ARTICLE IN PRESS

Transportation Research Part C xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Generalized Maximum Benefit Multiple Chinese Postman Problem

Ali Shafahi *, Ali Haghani

Department of Civil and Environmental Engineering, University of Maryland, College Park, MD, USA

ARTICLE INFO

Article history:
Received 16 October 2014
Received in revised form 6 January 2015
Accepted 22 January 2015
Available online xxxx

Keywords:
Chinese Postman Problem (CPP)
Arc routing problems with profit
Multiple vehicle routing
Security patrolling

ABSTRACT

This research is focused on a generalization on the Max Benefit Chinese Postman Problem and the multiple vehicle variant of the Chinese Postman Problem. We call this generalization, the Generalized Maximum Benefit k-Chinese Postman Problem (GB k-CPP). We present a novel Mixed Integer Programming (MIP) formulation for the GB k-CPP. Four different cases of the model are discussed. The first case, performs arc-routing with profits and assumes that the origin and destination for each vehicle is the same for each cycle and is given by the user. The next case relaxes the assumption that the origin and destination for each vehicle should be the same and allows the users to select possible origins/destinations for vehicles. Case three gets the origin for each vehicle as input and produces a solution based on finding the best destination for each vehicle. The last case, that is very general, allows the optimization model to select possibly different locations for vehicle origin and destination, during each cycle. The different cases are applied to a security patrolling case conducted on the network of University of Maryland at College Park campus and the results are compared.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Public safety is one of the big concerns of cities and communities. One possible way to increase safety and decrease the occurrence of crimes is patrolling. In the presence of police patrol, the number of crimes would decrease (Sherman and Weisburd, 1995). To have an efficient patrolling system, the authorities, in most cases, are interested in finding optimum routes for the vehicles. Optimum routes are those which are able to cover the area more thoroughly.

Research related to routing problems mainly fall into two categories of either node routing problems or arc routing problems. One difference between these two is coverage. In arc routing problems, the objective is to find a rout in a graph which covers all or a subset of arcs. In node routing problems, the objective is to find a route which covers all or a subset of nodes. The most famous examples in arc routing and node routing problems are Chinese Postman and Travelling Salesman problems.

The majority of mathematical modeling done in the past has focused on node routing problems this is while there are many important applications for arc routing problems. In cases that more coverage is desired, arc routing problems can be more suitable. For instance in Fig. 1, all of the nodes can be covered by going through this path: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$. However,

* Corresponding author at: Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20740, USA. Tel.: +1 (301) 789 9787.

E-mail address: ashafahi@umd.edu (A. Shafahi).

http://dx.doi.org/10.1016/j.trc.2015.01.017
0968-090X/© 2015 Elsevier Ltd. All rights reserved.

Please cite this article in press as: Shafahi, A., Haghani, A. Generalized Maximum Benefit Multiple Chinese Postman Problem. Transport. Res. Part C (2015), http://dx.doi.org/10.1016/j.trc.2015.01.017

if this path is taken, one of the two arcs labelled "1–3" will be left uncovered. The entire coverage however can be achieved by covering all arcs and for example going through this path: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 3 \rightarrow 1$.

Vehicle patrolling problems require more coverage hence they should be modeled as arc routing problems. Routing of security guards (Wolfer Calvo and Cordone, 2003; Willemse and Joubert, 2012; Shafahi and Haghani, 2015), school bus routing (Delgado and Pacheco, 2001), delivery of newspapers to customers (Applegate et al., 2002), waste collection (Lacomme et al., 2001; Filipiak et al., 2009), search and rescue teams (Zhuang et al., 2007), and snowplow routing (Rao et al., 2011) are some other applications that have been modeled as arc routing problems.

As mentioned earlier, Chinese Postman Problem (CPP) is among the most famous arc routing problems. CPP was first coined by Mei-Ko (1962). In this problem, the objective is to find a tour such that all of the edges (arcs) of a graph have been covered at least once within minimum cost. The Rural Postman Problem (RPP) is a related problem which assumes that not all of the arcs should be covered at least once and only a subset of them require coverage. Since covering all arcs is one of the goals of patrolling the backbone of our model is based on CPP. However CPP generally have some limiting assumptions including: The route designed is for a single vehicle; and routes are in closed loops. For the CPP problem an additional implied assumption is that all of the arcs have the same priorities and weights.

In reality, especially in the presence of historical data, arcs and nodes have priorities. For example in the case of snow plowing, some of the major roads should be cleared more often either due to heavier snow or since they carry a big load of traffic. For the example of patrolling to increase safety, some of the roads should be patrolled more often because historically they have been more prone to crimes and incidents. The classic CPP and RPP fail to fully model the reality that each arc generally has both a profit and cost associated with it.

The remaining of the paper is organized as follows. First, we review some of the literature related to arc routing problems with profit and arc routing problems with multiple vehicles. In Section 3, we describe the modeling and present the mathematical formulation for four different cases. Next in Section 4, we discuss the real world example which is patrolling for increasing security in the UMD campus. Then we apply the models to the example and present the results. Finally we present the summary and conclusion for this research.

2. Literature review

In the literature two fields that assign both profits (weights) and costs to arcs are called CPP with profit, and arc routing problems with benefit. Two variations of the CPP which consider weights are the Priority constrained CPP (PCCPP) defined in (Kramberger and Zerovnik, 2007) and the Max Benefit CPP (MBCCP) defined in (Malandraki and Daskin, 1993). In PCCPP weights are assigned to nodes and the objective is to cover all arcs such that the nodes with higher priorities are visited sooner. MBCPP assumes that weights (profits) and costs are associated with arcs and the objective is to collect as many weights possible. It is worth noting that even though MBCPP is for a one vehicle case it is still a hard problem to solve. Pearn and Wang (2003) prove that MBCPP is more complex than RPP by presenting a linear transformation of the RPP into a special case of the MBCPP.

Given that generally more than one vehicle will be serving a network especially if the network is not very small, a portion of the research related to arc routing problems is related to fleet size (Ulusoy, 1985). The k vehicle variant of the CPP is called k-CPP. In k-CPP the objective is to cover the all the arcs in the network at least once using a collective of k vehicles with the minimum cost. However, a more common objective in the case of k-CPP is balancing the workload among the multiple vehicles. Min–Max k-CPP (MMCPP) is a category of problems which tackles the problem of workload balancing by minimizing the maximum cost incurred by any vehicle. Three other variants with the goal of balancing workload are mentioned in Osterhues and Mariak (2005). Benavent et al. present a min–max model for balancing workload in the Windy Rural Postman Problem (Benavent et al., 2009). Due to the complexity of finding solutions, a big portion of the studies related to MMCPP have been dedicated to developing solution algorithms. Some of the solution methods and algorithms used are Tabu-search (Ahr and Reinelt, 2006; Willemse and Joubert, 2012), Branch and Cut (Benavent et al., 2009) (Corberan et al., 2007), Ant Algorithm (Degenhardt, 2004), and Genetic Algorithm (Rao et al., 2011).

All these research related to MMCPP ignore profits as their objective function is to minimize the maximum cost and not maximizing the total (profit) weight collected. However, another group of studies exist which take into consideration profits

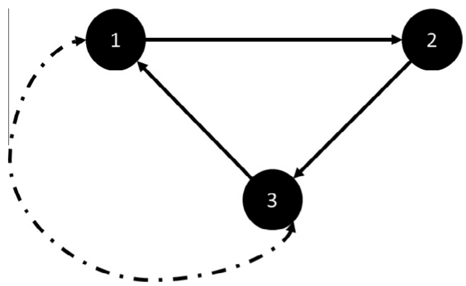

Fig. 1. Coverage difference between arc routing problems and node routing problems.

associated to arcs and cases for multiple vehicles. This group of studies are called Arc routing problems with profit with multiple vehicles. Three classes exist for such studies: Profitable Arc Tour Problem, Undirected Capacitated Arc Routing Problems with Profit, and Team Orienteering Arc Routing Problems. A brief explanation of each is presented in the following.

Feillet et al. introduced the Profitable Arc Tour Problem (PATP) (Feillet et al., 2005). The objective in PATP is to find a set of cycles for vehicles that maximizes the collection of profit minus costs. In PATP, all the demands do not need to be satisfied. The mathematical formulation presented in their work is a master problem formulation. In this formulation, the candidate routes (called collection cycles) are generated using the sub-problem. The objective of the master problem is to identify how many time each candidate route should be traversed. This approach for modeling demands many preprocessing and calculations because enumerating all possible combinations for each route in the sub-problem stage is a very time-consuming task.

Undirected Capacitated Arc Routing Problem with Profits (UCARPP) first stated in Archetti et al. is another class of problems that assumes that not all of the arcs have to be served (Archetti et al., 2010). It restricts service to each arc to be no more than one time. In terms of mathematical modeling a master problem approach is utilized. In this study the assumption that the origin and destination for all vehicles are the same node is imposed. This assumption can significantly reduce the number of candidate routes.

The Team Orienteering Arc Routing Problem is another example that was introduced by Archetti et al. in which the goal is to cover a subset of given arcs and among the other arcs, cover those which are more profitable (Archetti et al., 2013). This problem again assumes that the origin and destination for all vehicles are the same and is given. If we assume that all arcs have to be covered, and we ignore workload balancing among vehicles, the problem becomes similar to the first case presented in our paper. However, our model is more generalized because it allows the vehicles to have different origin/destinations.

All these three afore-mentioned classes of problems somehow assume that all of the arcs of the system do not necessarily need to be served. However, for cases such as patrolling, covering all roads in a network seems more adequate. That is why we have used the k-CPP as the backbone of this study. Another common assumption in two out of three of these studies is that they assume that the origin (depots) and destinations are the same for all vehicles. In this research we show how the users of our model can benefit by relaxing this assumption which is a common assumption in the k-CPPs as well. One of the rare works done in k-CPP which allows for different depots for different vehicles is (Platz and Hamers, 2011). In their work, they solve the game of assigning vehicles to depots. Another assumption they use is that each vehicle loops one time over a cycle and at the end returns to its own depot. Overall, not many research has been devoted to problems which require a minimum coverage of more than once during each period. Most of the k-CPP problems were modeled to cover each road only once. In our research we consider covering the road at least n times during each time period to increase area security. If we set $n = 1$ the problem is then similar a maximum benefit k-CPP problem. In practice, in cases such as patrolling or snowplowing, the objective is to cover the arcs of a network multiple times during each time period. This generalization relaxes some of the constraints on the problem and therefore may increase the feasible region and possibly improve the systems performance. In this research we enlighten the areas that can improve the performance and propose appropriate modeling for each case. In addition the developed model is equipped with workload balancing. The model developed in this research a Generalized Balanced multiple vehicle (k) Chinese Postman Problem with Profit (GB k-CPPP). The presented mathematical model helps public authorities in routing vehicles such that within a time frame, each arc is covered at least n times and the arcs with more profit are covered as many times possible within its allowable range.

3. Modeling

The mathematical model presented in this research for the Generalized Maximum Benefit k-Chinese Postman Problem (GB k-CPP) is based on covering each arc at least n times. If each arc has to be covered at least n times during each shift, T, we formulate the problem such that each arc is covered at least once during each cycle duration, T/n. The vehicle then loops over its cycle n times to meet the problems objectives. This approach is beneficial because the latest an edge (arc) will be visited will be no later than T/n as opposed to the original T. Another advantage of this approach is in the timespan between two simultaneous visits to an edge. The maximum duration between each two simultaneous visits of any edge will be T/n. Many other advantages result from this approach. Some of which are: opportunity for recourses and updates, smaller and more understandable solutions.

In the following subsections, we provide four different cases for the maximum benefit k-CPP. Each case has its assumptions which makes it applicable and useful for different situations. As we proceed, we relax more assumptions and make the model more general as the last case is the most general case among all. Initially, we will propose a basic model that the vehicles' cycles are in the shape of closed loops. In other words, the origin and destination for each vehicle in each cycle is the same and given by the user (case 1). Secondly, we extend the model by modifying it such that the origin and the destination do not necessarily need to be the same but the possible origin and destination should be given by the user (case 2). Case 2 allows each vehicles' cycle to be a direct path and not a closed loop. Then, we modify the mathematical formulation such that the model itself can find the optimal location for the destination of each cycle after being given the origin for each vehicle as input (case 3). Finally, the most general case is presented (case 4). In this general case, the mathematical model is formulated in a way that the most optimal origin and destinations for each cycle of each vehicle is an output of the model and the users do not enforce either the origin or the destination to the model.

Case 1 can be used for situations such that the vehicles need to return to their already existing depot after each cycle for reasons such as reporting. Case 2 is useful for situations that each vehicles has two predetermined depots and should start and end at either one of them. Case 3, which is more general, assumption is that each vehicle has an initial depot and it is only required to return to that depot at-least once every two cycles. Case 4, which is the most general case, does not assume a fixed origin or destination for each vehicle. This last case is useful for situations that the vehicle is only required to come back to its main depot at-least once each shift (n cycles).

3.1. Origin and destination the same and given by users (case 1)

This case is similar to what has been mainly studied in previous research in the field of arc-routing problems with profit and k-CPP. The major assumption for this case is the origin for each vehicle is given and the destination for that vehicle after one cycle is the same as its origin. The origin and destination being the same and given is useful for situations which the vehicle has to return to its depot after each cycle to either report or refuel. For convenience, the decision variables and parameters used in this model are summarized in Table 1.

The entire formulation for case 1 can be seen in the following:

$$\text{maximize } z = \sum_i \sum_j \sum_k W_{ij} x_{ijk} \tag{1}$$

Subject to:

$$\sum_k x_{ijk} + x_{jik} \geqslant 1 \quad \forall i,j|(ij) \in A \tag{2}$$

$$\sum_i x_{ilk} = \sum_j x_{ljk} \quad \forall l \in V, k \tag{3}$$

$$\sum_i \sum_j T_{ij} x_{ijk} \leqslant T_{max} \quad \forall k \tag{4}$$

$$x_{ijk} \leqslant Cap\_Vehicle \quad \forall i,j,k \tag{5}$$

$$\sum_k x_{ijk} \leqslant Cap\_Arc \quad \forall i,j \tag{6}$$

$$\sum_j x_{jik} + \sum_j x_{ijk} \leqslant 2Mb_{ik} \quad \forall i,k \tag{7}$$

$$\sum_j y_{ijk} - \sum_j y_{jik} = -1 b_{ik} \quad \forall i \in (V - O_k), k \tag{8}$$

$$y_{ijk} \leqslant M x_{ijk} \quad \forall i,j,k \tag{9}$$

$$d_k = \sum_i \sum_j T_{ij} x_{ijk} \quad \forall k \tag{10}$$

$$\bar{d}(1+\alpha) \geqslant d_k \quad \forall k \tag{11}$$

$$\bar{d}(1-\alpha) \leqslant d_k \quad \forall k \tag{12}$$

$$x_{ijk} \geqslant 0 \text{ and integer} \quad \forall i,j,k \tag{13}$$

$$b_{ik} \in \{0,1\} \quad \forall i,k \tag{14}$$

$$y_{ijk} \geqslant 0 \quad \forall i,j,k \tag{15}$$

The objective function (1) maximizes the total weight covered by all vehicles. Constraints (2) assure that each arc that exists is covered at least once during each cycle regardless of its direction using all vehicles. Constraints (3) are for conservation of flow at each node of the network for each vehicle. Constraints (4) are the time limitation per cycle for each vehicle. Constraints (5) are caps on the number of times each vehicle can cover each arc. These constraints are used to eliminate undesignated behaviors such as a single vehicle traversing an arc so many times. Constraints (6) are also set to eliminate focusing on an arc too much overall using all vehicles.

Constraints (7)–(9) are added to eliminate sub-tours and discontinuities. Constraints (7) relate b_{ik} with x_{ijk}. Constraints (8) are the main constraints enforcing route connectivity (continuity) for each vehicle. These constraints state that one unit of the artificial commodity should be left at each node that the vehicles visit excluding the origin of each vehicle (O_k). Constraints (9) ensure that there cannot be any flow of the artificial commodity in any arc if the vehicle is not traversing that arc.

The M in constraints (7) and (9) is a large number that acts as big-M. Selecting the M to be a very large number may cause rounding errors. It is advised to work with smaller numbers. $K \times Cap\_Arc$ is a large enough number where K is the maximum number of vehicles. Constraints (10)–(12) are in charge of balancing the workload among vehicles. Constraints (13)–(15) are restrictions on the variables.

It is worth noting that this model can also assist the users in finding the minimum number of vehicles which can perform the patrolling job. The minimum number can be found by performing sensitivity analysis on the number of vehicles, K. If we find a feasible solution for K' we should replace K' with $K' - 1$. This procedure should be iterated until we find the smallest

Table 1
Mathematical model parameters and variables.

Variables and parameters for case 1	
x_{ijk}	The main decision variable which represents the number of times arc ij is traversed in each cycle using vehicle k starting from node i ending at node j. This variable is integer
y_{ijk}	A dummy variable used for eliminating sub-tours. In charge of the flow of fictional commodity being transferred from node i to node j using vehicle k. This variable is continuous
b_{ik}	Another dummy variable for eliminating sub-tours. This variable is binary and equals one if node i is visited by vehicle k and zero otherwise
d_k	Duration it takes vehicle k to cover its route one time
$\bar{d}$	Average route duration for one vehicle
W_{ij}	The weight for traversing arc ij starting from node i ending at node j
T_{ij}	The duration for traversing arc ij starting from node i ending at node j
T	The duration of each shift. The total time for covering the entire network as many times as possible beyond requirement
n	The minimum number of times each arc has to be traversed regardless of the direction in each shift, T
T_{max}	The maximum time available in each cycle for the vehicles to cover all of the edges in the network at least once ($T_{max} = \frac{T}{n}$)
O_k	Origin for vehicle k
V	Set including all nodes in the network
M	A large number
Cap_Vehicle	The maximum number of times per cycle any vehicle is allowed to cover arc ij going from i to j
Cap_Arc	The maximum number of times any arc can be traversed in each direction per cycle using all vehicles
α	Parameter for balancing workload (allowable deviation from average route time for vehicles)
Additional parameters for case 2	
D_k	The possible destination for vehicle k at the end of its cycle
Additional variables for case 3	
u_{ik}	Dummy binary variable used for case 3 and case 4. u_{ik} equals zero if node i is the origin or potential destination for vehicle k, and one otherwise
q_{ik}	Dummy continuous variable used for case 3 and case 4 that acts as a slack variable
Additional variables for case 4	
c_{ik}	Dummy binary variable used for case 4. c_{ik} equals one if continuity of flow is enforced for vehicle k at node i and zero otherwise
f_{ik}	Dummy continuous variable used for case 4 as a slack variable

value for K such that the problem is feasible ($K = K^*$). For any $K < K^*$ the problem will be infeasible. K^* is the minimum number of vehicles required for doing the job.

3.2. Origin and destination not necessarily same but both given by user (case 2)

For this case we allow the vehicles' routes to be direct paths if more beneficial. The routes per cycle no longer need to necessarily be closed cycles. If the possible destination is known, the conservation of flow at nodes constraints (3), should be for all nodes excluding the origin and destination. Not looping for each cycle has many potential benefits. Eliminating the need for covering arcs with large costs and relatively small profit more than once along with more freedom in decisions and routes are among its benefits. Allowing the origins and destinations to be different reduces the overall cost per cycle for all vehicles which itself results in finding more feasible solutions. This can potentially decrease the number of vehicles required for covering the entire network.

For example assume Fig. 2 is the network which we require to cover all of its arcs at least once. The number close to each arc represents the cost associated with that arc. If the vehicle has to complete a loop and its origin is node 1, the optimal path with the least cost will be $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 5 \to 3 \to 4 \to 1$ with the total cost of 544. However, if we allow the destination of the vehicle to be node 6, the optimal path will then be $1 \to 2 \to 3 \to 5 \to 4 \to 1 \to 4 \to 3 \to 5 \to 6$ with the total cost of 446 per cycle. This is a reduction of 98 units of cost per cycle. The overall savings for this problem will be equal to $98n$. Note that If the number of cycles, which is equal to the minimum number of times each arc has to be covered (n), is even, the vehicle will return to its origin at the end of its shift (node 1 in this example).

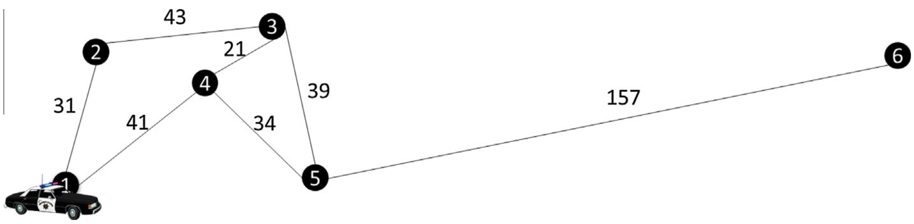

Fig. 2. Example illustrating the potential benefit in allowing te routes to be direct paths instead of closed loops.

Please cite this article in press as: Shafahi, A., Haghani, A. Generalized Maximum Benefit Multiple Chinese Postman Problem. Transport. Res. Part C (2015), http://dx.doi.org/10.1016/j.trc.2015.01.017

Case 2 is useful in situations that budget (resources) are limiting and some arcs have large costs and due to their location have to be covered multiple times in case looping in each cycle is enforced. Given that the assumption for this case is that the origin and destination for each cycle of each vehicle is known prior to running the model, the only modification required is in the conservation of flow constraints (3). These constraints should be for all nodes but the origin and the destination for each vehicle (3′).

$$\sum_i x_{ilk} = \sum_j x_{ljk} \quad \forall l \in (V - O_k - D_k), k \tag{3'}$$

Note that by this modification, the mathematical model of (1), (2), (3′), (4)–(15) might still contain a loop in each cycle. However, there is no necessity for looping anymore.

3.3. Origin and destination different-origin given destination free (case 3)

If the destination for each cycle is not known or does not need to be enforced, we can allow the destinations to be selected by the optimization problem based on the resources (time and cost) available. To do so, we need to say that the conservation of flow constraints (3) are valid for all nodes except the origin for each vehicle, O_k, and one other node which vehicle k does cover ($b_{ik} = 1$ for that specific node i). To do this, we need to introduce a dummy binary variable, u_{ik}, which states whether constraint (3) is holding for node i for vehicle k (=1) or not (=0) and another dummy variable, q_{ik}, which is continuous and acts as slack. Constraint (3) should be for all nodes covered by each vehicle excluding its cycle's origin and destination ($u_{O_k k} = u_{D_k k} = 0 \quad \forall k$). Constraints (16)–(22) replace constraints (3) in the main formulation in this case.

$$u_{O_k k} = 0 \quad \forall k \tag{16}$$
$$u_{ik} \leqslant b_{ik} \quad \forall i, k \tag{17}$$
$$\sum_i b_{ik} - 2 = \sum_i u_{ik} \quad \forall k \tag{18}$$
$$\sum_i x_{ilk} - \sum_j x_{ljk} = M(1 - u_{lk}) - q_{lk} \quad \forall l \in V, k \tag{19}$$
$$q_{ik} \leqslant 2M(1 - u_{ik}) \quad \forall i, k \tag{20}$$
$$q_{ik} \geqslant 0 \quad \forall i, k \tag{21}$$
$$u_{ik} \in \{0, 1\} \quad \forall i, k \tag{22}$$

Constraints (16) ensure that the origin for each vehicle is one of the points where the conservation of flow is not needed. Constraints (17) assure that the conservation of flow constraint is only for those nodes which are being covered by each vehicle. Constraints (18) ensure that for each vehicle there are only two nodes which the vehicle covers and conservation of flow does not need to be valid.

Constraints (19) are the modified and sophisticated conservation of flow constraints. Multiplying $(1 - u_{lk})$ by M will eliminate the need for introducing another dummy variable to take care of the positive and negatives values for the left-hand-side. Constraints (20) restrict the dummy slack variable, q_{ik}, to assume values only for nodes for which conservation of flow is not required. The entire formulation will be 1, 2, (4)–(22).

3.4. Origin and destination different, general case (case 4)

In case 4, in addition to the destination, the origin per cycle is an output of the mathematical model. Consequently, the size of the feasible region of this problem will expand. Potentially leading to better solutions.

Fig. 3 depicts the potential benefits of allowing the origin to be flexible. The assumption in this example is that the two vehicles have to cover all of the arcs of the network and return to their depot (origin). The total budget available for each vehicle is 220 units. In the example illustrated at the left, vehicle one's origin is enforced to be node 1 and vehicle two's origin is node 6. The path with the least cost for vehicle one is $1 \to 6 \to 5 \to 2 \to 1$ and the path for vehicle two is $6 \to 2 \to 3 \to 4 \to 5 \to 6$. This makes the problem infeasible because the cost for vehicle two is 242 which is more than the available budget, 220. However, if we do not force any restrictions on the location of vehicle two and allow the second vehicle's origin to be node 4, the problem will become feasible. The optimal route for vehicle one in this case is $1 \to 6 \to 5 \to 6 \to 2 \to 1$. And the optimal route for vehicle two is $4 \to 3 \to 2 \to 5 \to 4$. The cost for vehicle one and two will then be 193 and 213, respectively.

Case 4 is useful when we want to cover an entire area a couple of times but we do not have any restrictions for the starting point of vehicles. Or, the time spared for going from the main depots that the vehicles have to start at the beginning of their shift and the suggested cycle starting point of the optimization model is negligible compared to the total allowable time the vehicle will be cycling during its shift, T.

Formulation wise, since the origin is not predetermined, we let the optimization problem select the two nodes at which the conservation of flow is not required to be held (the origin and destination) by itself. The difference in mathematical

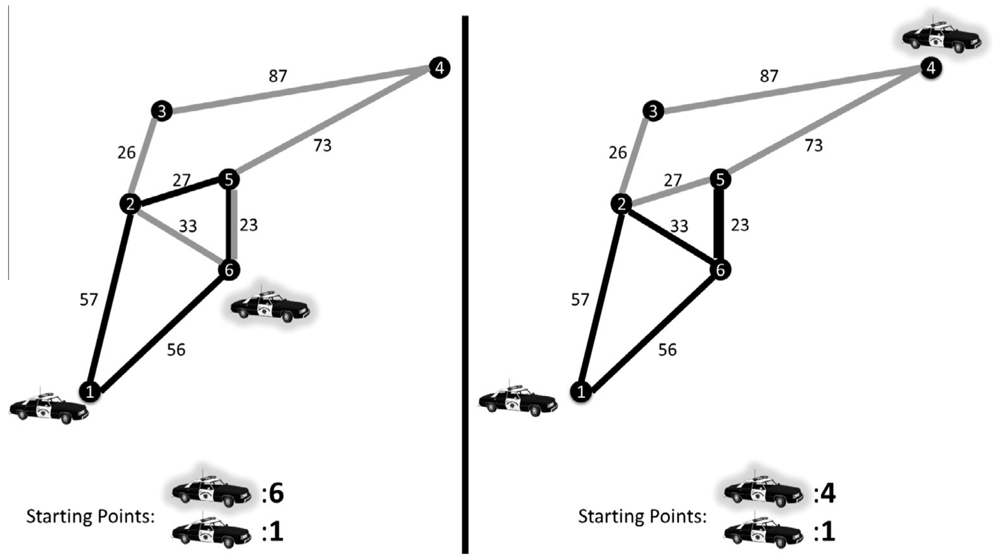

Fig. 3. Potential benefit of allowing the origin and destination per cycle for each vehicle to be optimized.

modeling compared to case 1 is in the conservation of flow constraints (3) and the continuity constraints (8). The new sets of conservation of flow constraints are constraints (17)–(22).

To reformulate the connectivity (continuity) constraints, we need to add another binary variable, c_{ik}. c_{ik} is one if the continuity constraint is required for vehicle k at node i and zero otherwise. In addition, f_{ik} which is a dummy slack variable is also needed. The reformulated continuity constraints will then be as (23)–(30).

$$c_{ik} \geqslant b_{ik} + u_{ik} - 1 \quad \forall i, k \tag{23}$$

$$\sum_i c_{ik} = \sum_i b_{ik} - 1 \quad \forall k \tag{24}$$

$$c_{ik} \leqslant b_{ik} \quad \forall i, k \tag{25}$$

$$\sum_j y_{ijk} - \sum_j y_{jik} = -1c_{ik} + f_{ik} \quad \forall i \in V, k \tag{26}$$

$$f_{ik} \leqslant Mb_{ik} \quad \forall i, k \tag{27}$$

$$f_{ik} \leqslant M(1 - c_{ik}) \quad \forall i, k \tag{28}$$

$$f_{ik} \geqslant 0 \quad \forall i, k \tag{29}$$

$$c_{ik} \in \{0, 1\} \quad \forall i, k \tag{30}$$

Constraints (23) assure that connectivity (continuity) constraints are valid for those nodes that are being covered by each vehicle and at the same time are not the cycle origin or destination. Constraints (24) make sure that connectivity (continuity) constraints hold for all points that are covered by each vehicle but for one point (the origin). Constraints (25) prevent the enforcement of continuity constraints for those nodes that are not covered by each vehicle. Constraints (26) are the major new modified continuity constraints. Constraints (27) prevent the slack variable of getting any values if node i is not covered by that vehicle. Constraints (28) prevent the slack variable of assuming any value if continuity constraints should not be enforced for its corresponding node and vehicle. The complete formulation for this case will be 1, 2, (4)–(7), (9)–(15), (17)–(30).

4. Data gathering

The main input parameters of the model presented in this paper are the traversal duration of each arc and the collective crime weight for each arc. The traversing durations were calculated based on the length of its respective arc and its speed limit.

The legitimate assumption for weights of roads (arcs) was that the weights depend on the number and type of crime that happen within its vicinity. The vicinity of an arc was set to be 400 ft. If a crime was within the 400 ft range of multiple arcs, the crime was assigned to that arc which had the minimum distance to the crime. To accommodate different shifts for patrolling vehicles, the day was broken into four six hour periods starting from 12 a.m. A different crime weight was calculated for each time period, T.

To calculate the collective crime weights we gathered data reported by the campus police department. We looked into crimes that happened between November 1st 2013 and April 1st 2014. Eq. (31) was used for calculating the collective crime weights for each arc (road).

$$W_{i,T} = \sum_{k} \left(\int_{T_S}^{T_F} (N_{i,k,t} \times M_k) dt \right) \tag{31}$$

where $N_{i,k,t}$ is the number of type k incidents which happened at time t and is assigned to arc i, M_k is the weight of incident type k (see Table 2), T_S and T_F are the start and end time for each period T, respectively, and $W_{i,T}$ is the collective weight for arc i for time period T.

Table 2 shows the assumed weights for each type of crime. These weights were assumed based on the severity of the type of crime.

We constructed the campus network by including the main roads of the campus and by placing a node at each intersection between the main roads. At these intersections, the vehicles were allowed to make a U-turn and/or change their road. Overall the constructed network had 12 nodes and 36 arcs (18 arcs which have two directions). The network can be seen in Fig. 4.

The heat-map of the crime weights for the first time period (12–6 a.m.) is shown in Fig. 5.

Table 2
Weights assigned to each occurrence of a crime type, k.

ID (k)	Name	Weight (M_k)
1	Assault	3
2	Other Sexual Offense	2.5
3	Robbery	2
4	Burglary	1.5
5	Theft from vehicle	1.5
6	Theft	1

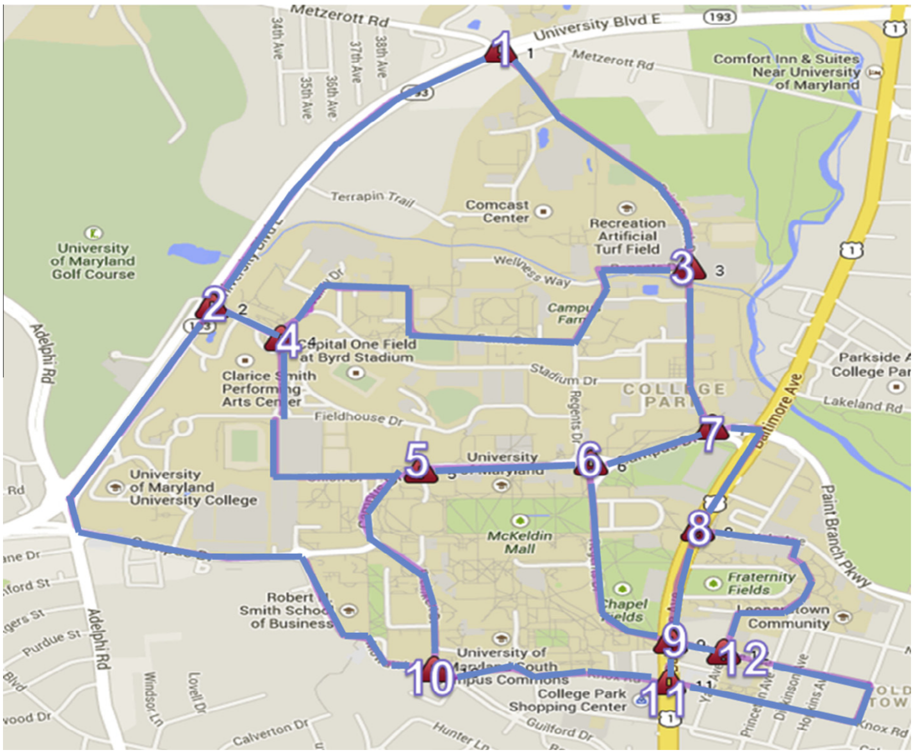

Fig. 4. University of Maryland campus constructed network.

Please cite this article in press as: Shafahi, A., Haghani, A. Generalized Maximum Benefit Multiple Chinese Postman Problem. Transport. Res. Part C (2015), http://dx.doi.org/10.1016/j.trc.2015.01.017

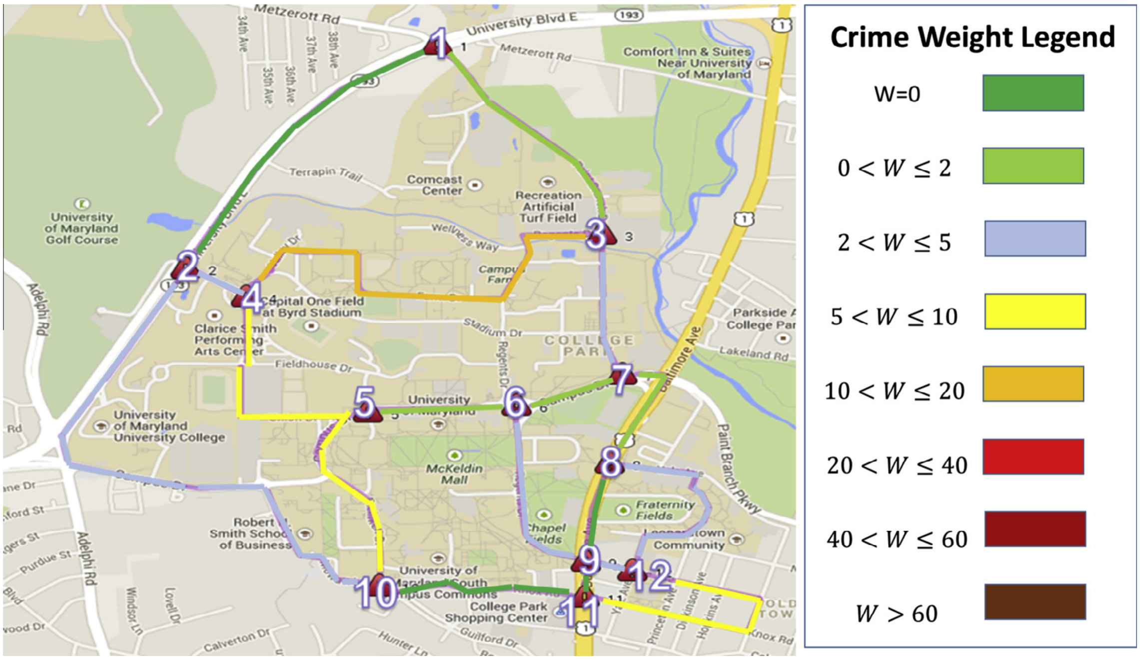

Fig. 5. Crime weight heat-map for 12–6 a.m.

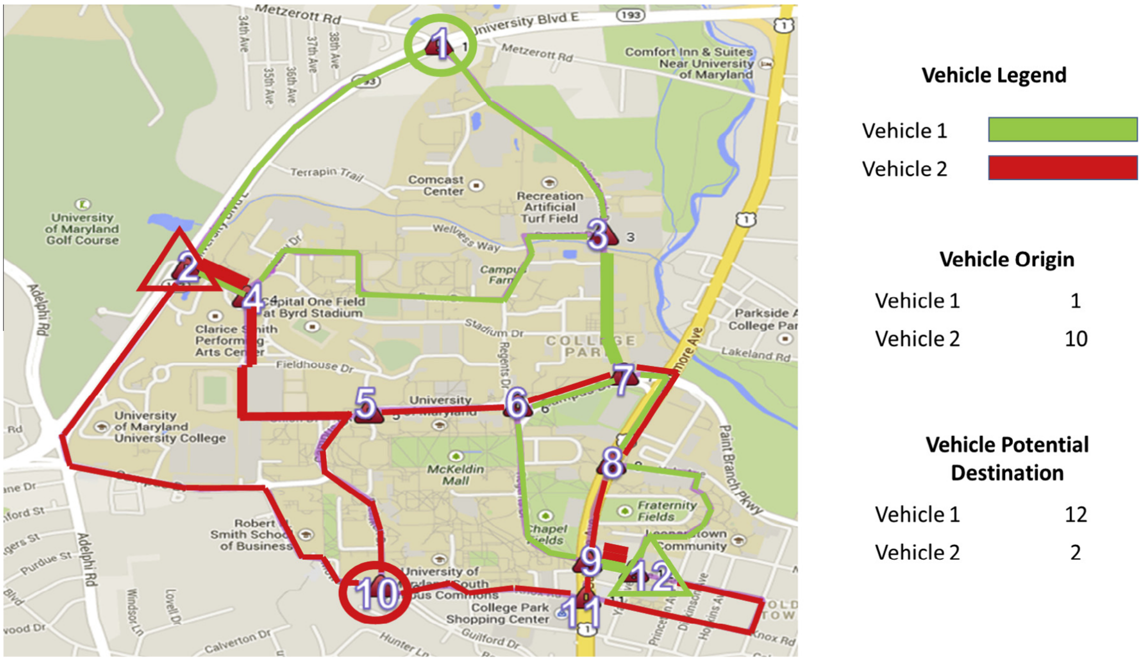

Fig. 6. Best routes per cycle for each vehicle (case 2).

5. Example and results

To illustrate how the model behaves, we assumed that during each shift, $T = 6$ h, 2 vehicles will be patrolling the campus. Even though, the minimum number of times each arc should be traversed during each shift, n, effects the outputs of the model, it does not greatly affect the performance of the model since the modeling is done *per* cycle. n should be inputted based on expert opinions in the field of safety or users of this model. We set the minimum number of times each arc should be covered in 6 h to be $n = 26$ for illustration purposes. As a result the maximum duration for each cycle would be 6 h/26 ≈ 830 s. Using the input data gathered, we ran the model initially without workload balancing. The problem was coded in Xpress that used branch and bound for solving the Mixed Integer Linear Programming (MILP) model. The results for cases 2, 3, and 4 are discussed in the following subsections.

5.1. Results for origin and destination different but predefined (Case 2)

We selected the origin for vehicles 1 and 2 to be nodes 1 and 10. And their possible destinations to be nodes 12 and 2 respectively. This selection was based on spreading the vehicles in the map to allow for better coverage. The best found routes for this case can be seen in Fig. 6.

The best route for vehicle 1 is 1 → 3 → 7 → 8 → 12 → 9 → 12 → 9 → 12 → 9 → 6 → 7 → 3 → 4 → 2 → 1. And the best route for vehicle 2 is 10 → 11 → 9 → 11 → 12 → 9 → 12 → 9 → 8 → 7 → 6 → 5 → 4 → 5 → 10 → 2. As it can be seen the second and third shortest arcs ("9–12", "2–4") have been traversed multiple times using each vehicle while the shortest arc ("9–11") has only been traversed once. The reason behind such behaviors can be seen in the crime weight heat-map (Fig. 5). While traversing the shortest arc consumes the least amount of resource (time), there is no benefit in traversing it more than the minimum required because it has a weight of zero. On the other hand, the next two shortest arcs have weights greater than zero and a relatively small cost at the same time and are worth traversing multiple times. Arc "4–5" is traversed two times by vehicle 2. This is because it has a relatively high weight and it does not have a very high cost compared to other arcs such as "3–4" for example.

Looking at the origin and destinations of each cycle for this case we can see that the destination for vehicle 1 is at its origin, even-though we had allowed it to be node 12. By allowing node 12 to be the destination of this vehicle, we eliminated the need for having conservation of flow at this point. However, if it the solution is better when the vehicle loops per each cycle, the program itself will allow this. The objective function value for this case is $z = 92$.

5.2. Results for origin set but destination free (case 3)

In this case for the purpose of comparison with other case 2, we set the origin of the two vehicles to be the same as case 2. Vehicle 1 and vehicle 2's origins were nodes 1 and 10 respectively. The results for the best paths found are depicted in Fig. 7.

The best path found for vehicle 1 is 1 → 3 → 7 → 8 → 9 → 12 → 9 → 12 → 9 → 6 → 7 → 3 → 4 → 2 → 4 → 2 → 1. The path for vehicle 2 would then be 10 → 2 → 4 → 2 → 4 → 5 → 6 → 5 → 10 → 11 → 9 → 12 → 9 → 12 → 9 → 11 → 12 → 8. As it can be seen, in this case, the vehicles have more available resources to cover arcs "2–4" and "9–12" more. Similar to the solutions for case 2, vehicle 1 will be looping and vehicle 2 will have a different destination than its origin. However, vehicle 2's destination will this time be node 8. This freedom has caused this case's objective function value to be 98.5.

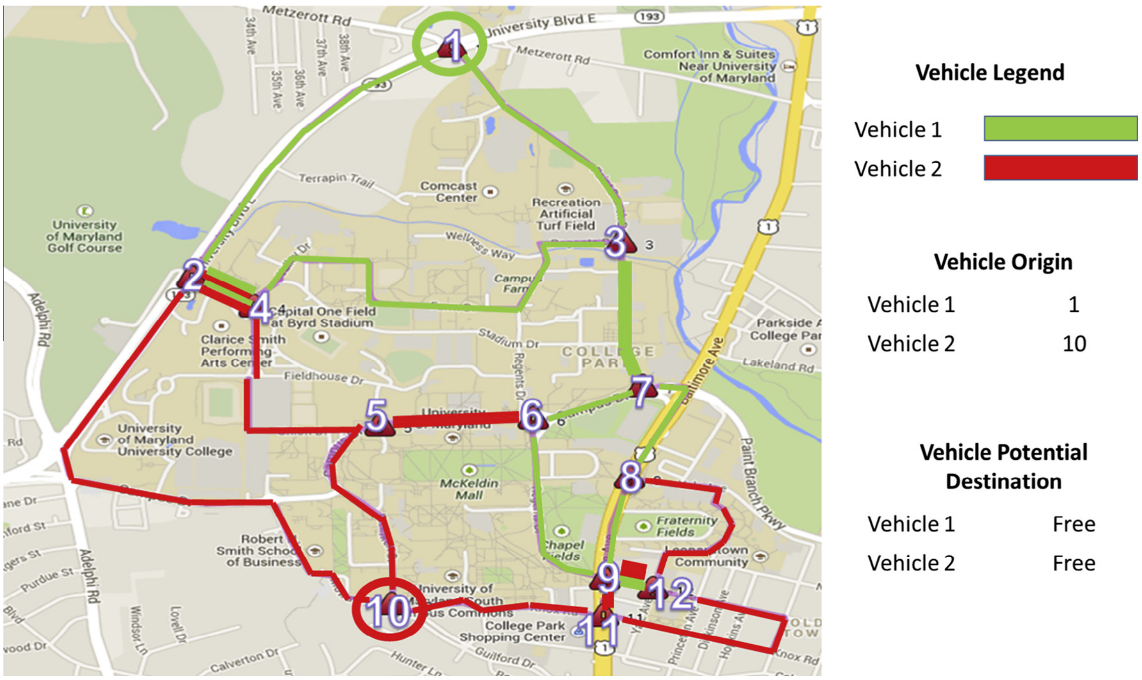

Fig. 7. Best solution found for case 3 (vehicle cycle destination free).

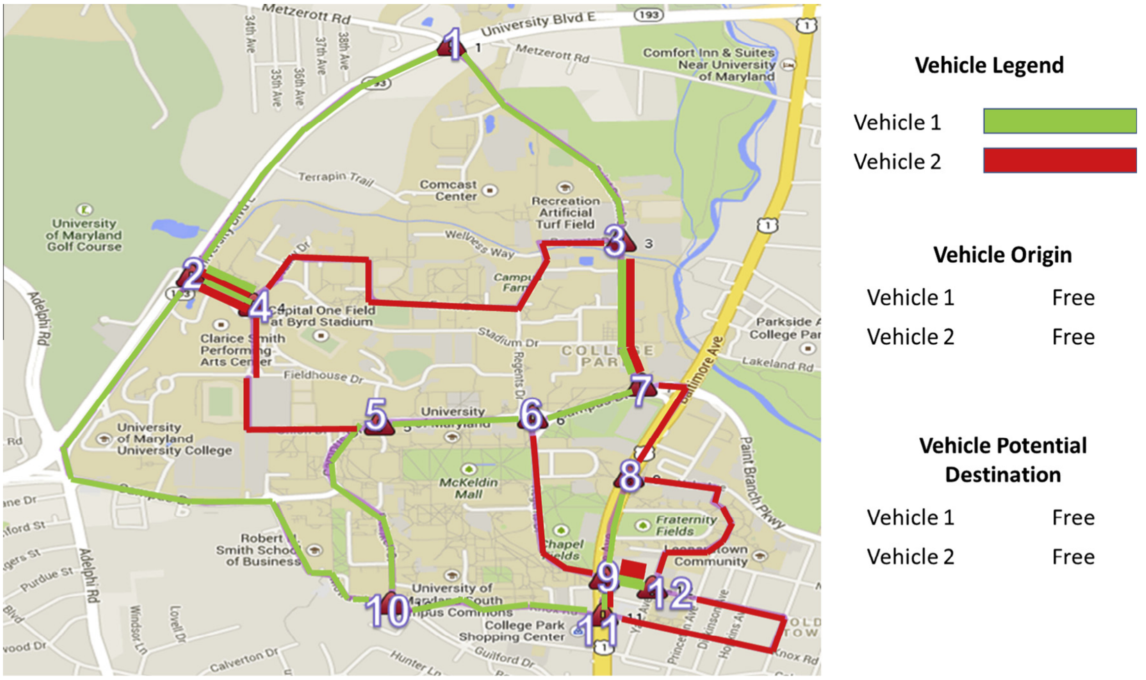

Fig. 8. Results for general case (case 4 that the origin and destination per cycle are free).

Table 3
Summary of results for cases 2–4.

Case	Problem	Origin	Destination	Number of vehicles	Number of integer solutions found	Gap	Objective function value	Time best solution was found (s)	Execution time (s)
2	Origin and destination not necessarily the same but both given as inputs	✗	✗	2	2	0	92	0.7	71
3	Origin given as input; destination set to be free (optimized)	✗	✓	2	5	0	98.5	16531.8	154,194
4	General (Origin and destination set to be free)	✓	✓	2	8	4.95%	102.5	887.5	300,000

5.3. Results for general case (case 4)

In this general case, we do not need to set the origin and/or the destination for the vehicles. The optimization problem itself will find the best locations. The solutions for this comprehensive case are illustrated in Fig. 8.

The cycling route for vehicle 1 would be $8 \to 9 \to 12 \to 9 \to 12 \to 9 \to 11 \to 10 \to 2 \to 4 \to 2 \to 4 \to 2 \to 1 \to 3 \to 7 \to 6 \to 5 \to 10$ and the route for vehicle 2 would be $5 \to 4 \to 2 \to 4 \to 2 \to 4 \to 3 \to 7 \to 8 \to 12 \to 9 \to 11 \to 12 \to 9 \to 12 \to 9 \to 6$. In this case unlike the other cases, the best routes found per cycles are not closed loops and the origin and destination for both vehicles are different. Vehicle 1's origin and destination are nodes 8 and 10. Vehicle 2's origin and destination are nodes 5 and 6. In this case the objective function value is 102.5. Table 3 summarizes the solution results for cases 2, 3 and 4.

The results indicate how the objective value increases for each cycle as we generalize the model and allow the model to select the origin and/or destination for the vehicles itself.

6. Summary and conclusions

In this paper we tackled the problem of routing patrol vehicles and presented a novel mathematical model that can be used to find optimum routes based on different cases. The four cases in this study varied based on the origin and destination nodes for each vehicle during each cycle. Case 1 is the most restrictive case which assumes that the vehicles' routes should be in loops and the depot is an input to the model. The other cases, allow the possibility of having routes which do not form

Please cite this article in press as: Shafahi, A., Haghani, A. Generalized Maximum Benefit Multiple Chinese Postman Problem. Transport. Res. Part C (2015), http://dx.doi.org/10.1016/j.trc.2015.01.017

loops. Case 2 assumes that the origin and possible destination for each vehicle is an input to the model. Case 3 presumes that the origin for the vehicles is an input and the destination can be any node which produces better solutions. The most general case, case 4, has no assumption on the origin and destination of the vehicles and allows the problem itself to find the best locations. The models were based on the k-CPP and arc routing problems with profit. We illustrated the performance of the models through applying them to the network of University of Maryland's campus. The results indicate how the solution improves by allowing the model to pick the optimal location for the destination and/or origin itself.

Looking at the solution times, we can observe that Xpress needs much time to find the best solution and even way more time to prove that the solution is optimal. This illustrates the need for developing algorithms for solving these types of problems. In addition to the need for developing solution algorithms, this research opens some other avenues for future research. Some of them are: performing sensitivity analysis on all input parameters such as travel times on each arc, and adding uncertainties to the problem such as uncertainties in travel times and modeling the problem as a stochastic problem as opposed to a deterministic model.

Acknowledgement

We would like to thank the students of ENCE360 in Spring 2014 for their assistance in gathering the data and preparing the inputs.

References

Ahr, D., Reinelt, G., 2006. A tabu search algorithm for the min-max k-Chinese postman problem. Comput. Oper. Res., 3403–3422
Applegate, D., Cook, W., Dash, S., Rohe, A., 2002. Solution of a min–max vehicle routing problem. INFORMS J. Comput. 14 (2), 132–143.
Archetti, C., Feillet, D., Hertz, A., Speranza, G., 2010. The undirected capacitated arc routing problem with profits. Comput. Oper. Res. 37 (11), 1860–1869.
Archetti, C. et al, 2013. The team orienteering arc routing problem. Transport. Sci. 48 (3), 442–457.
Benavent, E., Corberan, A., Plana, I., Sanchis, J.M., 2009. Min–max k-vehicles windy rural postman problem. Networks 54 (4), 216–226.
Corberan, A., Plana, I., Sanchis, J.M., 2007. A branch & cut algorithm for the windy general routing problem and special cases. Networks 49 (4), 245–257.
Degenhardt, J., 2004. An ant-algorithm for the k-Chinese postmen problem. s.l., Operations Research Conference.
Delgado, C.R., Pacheco, J., 2001. Minmax vehicle routing problems: application to school transport in the province of burgos. In: Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems, vol. 505, pp. 297–317.
Feillet, D., Dejax, P., Gendreau, M., 2005. The profitable arc tour problem: solution with a branch-and-price algorithm. Transport. Sci. 39 (4), 539–552.
Filipiak, K.A., Abdel-Malek, L., Hsieh, H.-N., Meegoda, J.N., 2009. Optimization of municipal solid waste collection system: case study. Pract. Periodical Hazard. Toxic Radioact. Waste Manag. 13 (3), 210–216.
Kramberger, T., Zerovnik, J., 2007. Priority constrained chinese postman problem. Logist. Sust. Transport 1 (1).
Lacomme, P., Prins, C., Ramdane-Cherif, W., 2001. A genetic algorithm for the capacitated arc routing problem and its extensions. Appl. Evol. Comput., 473–483
Malandraki, C., Daskin, M., 1993. The maximum benefit Chinese postman problem and the maximum benefit traveling salesman problem. Eur. J. Oper. Res., 218–234
Mei-Ko, K., 1962. Graphic programming using odd or even points. Chin. Mathematics, 237–277.
Osterhues, A., Mariak, F., 2005. On Variants of the k-Chinese Postman Problem. Operations Research and Wirtschaftsinformatik, University Dortmund, 30.
Pearn, W.L., Wang, K.H., 2003. On the maximum benefit Chinese postman problem. Omega 31, 269–273.
Platz, T.T., Hamers, H., 2011. On Games Arising from Multi-Depot Chinese Postman Problems. Essays in Cooperative Game Theory, vol. 53.
Rao, T.M., Mitra, S., Zollweg, J., 2011. Snow-Plow Route Planning using AI Search. IEEE, Anchorage, AK, pp. 2791–2796.
Shafahi, A., Haghani, A., 2015. Balanced routing of patrolling vehicles focusing on areas with historical crime. In: Transportation Research Board 94th Annual Meeting. CD-ROM, Washington DC, Article No. 15–4387.
Sherman, L.W., Weisburd, D., 1995. General deterrent effects of police patrol in crime "hot spots": a randomized, controlled trial. Justice Quart. 12 (4), 625–648.
Ulusoy, G., 1985. The fleet size and mix problem for capacitated arc routing. Eur. J. Oper. Res. 22 (3), 329–337.
Willemse, E.J., Joubert, J.W., 2012. Applying min–max k postmen problems to the routing of security gaurds. J. Oper. Res. Soc. 63 (2), 245–260.
Wolfer Calvo, R., Cordone, R., 2003. A heuristic approach to the overnight security service problem. Comput. Oper. Res. 30 (9), 1269–1287.
Zhuang, P., et al., 2007. Wireless sensor network aided search and rescue in trials. In: Suzhou, China, Proceedings of 2nd International Conference on Scalable Information Systems. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).