure 4 (a) and (b) illustrate that the higher value of VCAP can give the lower total costs. According to the results, the CO-OP planner should consider to increase the number of industries visited in one route.



(a)



(b)

Figure 4 Sensitivity analysis on the number of industries allowed in each route (VCAP)

6. Conclusions and further work

This paper has proposed a new model for routing the advisors in the RMUTL CO-OP program. The model has applications in a wide range of settings, including planning, and strategic environments. Moreover, the model gives consulting firm management a method for calibrating the impact of a number of industries allowed in each route on the total costs.

Since the routing for CO-OP is a new problem, we have generated three sets of benchmark instances including small-scale, medium-scale and large-scale instances. The proposed SA is used to solve three set of instances. The results show that the proposed SA outperforms GUROBI both in solution quality and solution time. Furthermore, GUROBI fails to solve medium- and large-scale instances, whereas the proposed SA heuristic solves all instances in the three sets. When the RMUTL implemented the solution in its routing planning process, it realized travel cost savings. Moreover, the sensitivity analysis shows that when the number of factories allowed in each route is increasing, it causes the lower total cost.

Future studies may consider CO-OP with more practical constraints. They can take uncertainty of traveling and service times into account so as to bring the problem closer to reality. Alternatively, they can focus on developing exact methods or different heuristics that exploit the problem characteristics for solving CO-OP instances.

References

[1] Mason, R.E., "Cooperative occupational education and work experience in the curriculum," 1989: ERIC.

[2] Blair, B.F., M. Millea, and J. Hammer, "The impact of cooperative education on academic performance and compensation of engineering majors," *Journal of Engineering Education*., 93(4): p. 333-338, 2004.

[3] Thiel, G.R. and N.T. Hartley, "Cooperative education: A natural synergy between business and academia," *SAM Advanced Management Journal*, 62(3): p. 19, 1997.

[4] Barbarosoglu, G. and D. Ozgur, "A tabu search algorithm for the vehicle routing problem," *Computers & Operations Research*, 26(3): p. 255-270, 1999.

[5] Eksioglu, B., A.V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, 57(4): p. 1472-1483, 2009.

[6] Christofides, N., "The vehicle routing problem," *Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle*, 10(V1): p. 55-70, 1976.

[7] Desrochers, M., J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations research*, 40(2): p. 342-354, 1992.

[8] Laporte, G., "The vehicle routing problem: An overview of exact and approximate algorithms," *European journal of operational research*, 59(3): p. 345-358, 1992.

[9] Toth, P. and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, 123(1-3): p. 487-512, 2002.

[10] Kek, A.G., R.L. Cheu, and Q. Meng, "Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots," *Mathematical and Computer Modelling*, 47(1-2): p. 140-152, 2008.

[11] Schrage, L., "Formulation and structure of more complex/realistic routing and scheduling problems," *Networks*, 11(2): p. 229-232, 1981.

[12] Sariklis, D. and S. Powell, "A heuristic method for the open vehicle routing problem," *Journal of the Operational Research Society*, 51(5): p. 564-573, 2000.

[13] Laporte, G., " Fifty Years of Vehicle Routing," *Transportation Science*, 43(4): p. 408-416, 2009.

[14] Yu, V.F., P. Jewpanya, and A.A.N.P. Redi, "Open vehicle routing problem with cross-docking," *Computers & Industrial Engineering*, 94: p. 6-17, 2016.

[15] Osman, I.H., "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of operations research*, 41(4): p. 421-451, 1993.

[16] Lee, Y.H., J.W. Jung, and K.M. Lee, "Vehicle routing scheduling for cross-docking in the supply chain," *Computers & Industrial Engineering*, 51(2): p. 247-256, 2006.

[17] Baker, B.M. and M. Ayechew, "A genetic algorithm for the vehicle routing problem" *Computers & Operations Research,* 30(5): p. 787-800, 2003.

[18] Kachitvichyanukul, V., "Particle swarm optimization and two solution representations

for solving the capacitated vehicle routing problem," *Computers & Industrial Engineering*, 56(1): p. 380-387, 2009.

[19] Berger, J. and M. Barkaoui, "A hybrid genetic algorithm for the capacitated vehicle routing problem," *Genetic and evolutionary computation conference*, Springer, 2003

[20] Berger, J. and M. Barkaoui, "A new hybrid genetic algorithm for the capacitated vehicle routing problem," *Journal of the operational Research Society*, 54(12): p. 1254-1262, 2003.

[21] Metropolis, N., et al., "Equation of State Calculations by Fast Computing Machines,". *The Journal of Chemical Physics*, 21(6): p. 1087, 1953.

[22] Kirkpatrick, S., J.C. Gelatt, and M. Vecchi, "Optimization by simmulated annealing," *science*, 220(4598): p. 671-680, 1983.

[23] Yu, V.F., et al., "A simulated annealing heuristic for the capacitated location routing problem," *Computers & Industrial Engineering*, 58(2): p. 288-299, 2010.

[24] Lin, S.-W. and V.F. Yu, "A simulated annealing heuristic for the team orienteering problem with time windows," *European Journal of Operational Research*, 217(1): p. 94-107, 2012.