

# A Bi-Modal Routing Problem with Cyclical and One-Way Trips: Formulation and Heuristic Solution

Roger B. Grinde

University of New Hampshire, USA

**Abstract** – A bi-modal routing problem is solved using a heuristic approach. Motivated by a recreational hiking application, the problem is similar to routing problems in business with two transport modes. The problem decomposes into a set covering problem (SCP) and an asymmetric traveling salesperson problem (ATSP), corresponding to a hiking time objective and a driving distance objective. The solution algorithm considers hiking time first, but finds all alternate optimal solutions, as inputs to the driving distance problem. Results show the trade-offs between the two objectives.

**Keywords** – Heuristic, mountaineering, optimisation, operations research, vehicle routing.

## I. INTRODUCTION

This paper addresses a combinatorial problem combining the classical Set Covering Problem (SCP) and the Traveling Salesman Problem (TSP). The problem is termed the Bi-Modal Covering Salesman Problem (BCSP). Motivated by a mountaineering problem, it has broader application in routing & delivery problems involving two modes of transport and the possibility of one-way routes while visiting certain nodes. Following the introduction, relevant literature is reviewed. The mathematical formulation is then developed, followed by details of the implementation of the heuristic algorithm, computational results, and conclusions with possible extensions.

The motivating application is used to describe the BCSP here. Given a set of mountain peaks one wishes to climb, a network of trails, a set of parking areas (which serve as trailheads), and network of roads connecting trailheads determine the set of optimal hikes to perform and sequence the hikes so as to minimise an objective involving both hiking and driving costs. The idea is that one will park a car at a trailhead, perform a hike, and drive to a new hike, eventually climbing all mountains (camping is generally needed between hikes and sometimes within hikes; this overnight time is not considered in the current problem). However, there is another element to the trail network. Sometimes it is beneficial and/or necessary to have a hike begin and end at different locations (trailheads), rather than traverse the same territory twice. These one-way hikes are a unique element to this paper, but in a hiking application they are very common. Therefore, it is assumed that there are at least two persons hiking together (very reasonable given safety considerations) as well as two cars. Two cars are needed to perform shuttling operations on the one-way hikes. The Problem Definition and Formulation Section explains this in more detail. The formulation and solution approach are not dependent on the need for two cars.

Specifically, the motivating problem here is the climbing of the 48 peaks above 4000 feet in the State of New Hampshire, USA [1]. Many hikers aim to become part of the “club” who have ascended these mountains. The question addressed in this paper is to find the “optimal” tour through these peaks, considering both hiking and driving objectives. This paper presents a precedence-based dual objective approach, which is implemented using a traditional programming language and optimisation engine, as well as a spreadsheet-based (with additional programming) approach. Because of the bi-modal nature of the problem and logistics of performing certain types of hikes, the TSP aspect of the problem becomes asymmetric.

This routing problem has application in other areas. Rather than mountain peaks and trailheads, let us consider customers and airports. One must visit all customers, having access to both flights and rental cars. Sometimes one would fly into a given airport, visit several customers, and return to the same airport before flying to another airport. Other times it may be beneficial to fly into the first airport, visit several customers before driving to a different airport (i.e., a one-way trip). In addition, with the increased availability of car-sharing services, there are potential applications involving driving and walking. Other applications are in situations with bi-modal delivery services.

## II. LITERATURE REVIEW

Routing problems have been studied in the literature for many years. For a detailed review of multi-objective problems, see Jozefowiez et al. [2]. Excellent references for the Traveling Salesman Problem (TSP), the Vehicle Routing Problem (VRP), and arc routing problems are by Gutin [3], Applegate et al. [4], Golden et al. [5], Dror [6], and Braekers et al. [7].

Specific problems with similarities to the BCSP addressed in this paper include the Covering Tour Problem (CTP). This was introduced by Current & Schilling [8] and further studied by Gendreau et al. [9] and Jozefowieza et al. [10]. This problem has two node sets as the BCSP does. In the CTP, the goal is to find a minimal-length Hamiltonian tour through a subset of the nodes of one set, such that every node in the other set will be within some critical distance of a visited node. Current & Schilling provide an example of visiting health care teams in developing countries: only a small number of villages can be visited, but villages not visited must be reasonably close to a visited village. In the BCSP, the nodes in the second set must all be explicitly visited.

The Prize-Collecting Traveling Salesman Problem has some similarities. See, for example, Fischetti & Toth [11] and Balas [12]. In this problem, a reward is received at each node visited

with the objective to find a minimal-length tour, which provides a certain level of total rewards. The Orienteering Problem also contains rewards at each node; the objective is to maximise the total reward collected subject to a time constraint. See, for example, Golden et al. [13] and Ramesh et al. [14].

Another similar problem is studied by Levy & Bodin [15]. They develop an approach for scheduling postal carriers who drive to a parking area, complete a walking tour delivering mail, return to the vehicle, and either drive to a new parking area or complete another walking tour. The problem is termed the Walking Line of Travel Problem (WLT) and presents an algorithm but no mathematical formulation. The BCSP studied in this paper is similar, but more general. The WLT problem includes only cyclical trips, whereas the BCSP can include one-way trips as well. The form of the BCSP studied in this paper also allows one to define walking trips a priori. Mail delivery problems are also studied by Gussmagg-Pfieggl et al. [16].

As the motivating problem for this paper lies in the recreational arena, it is fitting that other recreational routing problems have also been studied in the literature. Cycle-tourist routing is studied by Cerna et al. [17]. Tourist trip design problems are studied by Gavalas et al. [18].

### III. PROBLEM DEFINITION AND FORMULATION

The BCSP occurs on two inter-connected networks. One network is a network of parking areas connected by roads. The other network is a network of peaks connected by trails. Some trails also have connections to parking areas. It is useful to define the overall network as  $G$ , with two sub-networks  $G_1$  (road network) and  $G_2$  (trail network).  $G_1$  comprises the set of parking areas denoted by  $V_1 = \{P_1, P_2, \dots, P_m\}$ , where  $P_i$  is parking area  $i$ , and  $E_1$  the set of roads connecting the parking areas. Similarly,  $G_2$  is the set of peaks (“customers” in a traditional TSP language), each of which must be visited at least once. The peaks will be denoted as  $V_2 = \{C_1, C_2, \dots, C_n\}$ , and  $E_2$  – the trails linking the peaks as well as the parking areas. Figure 1 displays an example.

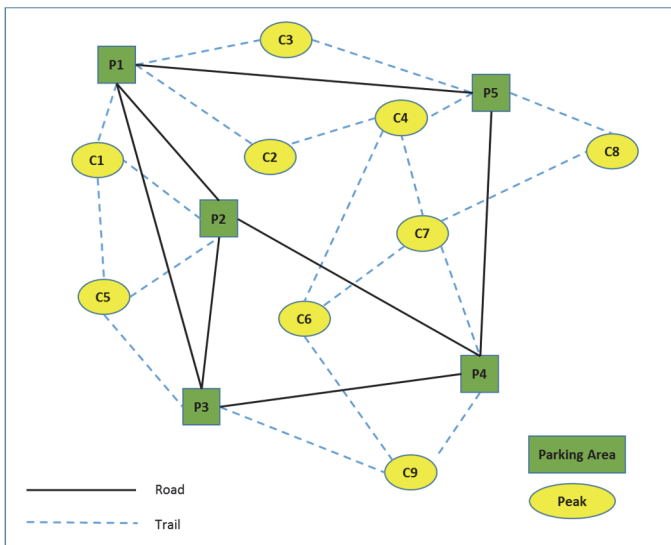


Fig. 1. Illustration of road and trail networks.

The concept of a trip (or hike in this problem context) should be explained. The car is driven to a parking area  $P_i$ . A hike is performed, defined by visiting one or more peaks. There are two possibilities for the end of the hike. A loop or cyclical hike is when the hikers return to  $P_i$ , and then continue on to another parking location (or perform another hike from  $P_i$ ). A one-way, or chain, hike is when the hikers end at a different parking area, say  $P_j$ . The chain hike is a key aspect of this problem which differentiates it from other problems in the literature, yet it is a very common occurrence in this application. In order to perform a chain hike, both hikers drive to  $P_j$  (the endpoint of the hike), leaving one car there. Then they drive to  $P_i$ , and leave on the hike. Finishing the hike at  $P_j$ , they drive back to  $P_i$  to retrieve the other car. Figure 2 illustrates a cyclical hike. Figure 3 illustrates a chain hike.

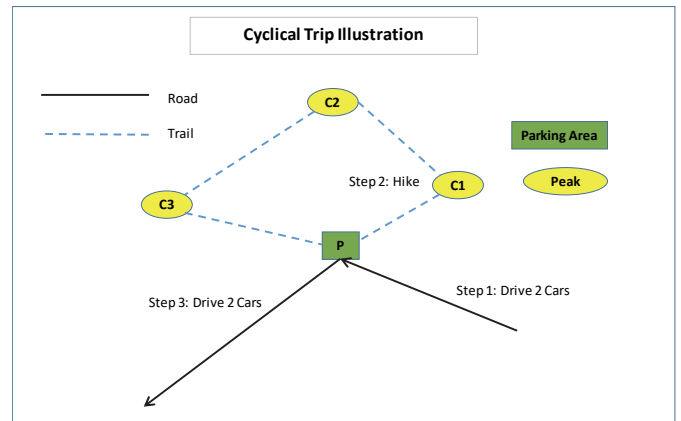


Fig. 2. Illustration of cyclical hike.

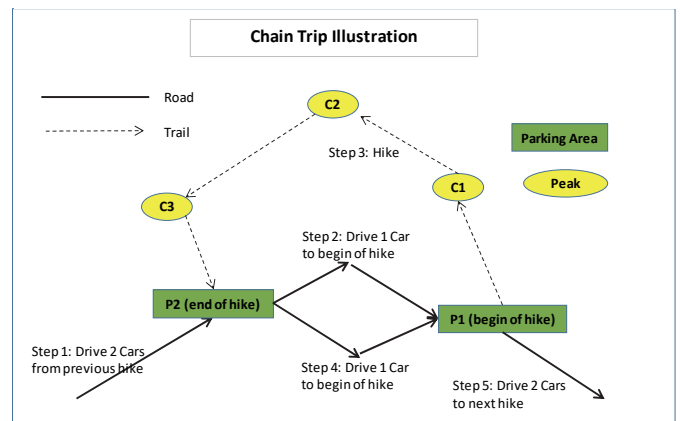


Fig. 3. Illustration of chain hike.

Practical feasibility of chain hikes depends on the application. In the motivating setting, it is common not only to have two hikers (from a safety and logistical aspect), but also that some hikes are more naturally done as one-way hikes. This has the potential to reduce hiking time considerably. Having two cars increases the total mileage driven, though not necessarily the time. Thus, it depends on the objective whether this is an issue. Chain hikes (trips) may be useful in other settings as well, for example, if the “parking” areas are airports, and customers are to be visited on one-way trips between airports. Another example may occur with bike or car sharing programs, where the pick-up and drop-off locations can be different.

For this paper, possible hiking trips are developed in advance. This is reasonable, as there are many hiking guides available and there tends to be a finite number of reasonable hikes. Therefore, the problem becomes one of selecting which hikes to perform (making sure to visit each peak at least once), and then how to sequence the hikes so that the overall objective is optimised. The objective can be time-based or distance-based, or a weighted combination.

Notation is as follows:

- $i = 1, \dots, m$  trip index (hiking trips),
- $j = 1, \dots, n$  customer (peak) index,
- $c_i$  = cost of trip  $i$  (e.g., time, distance),
- $d_{ik}$  = cost of driving from trip  $i$  to trip  $k$  (e.g., distance),
- $a_{ij} = 1$  if trip  $i$  visits customer (peak)  $j$ ; 0 otherwise.

The direction of travel may affect the hike cost (e.g., terrain features). It is common, therefore, to define a hike in each direction, especially for chain hikes (cyclical hikes can be done in the fastest way with no impact on driving logistics). The driving distance matrix  $D$  is a matrix of hike-hike distances. This matrix is in general asymmetric, especially in the presence of chain hikes.

Decision variables for the problem are:

- $x_i = 1$  if trip (hike)  $i$  is selected; 0 otherwise,
- $y_{ik} = 1$  if trip (hike)  $k$  follows trip (hike)  $i$  ( $k \neq i$ ); 0 otherwise.

The formulation of the problem is:

$$\min w_1 \sum_i c_i x_i + w_2 \sum_i \sum_{k \neq i} d_{ik} y_{ik},$$

subject to

$$\sum_i a_{ij} x_i \geq 1 \quad j = 1 \dots n, \tag{1}$$

$$\sum_{k \neq i} y_{ik} - x_i = 0 \quad i = 1 \dots m, \tag{2}$$

$$\sum_{i \neq k} y_{ik} - x_k = 0 \quad k = 1 \dots m, \tag{3}$$

subtour elimination constraints

$$x_i, y_{ik} \in \{0, 1\}.$$

The objective is the weighted sum of the hiking cost and the driving cost. Constraints (1) ensure that every customer (peak) is visited at least once. These are the traditional set covering constraints. Constraints (2) and (3) make sure that the selected hikes are connected to the actual driving network, i.e., if a hike is not selected, no drives should be selected to or from that hike. Similarly, if a hike is selected, there must be one drive to that hike and one drive away from that hike. The subtour elimination constraints ensure that the hikes selected form a full cycle.

Weights on the objective allow for differential importance of the hiking and driving objectives. For this paper, the problem is solved with the hiking objective taking precedence over the driving objective. This allows the problem to be partitioned into a Phase I problem, which is essentially a set covering problem to

select the hikes that minimise the hiking objective (or a number of possible sets of hikes, within some tolerance of the absolute minimum). Then these selected hikes are passed to Phase II, which becomes essentially an asymmetric traveling salesperson problem. For the motivating application, this partitioning is natural, and has the benefit of making the solution approach tractable. The next section discusses the heuristic approach to the solution.

#### IV. IMPLEMENTATION

The heuristic approach was implemented in two ways. The first way was using a traditional programming language (C), along with callable optimisation routines (CPLEX). But, because of the motivating application and the manageable size of the problem, the problem was also developed and addressed implementing a spreadsheet-based approach using Excel and Visual Basic for Applications (VBA). This latter approach, in whole or in part, can be incorporated into classes in Business Analytics and similar courses that cover optimisation. Using the spreadsheet-based model, students can experiment with different solutions and approaches, including side constraints and/or different objectives, and also see the benefit in some cases of a higher-powered solution approach.

Figure 4 provides an overview of the heuristic. Possible hiking trips are pre-defined by the user. Each node in  $V_2$  (i.e., each peak or customer) must be in at least one hike to ensure feasibility. Then the set covering problem (SCP) is solved (Phase I in Fig. 4) to identify the set of hikes that minimises the hiking objective (in this paper, total hiking time). These selected hikes are then sequenced to minimise the total driving time. This in general is an asymmetric traveling salesperson problem (ATSP). In this paper, the well-known Cheapest Insertion heuristic was used to solve the ATSP to ensure quick solution time with good solution quality. This heuristic is sensitive to the starting node, so the heuristic was used from each starting node, and the best solution kept.

**Input:** Network information: parking nodes, distance matrix between nodes, customer (peak) nodes, trip (hike) information (peaks covered, starting/ending nodes, time), tolerance level for covering problem solution.

**Phase I:** Determine trips to perform (hiking objective)

1. Solve Set Covering Problem: find  $Z_{SCP}$ .
2. Find all  $\alpha$ -optimal solutions to the SCP. Store in list, say  $T$  (depth-first search, modified branch and bound procedure).

**Phase II:** Determine sequence of trips (driving objective)

1. For a solution in  $T$ , create the complete directed network  $G_3 = (V_3, E_3)$  as follows:  $V_3$  contains one node for the depot, and one node for every trip in the optimal SCP solution. Edges in  $E_3$  represent the driving distance between trips. Distances between trips are not necessarily symmetric.
2. Solve the asymmetric TSP defined on  $G_3$ .
3. Repeat steps 1–2 for every  $\alpha$ -optimal SCP solution from Phase I.

Fig. 4. Heuristic approach for the BCSP.

The SCP may have alternate-optimal solutions. Because each of these represents a different set of hikes, it is important to consider each solution in order to find the best solution for the hiking objective. Actually, the heuristic was implemented in such a way to find all SCP solutions within  $\alpha$  % of the optimal solution (termed  $\alpha$ -optimal solutions). By varying  $\alpha$ , one can explore the trade-offs (essentially the efficient frontier) between the objectives and choose a solution satisfying the user.

To discover all  $\alpha$ -optimal solutions, a modified branch-and-bound approach was used. A binary search tree was created with an optimal SCP solution as the root node. Depth-first search was used at each node, with each level of the tree representing setting one hike ( $x_i$ ) equal to zero (left branch) or to one (right branch). If the resulting objective to the SCP is greater than  $(1 + \alpha) Z_{SCP}$  (where  $Z_{SCP}$  is the optimal objective value of the set covering problem), or if the SCP is infeasible, that branch of the tree is fathomed at that point. In this way, a list of all  $\alpha$ -optimal solutions of the SCP is found.

Once an  $\alpha$ -optimal trip set is identified, the trips (hikes) must be sequenced to minimise the driving objective (Phase II). Let  $G_3 = (V_3, E_3)$  be the network defined where  $V_3$  is a set of nodes with one node representing each hike (plus one node for the depot), and  $E_3$  represents the distances between pairs of hikes. The distances for the edges in  $E_3$  can be asymmetric, i.e., the distance from hike  $i$  to hike  $k$  ( $d_{ik}$ ) is not necessarily equal to  $d_{ki}$ . This usually occurs in the presence of chain hikes, which start and end at different nodes of  $V_1$ . Recall Fig. 3 and the logistics of a chain hike. For a chain hike, the drives from the end of the hike to the beginning of the hike are termed *intra-hike* driving. Driving between trips is termed *inter-hike* driving, and the sequencing algorithm seeks to minimise inter-hike driving for a given set of hikes (intra-hike driving is a function of the set of hikes, but not to their sequencing).

Let the matrix  $D = [d_{ik}]$  be the inter-hike driving distances between selected hikes (in addition to a home or depot node). In general,  $D$  is asymmetric. This can be due to normal routing reasons (terrain, one-way roads), but mainly in this application due to the logistics of chain hikes. Once  $D$  is defined, the asymmetric TSP is solved to determine the optimal hike sequence. For this research, the well-known Cheapest Insertion Heuristic (CIH) was used. This is a construction heuristic which builds up a subtour until it contains all nodes. At every iteration, it adds the node to the subtour which increases the subtour cost by the least amount. Since the CIH is sensitive with respect to the node it is started from, it was run starting from every node in  $V_3$  and the best tour was retained.

V. COMPUTATIONAL RESULTS

There are 48 peaks above 4000 feet in New Hampshire. Each peak has at least one trail leading to the summit. Based on these peaks, 42 hikes were defined, covering between 1 and 14 peaks, with varying hiking times. Hiking time is expressed in hours. Similarly, there are 25 parking areas identified, and the distance between parking areas was identified.

With  $\alpha = 0$  %, we find just the pure optimal solutions to the SCP (hiking time objective). For this instance, there are eight alternate-optimal solutions to the SCP problem. For each one, the

ATSP was solved to find the minimum driving distance (i.e., the best sequence of the selected hikes). Summary results are shown in Table I. The minimal hiking time was found to be 138.75 hours. The best driving time was found to contain 301 miles of inter-hike driving (i.e., driving from one hike to the next), and 146 miles of intra-hike driving (i.e., driving to coordinate the logistics of chain hikes). Considering that two cars are used for the inter-hike driving portion, the total driving miles is  $301 \cdot 2 + 146 = 748$  miles.

TABLE I  
SUMMARY OF RESULTS FOR  $\alpha = 0$  %

Solution	Hike Time, hrs	Intra-Hike Driving, mi	Inter-Hike Driving, mi	Total Driving, mi
1	138.75	146	307	760
2	138.75	180	362	904
<b>3</b>	<b>138.75</b>	<b>146</b>	<b>301</b>	<b>748</b>
4	138.75	180	368	916
5	138.75	170	303	776
6	138.75	204	351	906
7	138.75	170	303	776
8	138.75	204	357	918

Note that even for the same hiking time, there can be considerable differences in the driving distance. For the eight solutions in Table I, the total weighted driving distance ranges from 748 to 918. Expanding upon this, if  $\alpha$  is allowed to be non-zero, we can relax the hiking time objective in an effort to reduce the driving distance. A set of computational experiments was run at different levels of  $\alpha$  in order to determine how small changes in the hiking time can affect the minimal driving distance. In effect, this is mapping out the efficient frontier between these objectives, i.e., considering Pareto-optimal solutions.

TABLE II  
SUMMARY RESULTS FOR DIFFERENT VALUES OF  $\alpha$   
(tolerance above optimal hiking time; best solution based on total weighted driving distance)

$\alpha$ , %	Number SCP Solutions	Hike Time, hrs	Percent above Optimal Hiking Time, %	Intra-Hike Driving, mi	Inter-Hike Driving, mi	Total Weighted Driving, mi
0.0	8	138.75	0.00	146	301	748
0.5	124	139.34	0.43	158	264	686
1.0	558	139.91	0.84	146	255	656
2.0	2146	141.33	1.86	146	236	618
3.0	5708	142.08	2.40	146	217	580
4.0	14 894	144.25	3.96	140	220	580

Table II displays summary results from the computational experiments. As the hiking objective is relaxed (higher values tolerated), it is possible to reduce the driving objective. Initially there is a large benefit to this. For example, at  $\alpha = 0.5$  %, the hiking objective is allowed to be as large as  $138.750 \cdot 1.005 = 139.440$  hours. Solving the problem by using this parameter results in 124  $\alpha$ -optimal SCP solutions. Solving the associated

ATSP for each one leads to the least-distance driving solution shown. The best solution found has a hiking time of 139.34 hours, and a total weighted driving distance of 686 miles. Comparing to  $\alpha = 0\%$  results, allowing for a 0.5% increase in the hiking objective allows one to reduce the driving objective by  $(748 - 686) / 768 = 8.3\%$ . Further improvements in the driving objective are possible, at a diminishing rate as  $\alpha$  increases. For example, going from  $\alpha = 3\%$  to 4% no longer allows an improvement in the driving objective. Thus, Table II essentially maps out alternatives along the efficient frontier of the two objectives. See Fig. 5 for an illustration of this. Figure 6 shows the relative change in the two objectives as a function of  $\alpha$ . For some people, it may be worthwhile to relax the hiking objective somewhat in order to reduce the driving objective. One other noteworthy item from Table II is the number of  $\alpha$ -optimal solutions as  $\alpha$  increases. Not surprisingly, for higher values, the number of  $\alpha$ -optimal solutions to the SCP increases dramatically.

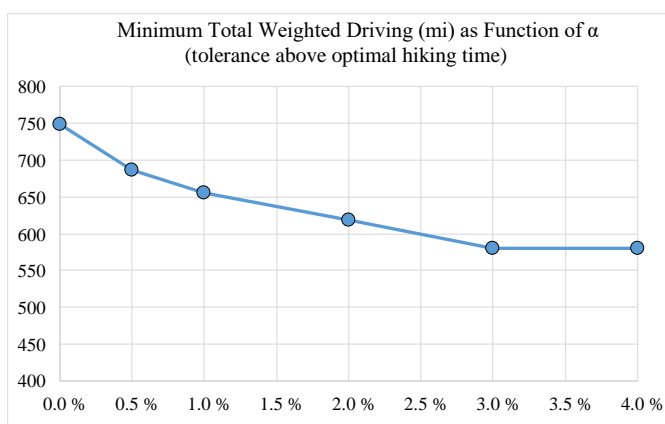


Fig. 5. Percent change from optimal for hiking and driving objectives, as function of  $\alpha$  (tolerance above optimal hiking objective).

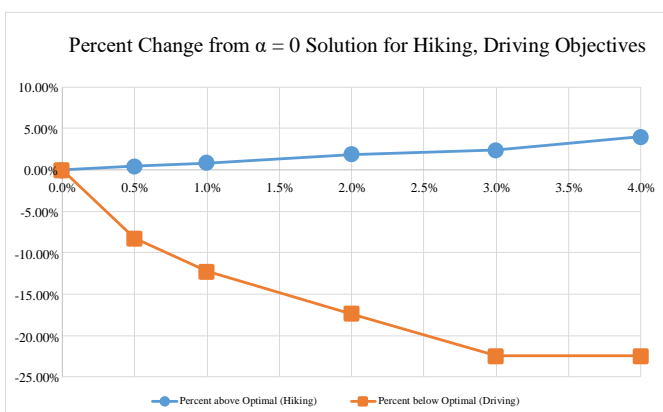


Fig. 6. Optimal total weighted driving as function of  $\alpha$  (tolerance above optimal hiking time).

## VI. CONCLUSIONS AND POSSIBLE EXTENSIONS

This paper introduces a new covering/routing problem termed the Bi-Modal Covering Salesman Problem (BCSP). It is stated as a multi-objective problem, with walking/hiking “cost” (here, time) and driving “cost” (here, distance) as the two objectives. It is similar to, but distinct from, other problems in the literature. The problem was motivated and illustrated based on a hiking problem, but has application to business routing problems as well.

A mathematical programming formulation was given, and a solution algorithm developed using essentially a preemptive goal programming approach. Implementation was done both with C/CPLEX and with Excel/VBA, to allow more accessible usage of the model in introductory Business Analytics courses. In the implementation, the hiking objective is given first priority, and is addressed as a set covering problem (i.e., all peaks/customers must be visited). Once all alternate-optimal (or  $\alpha$ -optimal solutions) are found using a depth-first, branch and bound procedure, the trips (hikes) are sequenced by solving an asymmetric traveling salesman problem.

Results are presented for the base case of  $\alpha = 0\%$ , and then summary results are presented for non-zero values of  $\alpha$ . As  $\alpha$  increases, the hiking time objective worsens, but the driving objective gets better. Essentially, this maps out the efficient frontier between these two objectives, and the user can decide the best trade-off between the two objectives.

There are a couple of possible extensions. Obviously one could use different heuristics for the ATSP. Given the relatively small size of the problem, the one used here is deemed adequate. Another option is to consider a possibility when performing a chain hike. Let us suppose two hikers perform a chain hike, but before driving from the end of the hike back to the beginning, they perform one or more cyclical hikes, starting from the endpoint of the chain hike. This has the potential to decrease the total driving required. The formulation would need to account for these possibilities, and the solution approach altered.

Treating the two objectives simultaneously is another possibility. The problem becomes considerably more complex in this case, because there is no longer the clean decomposition into the SCP and ATSP. For the current application the decomposition makes sense, as there is a relatively small universe of reasonable hikes. Embedding the trip-definition into the formulation and algorithm may yield somewhat better solutions, at the expense of a more complex problem with higher solution difficulty. This is a promising area of future research.

## REFERENCES

- [1] S. D. Smith and M. Dickerman, Eds., “AMC White Mountain Guide with Maps,” in *The Appalachian Mountain Club*, 29<sup>th</sup> edition, 2012.
- [2] N. Jozefowicz, F. Semet and E. G. Talbi, “Multi-Objective Vehicle Routing Problems,” *European Journal of Operational Research*, vol. 189, no. 2, 293–309, 2008.
- [3] G. Gutin, Ed., *The Traveling Salesman Problem and Its Variations*. Springer, 2007.
- [4] D. L. Applegate, R. E. Bixby, V. Chvatal, W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.

- [5] B. Golden, S. Raghavan, and E. Wasil, Eds., "The Vehicle Routing Problem: Latest Advances and New Challenges," *Operations Research/Computer Science Interfaces*, 2008. <https://doi.org/10.1007/978-0-387-77778-8>
- [6] M. Dror, Ed., *Arc Routing: Theory and Solutions*. Springer, 2000.
- [7] K. Braekers, K. Ramaekers and I. Van Nieuwenhuysse, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, vol. 99, pp. 300–313, Sep. 2016. <https://doi.org/10.1016/j.cie.2015.12.007>
- [8] J. R. Current and D. A. Schilling, "The Covering Salesman Problem," *Transportation Science*, vol. 23, no. 3, pp. 208–213, Aug. 1989. <https://doi.org/10.1287/trsc.23.3.208>
- [9] M. Gendreau, G. Laporte, and F. Semet, "The Covering Tour Problem," *Operations Research*, vol. 45, no. 4, pp. 568–576, Aug. 1997. <https://doi.org/10.1287/opre.45.4.568>
- [10] N. Jozefowicz, F. Semet, and E.-G. Talbi, "The bi-objective covering tour problem," *Computers & Operations Research*, vol. 34, no. 7, pp. 1929–1942, Jul. 2007. <https://doi.org/10.1016/j.cor.2005.07.022>
- [11] M. Fischetti and P. Toth, "An Additive Approach for the Optimal Solution of the Prize-Collecting Traveling Salesman Problem," in *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad Eds. Amsterdam: North-Holland, pp. 319–343, 1988.
- [12] E. Balas, "The prize collecting traveling salesman problem," *Networks*, vol. 19, no. 6, pp. 621–636, Oct. 1989. <https://doi.org/10.1002/net.3230190602>
- [13] B. L. Golden, L. Levy and R. Vohra, "The Orienteering Problem," *Naval Research Logistics*, vol. 34, 307–318, 1987. [https://doi.org/10.1002/1520-6750\(198706\)34:3<307::aid-nav3220340302>3.0.co;2-d](https://doi.org/10.1002/1520-6750(198706)34:3<307::aid-nav3220340302>3.0.co;2-d)
- [14] R. Ramesh, Y.-S. Yoon, and M. H. Karwan, "An Optimal Algorithm for the Orienteering Tour Problem," *ORSA Journal on Computing*, vol. 4, no. 2, pp. 155–165, May 1992. <https://doi.org/10.1287/ijoc.4.2.155>
- [15] L. Levy and L. Bodin., "Scheduling the Postal Carriers for the United States Postal Service: An Application of Arc Partitioning and Routing," in *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad Eds. Amsterdam: North-Holland, pp. 359–394, 1988.
- [16] E. Gussmagg-Pflegel, F. Tricoire, K. F. Doernery, R. F. "Hartl, Mail-Delivery Problems with Park-and-Loop Tours: A Heuristic Approach" in ORP3 Meeting, Cadiz. September 13–17, 2011.
- [17] A. Černá, J. Černý, F. Malucelli, M. Nonato, L. Polena, and A. Giovannini, "Designing Optimal Routes for Cycle-tourists," *Transportation Research Procedia*, vol. 3, pp. 856–865, 2014. <https://doi.org/10.1016/j.trpro.2014.10.064>
- [18] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems," *Journal of Heuristics*, vol. 20, no. 3, pp. 291–328, Mar. 2014. <https://doi.org/10.1007/s10732-014-9242-5>

**Roger B. Grinde** received his Ph.D. in Industrial Engineering & Operations research from the Pennsylvania State University, USA (1993). His M.S. in Industrial Engineering/Operations research is from Oregon State University (1986), and he earned his B.A. in Mathematics from Carroll College, MT, USA (1984). He is an Associate Professor of Management Science in the Peter T. Paul College of Business & Economics at the University of New Hampshire. He currently serves as the Chair of the Decision Sciences Department. Previously he served as an Associate Dean for Undergraduate Programs, as the Faculty Fellow for Accreditation and Assurance of Learning at the Paul College, and has held many other service/administrative roles. Prior to UNH, he was a Research Engineer at Battelle/Pacific Northwest National Laboratory. He is a member of INFORMS and DSI. He has received teaching awards, as well as a best paper award from the Institute for Industrial Engineers. Address: Paul College of Business & Economics, University of New Hampshire, 10 Garrison Road, Durham, NH, USA 03824. E-mail: roger.grinde@unh.edu