

ABSTRACT

Title of Document: Optimizing patrolling routes using Maximum Benefit k-Chinese Postman Problem

Arezoo Samimi Abianeh
MSc thesis, 2015

Directed By: Dr. Ali Haghani,
Department of Civil and Environmental
Engineering

Providing security and safety in urban areas is of paramount importance. The primary objective of this study is to find several routes for police patrolling vehicles in order to maximize the benefit achieved based on the historical data regarding the crime rate of each part of a given area. To this end, we first formulate the problem as a Maximum Benefit k-Chinese Postman Problem and find the optimal solution for small size networks. We also develop a new metaheuristic algorithm to find suboptimal solutions for the networks. A comparison between the results of the mathematical model and the metaheuristic algorithm reveals that the results are in a good agreement in terms of accuracy and the quality. The proposed metaheuristic algorithm is then employed to find solutions for a larger network.

OPTIMIZING PATROLLING ROUTES USING MAXIMUM BENEFIT K-
CHINESE POSTMAN PROBLEM

By

Arezoo Samimi Abianeh

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
[2015]

Advisory Committee:
Professor Ali Haghani
Professor Paul M. Schonfeld
Associate Professor Philip T. Evers

© Copyright by
Arezoo Samimi Abianeh
2015

Dedication

I dedicate this thesis to my loving parents and my wonderful husband for their support and love. I also dedicate it to my sister and brother who never left my side.

Acknowledgements

I would like to express my special appreciation and thanks to my advisor Professor Ali Haghani. He has been a tremendous mentor for me. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. His advice on both research as well as on my career have been priceless.

I owe my deepest thanks to my family. Words cannot express how grateful I am to them for all of the sacrifices that they have made on my behalf. Your prayer for me was what sustained me thus far.

Thanks are due to Professor Paul M. Schonfeld and Dr. Philip T. Evers for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript.

I am very grateful to my friends for ideas, energy, criticism, advice, support and forgiveness. Their love keeps me going.

I am indebted to numerous people for their continuous support.

I could not have completed my research without the support of all these wonderful people!

Lastly, thank you all.

Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Tables.....	v
List of Figures.....	vi
Chapter 1: Introduction.....	1
Chapter 2: Literature Review.....	6
Models.....	6
Solution approaches.....	13
Chapter 3: Formulation.....	21
Chapter 4: Heuristic.....	26
Tabu search.....	26
Methodology.....	28
i. Initial Solutions.....	28
ii. Neighborhood Solutions.....	33
iii. Tabu search and stopping criteria.....	37
Chapter 5: Results.....	38
Chapter 6: Conclusion.....	61
References.....	65

List of Tables

Table 1. Crime types and weights (Shafahi & Haghani, 2015)	41
Table 2. Results of network 1	42
Table 3. Duration of each cycle	43
Table 4. Solution to the network 1	44
Table 5. length of each cycle of the tabu search solution	45
Table 6. Solution to the network 1	45
Table 7. Length of each cycle of the tabu search solution	45
Table 8. Results of network 2	46
Table 9. Duration of each cycle	47
Table 10. Solution to the network 2	47
Table 11. Results of network 3	49
Table 12. The statistical parameters for 50 replications of the algorithm	51
Table 13. Objective function value and running time of Xpress	53
Table 14. Routes for vehicles (nodes in order)	53
Table 15. The statistical parameters for 45 replications of the algorithm	56
Table 16. Solution 1	56
Table 17. Solution 2	57
Table 18. Running time for the best solutions	58

List of Figures

Figure 1. Saving Algorithm method (Lysgaard, 1997).....	14
Figure 2. Ant colony procedure	17
Figure 3. Tabu search procedure.....	27
Figure 4. Generating Initial Solution flowchart.....	30
Figure 5. Initial Solution Improvement flowchart	33
Figure 6. Generating Neighbor Solution flowchart	36
Figure 7. Small network 1 (city of College Park).....	39
Figure 8. Costs and benefits of edges in network 1	40
Figure 9. Routes of vehicles.....	42
Figure 10. Pace of finding optimal solution (Xpress).....	43
Figure 11. Tabu search objective value VS time	44
Figure 12. Cost and benefits of edges of network 2	46
Figure 13. Pace of finding optimal solution (Xpress).....	47
Figure 14. The objective function variation with respect to time.....	48
Figure 15. Cost and benefits of edges of network 3	49
Figure 16. Pace of finding optimal solution (Xpress).....	50
Figure 17. The objective function values with respect to time for tabu search and the optimal solution.	50
Figure 18. The objective function values with time for tabu search and the optimal solution.....	51
Figure 19. The consistency of the result for this problem.	52
Figure 20. City of College Park	54
Figure 21. The overall view of the map to be covered by patrolling vehicles.....	55
Figure 22. The objective function values with respect to time for tabu search	58
Figure 23. The values of the objective function with respect to time for tabu search	59
Figure 24. The consistency of the result for this problem.	59

Chapter 1: Introduction

Police Departments have the responsibility of providing security and safety for local residents in urban areas. Street crimes including assault, burglary, robbery and theft are threats to public safety. These crimes not only cause financial damages, but also they sometimes lead to physical and emotional harms for victims. Patrolling urban regions is one of the ways to reduce crime and provide public safety and security. Patrolling vehicles are a group of police vehicles whose task is to monitor a specific geographic region.

Significant increase in the number of patrolling vehicles will result in modest reduction in crime and considerable reduction in disorder within high crime locations (Sherman & Weisburd, 1995). Therefore, patrolling is one of the most successful plans for increasing safety and decreasing crime rates. To this end, police patrolling programs have several goals including crime prevention, quick response to emergencies, and surveillance of public buildings (Oliveira, et al., 2013).

Allocation of patrolling vehicles to certain routes, based on the crime rates of each part of the routes, is necessary to increase the performance efficiency. Although scientific approaches have been employed to increase the efficiency of the patrolling vehicles performance, the planning of these routes are mostly considered by empirical rules (Shafahi & Haghani, 2015). The patrol routes, however, should be as random as possible to increase the safety of different area (Rosenshine, 1970). In the current study, we address the task of planning of patrolling vehicles routes for quick responses to emergencies and crime prevention.

The routing problems are investigated by several researchers and can be categorized into two broad areas, namely node routing and arc routing problems (Assad & Golden, 1995). In the first category that is proposed by Dantzig and Rampsar (1959), the objective is to provide services for a set of nodes that can be all or some of the nodes in the network. The most important instance of the node routing problem is the Traveling Salesman Problem (TSP). In the TSP, the objective is to find a shortest cycle that goes through all of the nodes of the network once. In the arc routing problems, the goal is to find a route or several routes, which cover all or a set of links in the network. The most popular instance of the arc routing problem is the Chinese Postman Problem (CPP) whose objective is to find a route that visits every edge of the network at least once. If the network has an Eulerian circuit, the solution obtained by the CPP method traverses all edges of the network exactly once. Otherwise, the goal is to find a solution that traverses a minimum number of extra edges to visit all the edges of the network at least once.

Although there are many applications for arc routing problems, previous studies mostly focused on node routing problem (Corberán & Prins, 2010). The applications of arc routing problems are garbage collection, snow plowing, network maintenance, mail delivery, patrolling vehicles, etc. There are also many extensions on the topic of arc routing problems such as Rural Postman Problem (RPP), Windy Chinese Postman Problem (WCPP), Windy Rural Postman Problem (WRPP), Maximum Benefit k -Chinese Postman Problem (MBkCPP), k -Chinese Postman Problem (k -CPP), Capacitated Arc Routing Problem (CARP), Prize Collecting Rural Postman Problem (PCRPP), min-max k -Chinese Postman Problem, min-max k -vehicle, and min

Absolute Deviation k-Chinese Postman Problems. These models are explained briefly in the following paragraphs.

In CPP, the goal is to find a path that traverses all of the arcs while in the RPP, there is a set of required arcs in a network including vertices and directed or undirected arcs that need to be visited by the path provided (Kwan, 1962).

In WCPP, considering a graph with undirected edges, two values of cost are associated to each edge of the graph. The cost of traversing each edge can be determined based on the given data and the direction of the path passing the mentioned edge. In WCPP, as its names indicates, the goal is to find the minimum cost path traversing all edges of an undirected graph where the cost of traversing each edge depends on the direction of the movement. Similarly, in the WRPP, the goal is to find a path traversing a subset of the edges of a graph with the same condition for WCPP. Another objective in the WRPP, however, is to find a route that traverses all of the edges in the network at least once (Minieka, 1979).

In PCRPP, similar to the Maximum Benefit Chinese Postman Problem, each edge is associated with a profit that is collected when each edge is traversed (Aráoz, et al., 2009).

In the K-Chinese Postman Problem, the goal is to minimize the total cost of traversing all edges of a graph in K cycles. Three variants of the K-Chines Postman Problem including min-max K-Chinese Postman Problem, Min Absolute Deviation K-Chinese Postman Problem, and Maximum Benefit k-Chinese Postman Problem have been proposed by different researchers (Frederickson, 1979) (Degenhardt, 2004). The difference in the time required to traverse the cycles in k-Chinese Postman

Problem is considerable. To overcome this, min-max k-Chinese Postman Problem is proposed to balance the duration of each cycle by minimizing the maximum traversing time.

In the second aforementioned variant of the k-Chinese Postman Problem called Min Absolute Deviation (MAD) k-Chinese Postman Problem, the goal is to minimize the sum of absolute difference of the actual time and preferred time (average time) over k vehicles.

In the Maximum Benefit k-Chinese Postman Problem, the benefit is achieved each time an arc is traversed. There are some constraints here that may limit the number of times each arc can be traversed, or the total time of the cycle.

Most of the studies on k-Chinese Postman Problem focus on min-max k-CPP. These studies fail to consider benefit because their objective is to minimize the maximum cost. The studies considering profit include Profitable Arc Tour Problem, Undirected Capacitated Arc Routing Problems with Profit, and Team Orienteering Arc Routing Problems. Not all edges need to be covered in the network in these problems. In many applications, however, the primary objective is to visit all edges in the network.

Another assumption that has been considered in most of the previous studies is that all vehicles have the same depot. In the current study, all the edges in the network need to be traversed at least once and there are different depots as the starting and ending point for routes of patrolling vehicles.

To the best of the author's knowledge, the studies on the Maximum Benefit k-Chinese Postman Problem focus on formulating the problem. This problem, however, is a Non-deterministic Polynomial-time hard (NP-hard) and mathematical modeling

cannot find the optimal solutions in an acceptable computational time for large networks. We thus develop a heuristic approach to solve the problem for real size network.

Routine routes for the police patrol vehicles may not lead to increasing the safety of the residential area. To overcome this issue, we generate random solutions to make it unpredictable for criminals to track the patrolling path and schedules.

We consider a network of roads that includes streets and intersections. Each intersection is considered a node and each street is an edge connecting two nodes. Each edge in the network is associated with two values. The first value is the cost of traversing that edge, which is the travel time. The second value indicates the benefit achieved by traversing the edge. This benefit is determined by the crime rate of each segment of the street. The aim is to find several routes that maximize the total benefit achieved by traversing arcs of the network by all vehicles. The objective function for Maximum Benefit k -Chinese Postman Problem, which is investigated in the current study, proposed by Shafahi and Haghani (2015).

This thesis is organized as follows: In Chapter 2, we review the literature. In Chapter 3, we describe our methods and formulations, which is followed by results in Chapter 4. We finally summarize our conclusions in Chapter 5.

Chapter 2: Literature Review

Models

Recently, a mathematical model for optimizing patrolling routes was investigated by Shafahi and Haghani (2015). In their study, a graph with 12 nodes and 18 arcs was considered. Each arc in the graph is associated with a cost and a benefit. The cost is the time of traversing each arc and the benefit is determined by the crime rate of the arc. The objectives of their studies are to find several routes, which cover all edges of the network at least once and maximize the total benefit achieved by all vehicles. They considered four cases for origin and destinations of these patrolling vehicles. In the first case, origins and destinations are given. The second case relaxes the assumption that origins and destinations should be the same. The third case gets the origins and finds the best destination corresponding to each given origin. In the fourth case, the optimization model is allowed to select different locations.

Although a patrolling vehicle in their study can pass one beneficiary edge for several times in a row to maximize the objective function, this does not always lead to a practical maximum benefit such as safety of the residential area. To overcome this, repeating an edge in a row by patrolling vehicle is not allowed in the current study. Maximum Benefit k-Chinese Postman Problem is NP-hard. Therefore, a heuristic approach is required to find the best solution for large size networks, which cannot be solved by the mathematical model.

Shafahi and Haghani (2015) proposed two mathematical models constructed for workload balancing in order to minimize the maximum duration of each cycle and

deviation from the average length of the cycles. The model was tested on a small size network containing the main roads of the University of Maryland.

Benavent et al. (2014) introduced the profitable mixed capacitated arc routing problem. The goal of the study was to find a set of vehicles routes to serve a given number of edges by considering both values profit and cost on the arcs. They presented compact flow-based models in which two types of services mandatory and optional were tackled. The developed models were evaluated based on the quality of their bounds and the CPU time. Available instances in the literature have been tested in their model.

Orienteering Arc Routing Problem (OARP) was investigated by Archetti et al. (2014). The goal in this problem is to find a route visiting the customers, which maximize the total profit collected considering a limit on the length of the cycle. The authors described large families of facet-inducing inequalities for the OARP and presented a branch-and-cut algorithm to find a solution. In OARP, in addition to the regular customers that need to be serviced in arc routing problem, a set of potential customers are also available on arcs.

Corberán et al. (2013) proposed an IP formulation for the undirected Maximum Benefit Chinese Postman Problem. They developed a branch and cut algorithm to solve the problem. In their model, they considered n different benefits based on the assumptions that benefit of each edge decreases as the number of traversals of the edge increases. However, it is not necessary to cover all the edges.

Oliveira et al. (2013) proposed a model to construct routes that efficiently patrol a geographical region. They modeled the problem as an integer-programming problem

whose objective is to minimize the total length of all routes. They considered several assumptions for making routes and a set of given locations must be visited by those corresponding routes. Another set of locations, however, may exist in the model that may not belong to any routes. A third set of locations may also exist here that must be covered because they are not visited but they are close enough to at least one visited location. In this case, the number of routes must be equal to the number of available vehicles. The starting points and ending points of routes are the same. Moreover, for providing balance among routes, the number of visited locations must be approximately equal to each other. This model constitutes an NP hard integer-programming problem. Therefore, they developed a heuristic method to find suboptimal solution, which needs less time to solve the problem.

Willems and Joubert (2012) developed a Tabu search algorithm to generate multiple patrol routes for estates security guards. The objective of their study is to minimize the total travelling distance subject to a maximum route length for postman or to minimize the length of the longest route. This problem is known as the min-max k-Chinese Postman Problem. Their algorithm can be applied to both min-max k-Chinese Postman Problem (k-CPP) and min-max k-Rural Postman Problem (k-RPP). The algorithm proposed in this paper namely Tabu-Guard is implemented in three phases. In the first phase, a constructive heuristic, which is called Generate-Random-Initial-Solutions, is used to find different initial solutions. In the second phase, an improvement procedure namely Improve-Solutions is used in order to improve initial solutions. In the last one, another algorithm, which is known as Tabu-Search, is used to improve solutions further.

Corberán et al. (2011) introduced windy clustered prize-collecting arc-routing problem. In this problem, each arc is associated with a profit, which can be collected once. A mathematical formulation and the polyhedron associated with its feasible solution were studied.

Palma (2011) developed a Tabu search algorithm to solve the Prize Collecting Rural Postman Problem. The proposed algorithm includes two phases. In the first phase, two sub-algorithms are used to generate two feasible solutions. The first sub-algorithm is based on two other algorithms, Maximum Benefit Chinese Postman Problem by Pearn and Wang (2003) and the latter is the approximation algorithm for RPP by Frederickson (1979). The second phase uses Tabu search method to improve initial solutions.

Zachariadis and Kiranoudis (2011) studied the undirected capacitated arc routing problem with profit (UCARPP). The UCARPP has been modified in this study by using a hierarchical objective function that first maximizes profit and then minimizes costs. A local search approach is proposed and two neighborhood solutions are considered. The overall search is coordinated by the use of the promises concept.

Archetti et al. (2010) investigated the capacitated team orienteering (CTOP) and profitable tour problem (CPTP). They proposed exact and heuristic procedures to solve CTOP and CPTP. In CTOP a complete undirected graph, a node as the depot, other nodes as the potential customers, a non-negative demand and a non-negative profit are considered for each customers. The profit of each customer is available only once. The first objective is to find m identical vehicles of capacity Q assuming that each vehicle starts and ends its route at the depot. The second objective of this

problem is to maximize the profit collected by limiting the duration of each tour and the total demand that can be collected by each vehicle. In the CPTP, the objective is to maximize the difference between the total collected profit and the cost of the total distance travelled while satisfying the capacity constraint.

Aráoz et al. (2009) presented the Clustered Prize-Collecting Arc Routing Problem (CPCARP) in which there exists a cluster of arcs. There is only two options available for edges of a cluster in which all or none of them need to be serviced. In the prize-collecting arc routing problem, the profit is achieved only the first time an edge is traversed. The formulation of the problem and an exact enumeration algorithm are proposed in their study to solve the problem.

Brandão et al. (2008) proposed a heuristic algorithm based on tabu Search for solving the Capacitated Arc Routing Problem. The objective is to minimize the sum of total cost and a penalty cost, which is the sum of demands on the edges serviced in each route minus the capacity of each route.

Reis et al. (2006) used an evolutionary multi-agent-based simulation tool, namely GAPatrol, to design effective police patrol route strategies. The main objective of their studies is to find a set of patrol routes that minimizes the number of crimes in a given area. They present two scenarios. The first scenario that is devised as the control scenario in which the departure points are localized in the middle of four quadrants of the area. In the second scenario, however, the departure point of criminals start out from a unique source that forces them to initially roam out around the area. As the result, more dispersed distribution of the hotspots are obtained.

Ahr and Reinelt (2006) proposed a tabu search algorithm for the min-max k -Chinese Postman Problem. The goal is to find k tours that minimize the length of the longest tour in which each edge is traversed by at least one tour. Their focus for solving the problem is on investigating a tradeoff between running time and quality of the solutions. They developed three improvement procedures and analyzed three neighborhood algorithms. In the current study, it is found that tabu search outperforms all known heuristics and improvement procedures.

Aráoz et al. (2006) introduced Prize-Collecting Rural Postman Problem under the name of Privatized Rural Postman Problem. They considered two values benefit and cost for each edge of the network. Each time an edge is traversed a benefit b is achieved and a cost c is paid. The benefit is available on the edges only once. The objective is to find a cycle passing through the depot, which maximizes the benefit minus cost. Various systems have been provided to model the problem.

Osterhues et al. (2005) presented Minimum Absolute Deviation (MAD) k -Chinese Postman Problem (k -CPP), Minimum Square Deviation (MSD) k -Chinese Postman Problem (k -CPP) and the Minimum Overtime k -CPP. Comparing the results of MAD and MSD k -CPP, the objective functions are the same for the same instance. The MSD, however, results in a more balanced solution. The proposed algorithm only uses a heuristic to find a solution of the RPP; it therefore does not always find optimal solution.

Feillet et al. (2005) proposed a branch and price algorithm for solving the Profitable Arc Tour Problem. Their goal was to find a set of cycles with the objective of maximizing the sum of profit minus travel cost. The solution is limited by some

constraints including the maximum number that profit is available on arcs and the maximal length of each tour. Branch and price algorithm is a particular case of the branch and bound algorithm. In the search tree, an upper bound is computed at each node by finding a solution for the linear relaxation of the original problem. Column generation method has been used for solving the program.

Pearn and Wang (2003) proposed a solution procedure to solve the Maximum Benefit Chinese Postman Problem approximately. The algorithm applies the minimal spanning tree and the minimal cost-matching algorithm. A service cost, deadhead cost, and a set of benefits are assigned to each edge. The objective of the MBCPP is to find a tour traversing selected edges, which maximizes total net benefit. They investigated the MBCPP on an undirected network. The algorithm is as follows:

1. Network expansion: replace each edge with the new edge with net cost.
2. Minimal spanning tree: if connected proceed, else using minimum spanning tree find new edges which connect the network.
3. Minimal cost matching: determine the set of odd degree nodes and then construct a matching network. Find the minimal cost matching solution.
4. Benefit maximization: find cycles with negative net benefit. Remove them if there is any and it does not make the tour disconnected.

Calvo and Cordone (2003) have introduced overnight security service problem. A single objective mixed integer-programming model was developed to find the solution. Since this problem is NP-hard, the exact approaches are not practical for real life instances. Therefore, they solved the problem with a heuristic decomposition approach. The algorithm combines two sub-problems including capacitated clustering

problem and multiple traveling salesman problem with time windows. The objective of this problem is to minimize the number of guards subject to partitioning constraints, capacity constraints, radius constraints and, time windows. The partitioning constraints guarantee that one and only one guard should service each request. The capacity constraints demonstrate that each guard can be assigned a limited number of requests. Radius constraints indicate that the response to alert signals must be prompt and the last one, which is time windows, has been provided to assure that the quality of service for routine nodes must be better than a specified level.

The other works that have been done on arc routing problem are as follows: Grötschel and Win (1992) developed a cutting plane algorithm to solve the windy postman problem. Corberán et al. (2006) present a new family of facet-inducing inequalities (Zigzag inequalities) for the Windy General Routing Problem. Corberán et al. (2008) study the Windy General Routing Polyhedron (WGRP) in which they present the general properties and large families of facet-inducing inequalities.

As the literature shows, for the topic of k-Chinese Postman Problem, most of the studies are mainly focused on finding balanced routes. However, the case of maximizing benefit has many practical applications in real world and there is a great potential for further investigations in this area of study.

Solution approaches

The main approach for solving routing problems are heuristic methods because these combinatorial optimization problems are Non-deterministic Polynomial-time hard and cannot achieve optimal solutions in an acceptable computational time for the

large networks. Exact approaches such as branch-and-bound and branch-and-cut are proposed to explore every possible solutions until they reach the best one. Heuristics methods conduct a relatively limited search on the search space to produce good solution. Often their results are acceptable based on the quality of the solution and the required running time.

Savings algorithm, Matching Based Saving algorithm, and Multi-Route Improvement Heuristics are among the famous heuristics which were developed to solve VRPs. In 1964, Clarke and Write proposed a solution algorithm, namely Savings algorithm, to solve Vehicle Routing Problems (VRP) (Lysgaard, 1997). The problem was defined on a network with a set of customers and given demands. Each vehicle with certain capacity has the task of delivering some goods to a number of customers. Every vehicle starts its route from the specified depot and return to that depot again. The basic concept of saving algorithm is shown in Figure 1.

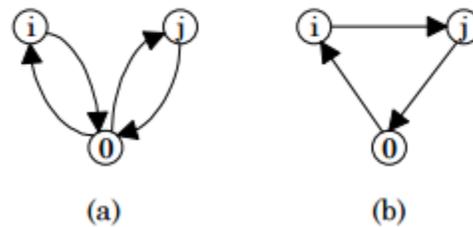


Figure 1. Saving Algorithm method (Lysgaard, 1997)

In Figure 1.a, customers i and j are visited by a vehicle separately but in Figure 1.b customers i and j are visited by one route. In this case, the total cost will be decreased by the value of (cost of traversing from i to depot + cost of traversing from j to depot – cost of traversing from i to j). There are two types for Savings algorithm Sequential Savings algorithm and Parallel Savings algorithm. In the first one, routes will be built

once at a time. In the second type of Savings algorithm, however, several routes would be made at each time. Matching based savings algorithm is a modification of the standard Savings algorithm.

Another heuristic used for solving VRPs is Multi Route Improvement algorithm. This algorithm tries to improve feasible solutions by improvement procedures including exchanging the sequence of edges or vertices within or between vehicles routes.

In several heuristic methods, the problems are decomposed into two components. Two types of these heuristics are cluster-First Route-Second algorithm, and Route-First, Cluster-Second algorithm. As their names suggest, the difference between these two algorithms is that, in the first one, a clustering will be done first and then a route will be assigned to each vehicle based on clusters. However, the latter provides a route of all vertices, which should be traversed, and then a clustering will be performed to find a route for each vehicle.

The next methods of finding solutions for VRPs are metaheuristic methods. Metaheuristics are solution methods, which guide a subordinate heuristic method to improve a set of candidate solutions. In comparison with exact methods for solving optimization problems, metaheuristic methods do not guarantee to find optimal solution. Unlike Heuristic methods, metaheuristics may accept temporary deterioration of the solution, which may prevent being trapped in local optimal solutions. Well-known metaheuristic methods are Ant Colony, Simulated Annealing (SA), tabu Search, and Genetic Algorithm (GA).

The Ant Colony algorithm was proposed by Marco Dorigo in 1992. This algorithm, as its name demonstrates, is based on the behavior of the ant looking for a path

between their colony and the source of food. Ants start their path for searching food randomly. Upon finding food, they return to their colony by laying down pheromone trails. If other ants find this path, they will use it for finding food instead of wandering randomly. Other ants drop down the pheromones in the way of returning to the colony if they found food at destination. Because these pheromones will be evaporated over time, laying down pheromones in the way of returning to the colony by ants will make pheromones trail stronger. As the length of the path from the colony to source of food is shorter, the time needed to traverse it and return to the colony is less. Therefore, there will be a stronger pheromones trail for other ants to traverse. When ants face a depleted food source, they will not return from the trail and lay down pheromones. Therefore, it will be evaporated as time goes on. Figure 2 shows the ant colony algorithm for finding the optimal solution for different combinatorial optimization problem.

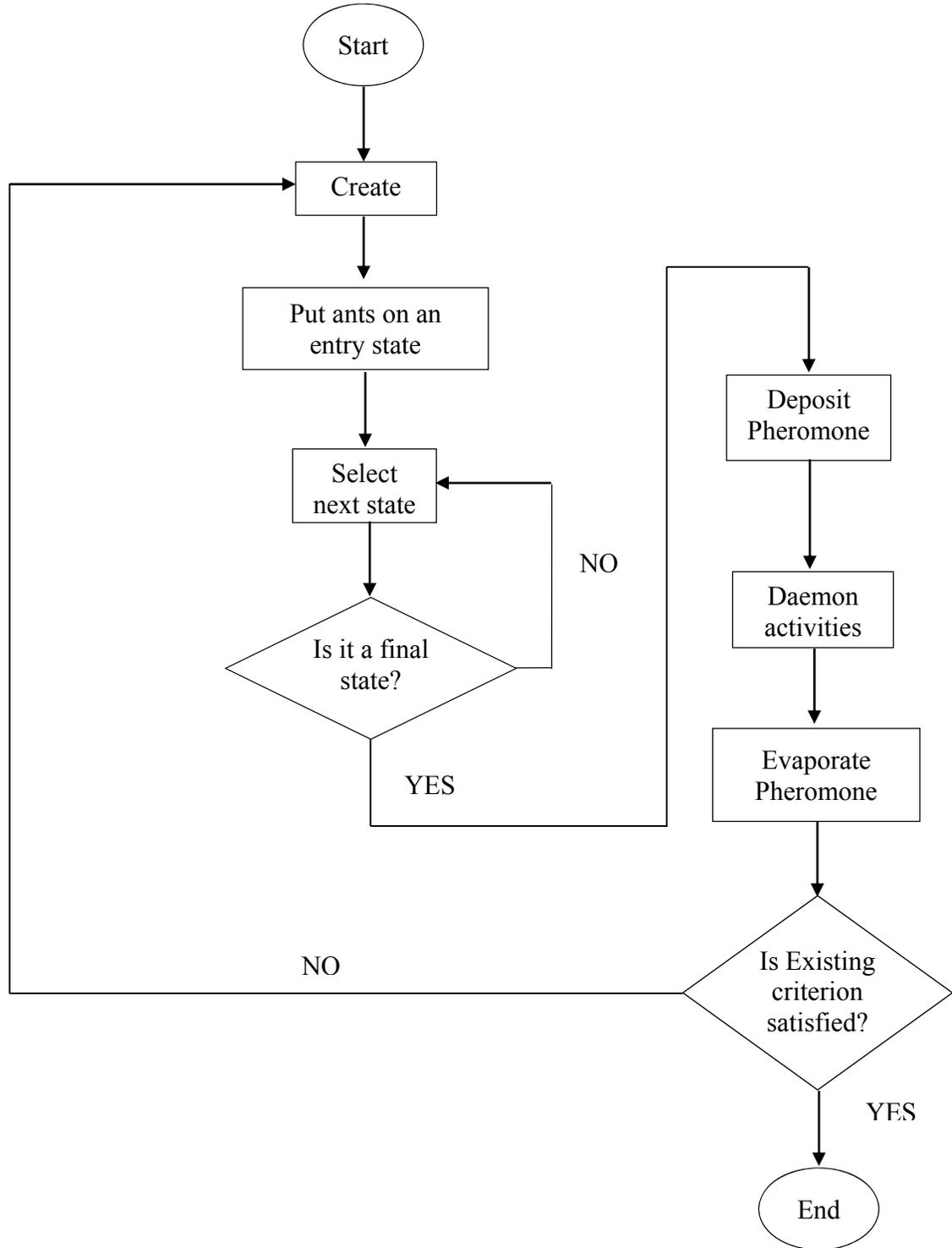


Figure 2. Ant colony procedure

Simulated Annealing algorithm (SA) is a generic probabilistic metaheuristics, which is used to find the global optimum for a problem. SA is based on annealing in metallurgy.

The process involves heating a solid to a high temperature and controlled cooling in order to crystalize in a low energy configuration. The rule of slow cooling of annealing is considered as a slow decrease in the probability of accepting worse solution during the process of SA algorithm. This type of metaheuristic is mostly used for optimization problem with discrete search space.

Yu, et al. (2010) proposed a SA algorithm to solve a location routing problem (LRP). LRP takes into account both facility location problem and vehicle routing problem. The proposed metaheuristic was tested on three networks and the results were compared with the result of the previous studies. It is found that the proposed SA algorithm obtained better results compared to the previous algorithms. Considering a set of customers with given demand and a set of potential sites for the depot the goal was to find the location of the depot and the route of vehicles in order to minimize the total cost of locating depot and traversing routes for servicing customers.

Genetic Algorithm (GA) is one of the famous metaheuristics, which is utilized in various studies. GA mimics the mechanics of natural selection and natural genetics to generate useful solutions to optimization and search problem. In GA, a population of the candidate solution is found initially. Then GA evolves the candidate solutions in order to find better solutions.

Lacomme et al. (2001) presented the first Genetic algorithm for solving capacitated arc routing problem (CARP) (Lacomme, et al., 2001). They used Hybrid GA, which

is a combination of GA and local search methods in order to speed up the process of converging to the Global optimum solution. They applied their algorithm to the extended CARP with mixed graph. The results show that the hybrid GA performs much better compared to the traditional GA.

The last well-known metaheuristic method is tabu search, which is one of the most powerful heuristic especially in arc routing problems. This method is described in more details in Chapter 4.

To the best of our knowledge, most of the above models do not cover all edges in the network, which may be required for some applications such as police patrolling vehicles. Many problems may also need more than one depot for all vehicles in the network, which rarely is considered in the previous studies. Moreover, in the presence of historical data, priority of some arcs may be different. For example in the case of snow plowing, some of the major roads should be cleared more often because of the variation in snow or traffic loads.

In most of the studies, the focus is on formulating the problem. However, since this problem is NP-hard, a heuristic approach is needed to solve the problem for real size networks. The other drawback of the previous studies focusing on maximizing benefit is that the vehicle may traverse an edge several times in a row to increase the value of the objective function. In this case, the maximum number for traversing the edge is set to prevent the vehicle from traversing an edge continuously. Although, this may lead to a higher value for the objective function, the proposed cycles are not practical for increasing network safety.

In the current study, we assume that some of the roads should be patrolled by police vehicles more often because they are more prone to crimes and incidents. However, edges cannot be traversed more than once in a row. The proposed cycles are thus practical for real situations. In most cases, there are different depots in the network. Therefore, the proposed algorithm should be able to consider more than one depot for all vehicles. Moreover, all existing edges in the network must be visited to obtain full coverage by police departments. This study also considers both the benefit and the cost associated with each edge in the network with the objective of maximizing the total benefit.

Chapter 3: Formulation

A mathematical formulation for Maximum Benefit k -Chinese Postman Problem is presented in this chapter. The Maximum Benefit k -CPP is defined on an undirected graph $G = (V, E)$. Where, V is the set of vertices and E is the set of edges of the given graph. The goal in this problem is to find k cycles covering all the edges of the network with the objective of maximizing total benefit. Each edge is associated with a cost and a benefit. The cost of each edge represents the duration of time needed to traverse it. The benefit is calculated based on the crime rate of each edge in the network. To model this problem, several assumptions need to be made.

The first assumption is that an edge cannot be traversed two times in a row. For instance, if vehicle k traverses edge ij (i toward j). It cannot return to node i right after reaching node j . It means edge ij and ji cannot be traversed one after the other. This assumption is a reasonable assumption for considering real situations. It is more reasonable to try not to traverse each edge repeatedly to get more benefit and then not traverse it any more during the cycle.

The second assumption for this problem is that cycles can have different depots. The origin and destination for each vehicle, however, is the same. The formulation provided is as follows.

Variables and Parameters:

x_{ijl}^k : The number of times each edge ij with next destination l is traversed by vehicle k .

y_{ijl}^k : The flow of a commodity from node i to node j with the next destination of l by vehicle k . This variable represents a fictional flow and is continuous. This variable is defined for breaking sub-tours.

h_{ijl}^k : The flow of a fictional commodity from node i to node j with the next destination of l by vehicle k . This variable represents a fictional flow and is continuous. This variable is defined for breaking sub-tours.

d_{ij}^k : $\begin{cases} 1 & \text{if node } j \text{ is visited as the second destination of node } i \\ 0 & \text{otherwise} \end{cases}$

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

b_{ij} : The benefit, which is achieved by traversing the arc with the start node of i and end node of j .

T_{ij} : The duration of traversing from node i to node j

T_{max} : The maximum duration that each vehicle can traverse during each cycle.

M : A large number.

$$\text{Maximize } z = \sum_i \sum_j \sum_l \sum_k b_{ij} x_{ijl}^k \quad (1)$$

$$\sum_l \sum_k x_{ijl}^k + x_{jil}^k \geq 1 \quad \forall i, j \quad (2)$$

$$\sum_i \sum_j \sum_l T_{ij} \times x_{ijl}^k \leq T_{max} \quad \forall k \quad (3)$$

$$\sum_i x_{ijl}^k = \sum_m x_{jlm}^k \quad \forall j, l, k \quad (4)$$

$$\sum_m \sum_n x_{lmn}^k = \sum_i \sum_j x_{ijl}^k \quad \forall l, k \quad (5)$$

$$x_{ijl}^k = 0 \quad \forall i, l \in V, i = l, j \in \{V - origin(j)\}, k \quad (6)$$

$$x_{ijl}^k = 0 \quad \forall i, j, l, k \quad j = l \quad (7)$$

$$\sum_j \sum_l x_{ijl}^k \leq M e_i^k \quad \forall i \in V, k \quad (8)$$

$$\sum_j \sum_j y_{ijl}^k - y_{jil}^k = -e_i^k \quad \forall i \in \{V - origin(k)\}, k \quad (9)$$

$$\sum_l y_{ijl}^k \leq M \sum_l x_{ijl}^k \quad \forall i, j \in V, k \quad (10)$$

$$\sum_j \sum_j h_{ijl}^k - h_{jil}^k = -e_i^k \quad \forall i \in \{V - origin(k)\}, k \quad (11)$$

$$h_{ijl}^k \leq M d_{ij}^k \quad \forall i, j, k \quad (12)$$

$$\sum_l h_{ijl}^k \leq M \sum_l x_{ijl}^k \quad \forall i, j \in V, k \quad (13)$$

$$M d_{il}^k \geq \sum_j x_{ijl}^k \quad \forall i, l, k \quad (14)$$

$$x_{ijl}^k \geq 0 \text{ and integer} \quad (15)$$

$$y_{ijl}^k \geq 0 \quad (16)$$

$$d_{ijl}^k \in \{0,1\} \quad (17)$$

$$e_i^k \in \{0,1\} \quad (18)$$

The objective function (1) maximizes the total benefit of all routes traversed by all k vehicles. Constraints (2) ensure that each arc of the network is traversed at least once no matter which direction of the arc is traversed. Constraints (3) represent the time limitation of each cycle, which is traversed by each vehicle. In other words, the time needed to traverse the cycle cannot exceed the time limit in order to provide several shifts during a day for vehicles.

Constraints (4) and (5) represent the conservation of flow at each node and second destination of each node of the network, respectively. Put differently, these constraints denote that vehicles entering each node should leave the corresponding node. Constraints (6) assure the first assumption in this problem stating that an edge cannot be traversed two times in a row. Constraints (7) ensure that the first and second destination of each node cannot be the same. Constraints (8) to (14) are used for sub tours elimination and continuity. These constraints are provided to assure continuity based on the first and second destination of each node. Constraints (8) to (10) are used for the continuity of the first destination and constraints (11) to (14) are used to assure continuity based on the second destination.

In the formulation proposed in the current study, we use flow commodity constraints to provide the continuity of the solution to the problem. These constraints are mostly used in vehicle routing problems in order to provide the amount of the goods on a vehicle before and after visiting a customer.

The benefits of the edges in the network have been considered in different studies as well as the cost. For example, in the formulation of Profitable Arc Tour Problem, the candidate routes (called collection cycles) are generated by using sub-problems. The

aim is to identify how many times each candidate route should be traversed. This approach demands many preprocessing and calculations for modeling because enumerating all possible combinations for each route in the sub-problem stage is a very time-consuming task.

Undirected Capacitated Arc Routing Problems with Profit restricts the traversing of each arc to be no more than one time. The origin and destinations for all the vehicles must be the same.

Team Orienteering Arc Routing Problems assumes that the origin and destination for all vehicles are also the same. Two sets of customers have been considered; a set of regular customers and a set of potential customers. Each customer is associated with an arc of a directed graph. The profit from a customer can be collected by one vehicle at most.

In the current study, all of the edges need to be traversed at least once for the purpose of increasing safety. Moreover, the origins and destinations for all the vehicles are not required to be the same, which makes it more practical for large networks.

Shafahi and Haghani (2015) also assumed multiple depots for different vehicles in k-CPP. A patrolling vehicle in their model passes one beneficiary edge several times in a row to maximize the objective function. However, this does not always lead to a maximum practical benefit such as safety of the residential area. To overcome this, repeating an edge in a row by patrolling vehicle is not allowed in the current study, to obtain the maximum practical benefit.

Chapter 4: Heuristic

Maximum Benefit k-Chinese Postman Problem (MBkCPP) is a combinatorial optimization problem. MBkCPP is a Non-deterministic Polynomial-time hard (NP-hard) problem. Therefore, we need to find an algorithm, which obtains solution close to the optimal solutions in an appropriate time limit. It is known that tabu search is one of the best heuristic methods for routing problems to find suboptimal solutions.

Randomness plays a vital role in the MBkCPP. Therefore, we need an algorithm to find appropriate random solutions that are close to optimal solution for patrolling vehicles. These solutions decrease the predictability quality of the patrolling vehicles and prevent potential criminals to find the patterns of the patrolling vehicles (Rosenshine, 1970).

The characteristic of providing appropriate solutions for different instances makes this algorithm practical in different applications with the objective of maximizing benefit. Moreover, the randomness feature of this algorithm helps to increase the safety of residential areas in the patrolling vehicles problem.

Tabu search

Tabu search is a meta-heuristic search method, which was created by Fred W.Glover (1986) to solve optimization problems. Tabu search employs local search methods and enhances the performance of this search by using a memory structure called Tabu list, which avoids visiting recently visited solutions. Figure 3 shows the simple demonstration of tabu search heuristic method flowchart.

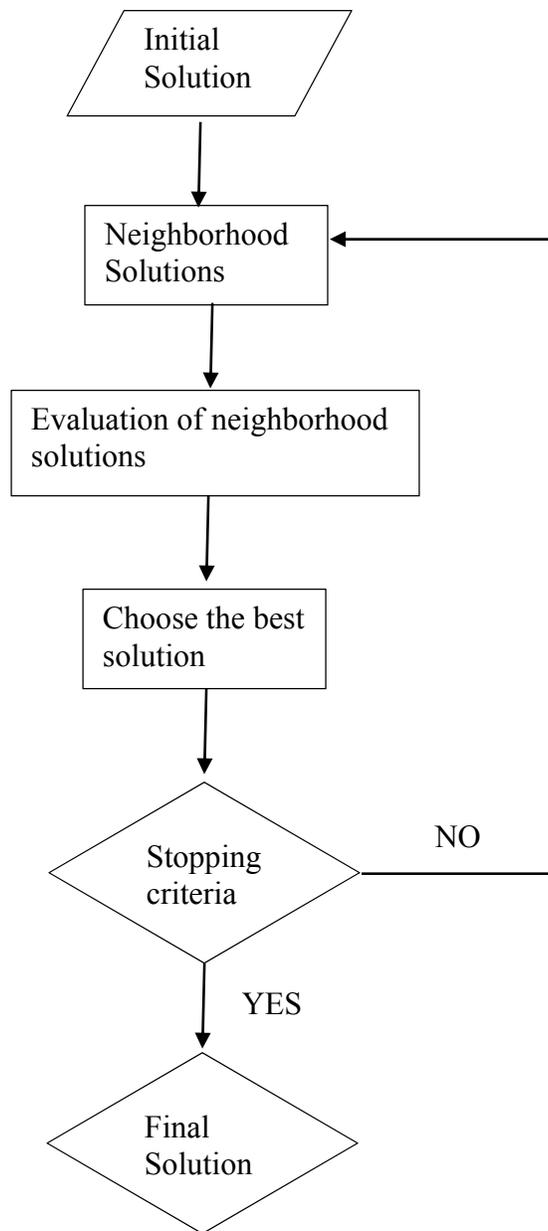


Figure 3. Tabu search procedure

A set of potential solutions is taken and then a set of neighborhood solutions is found by finding the locals of potential solutions. At each iteration, the forbidden moves for the next iteration are stored in the Tabu list and they remain in the Tabu list for a specific number of iteration, namely Tabu-tenure. During each iteration, the tabu search algorithm checks the new moves to see if they are Tabu or not. If the moves

are Tabu, which means that the results may be the same as the previous solutions, this move cannot be performed. There is one exception for the tabu search procedure. This exception occurs when the solution is Tabu and has a better value for the objective function. In this case, the solution is considered for the next move.

Methodology

The terminology, descriptions, and details of the proposed algorithm are explained here for finding optimal solutions in order to maximize profits achieved by patrolling vehicles.

i. Initial Solutions

The first step in the tabu search algorithm is to find a set of initial solution. For this purpose we propose an algorithm to find an initial solution. This algorithm is explained as follows:

In the present network, an edge is chosen randomly. A shortest path then will be found for each of the vehicles by using Dijkstra's algorithm. The minimum total distance, which is the sum of the distance from the depot to the corresponding edge, the length of the edge and the length for returning from the edge to the depot is selected. Afterwards, the path will be added to the chosen vehicle. This algorithm is repeated until all edges of the network are covered. Since edges are randomly selected, repeating this method will find different initial solutions. The constraint, which should be checked during the process of adding edges, is the time limit for cycles. This constraint is checked whenever an edge is going to be added to a cycle. If the time limit cannot be met, another cycle will be chosen for adding the path.

Finally, if the path cannot be added to any cycles the solution will be removed and the algorithm tries to find another initial solution. Using the proposed algorithm an initial solution can be found.

The algorithm is expressed as follows:

Step 1: Choose a random edge from the set including untraversed edges of the network.

Step 2: Connect the chosen edge to the nearest cycle.

Step 3: Remove all the edges that are added including the chosen edge and edges for connecting to the network.

Step 4: Check to see if all edges are covered, if Yes, STOP an initial solution is found. If NO, go to Step 1.

The Pseudocode is as follows:

Algorithm 1: Generate Initial Solution 1:

Input: number of initial solutions n_1 and number of patrolling vehicles k

Output: a set of n_1 initial solutions

For $i : 1$ to n_1 do

$R =$ all the edges of the network

While ($R \neq \emptyset$)

$(e_1, e_2) =$ random edge from the set R

For $j: 1$ to k do

$V =$ set of all nodes which are already in the cycle k

For $l \in V$

$$length_l^j = SP(l, e_1) + length(e_1, e_2) + SP(e_2, l)$$

$$\min = \min (\text{length}_l^j) \text{ on all } j, l$$

Add path associated to \min to the cycle j^{th} if it meets the feasibility condition, in the appropriate place,

Remove all edges of the corresponding path from R

Return (vehicles route)

Figure 4 shows the steps for finding the initial solution.

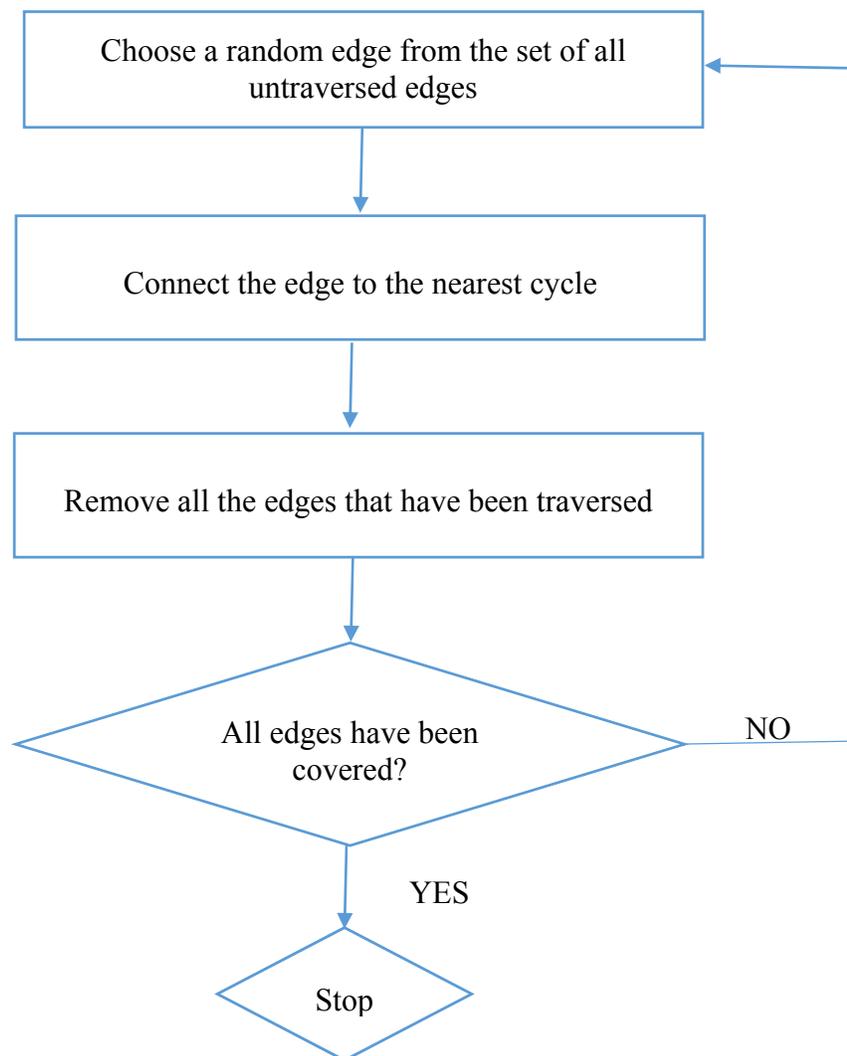


Figure 4. Generating Initial Solution flowchart

The next step after finding the initial solution is to improve it. Two improvement procedures are proposed to find the final initial solution. Since for generating initial solution the goal is to cover all edges, there still may be more time available for vehicles to cover more edges in each cycles. The improvement algorithms try to add edges, which are beneficiary to each cycles. In the first improvement procedure, the SP that is found by Dijkstra's algorithm gives more priority to the edges with benefit more than zero because this results in improving the objective function.

In the second improvement algorithm, there is not any priority to the edges with benefit in finding the SP. The aim is to add any beneficiary edges by SP, which results in finding shorter routing for adding a beneficial edge. The first improvement procedure is as follows:

Step 1: Sort the edges that have benefit higher than zero based on decreasing order of benefit.

Step 2: Choose a cycle from the set of cycles in initial solution.

Step 3: Choose the first element of the sorted edges and check to see if it is possible to add the edge to the cycle. If so, remove the edge from the set.

Step 4: If the set sorted edges is not empty go to Step 3.

Step 5: Remove the cycle from the list of cycles.

Step 6: If the list of cycles is not empty go to step 1. If it is empty STOP, the initial improvement has been finished

Algorithm 2: Initial improvement procedure:

Input: an initial solution and k : number of cycles

Output: an improved initial solution

$R = \text{sorted edges based on decreasing of benefit}$

$h = 0$

While ($h < \text{size of } R$)

$h++$

$(e_1, e_2) = R(h)$

For ($j=1$ to k)

$V = \text{set of all nodes which are already in the cycle } k$

For ($l \in V$)

$$\text{length}_l^j = SP(l, e_1) + \text{length}(e_1, e_2) + SP(e_2, l)$$

$\text{min} = \text{min}(\text{length}_l^j) \text{ on all } j, l$

Add path associated to min to the cycle j^{th} if it meets the feasible condition, in the appropriate place.

Figure 5 shows the initial solution improvement procedure.

Return (vehicles route)

The next step is to find a neighborhood solution by using the initial solution. It should be noted that at each iteration the best solution according to the objective function is found and compared to the best solution, which was previously stored. Finally, the one with the best objective function will be stored as the best solution.

ii. Neighborhood Solutions

The following algorithm (algorithm 3) is developed for finding neighborhood solutions. Each neighborhood solution is generated based on the previous solution. The generated neighborhood solution do not need to be better than the previous solution

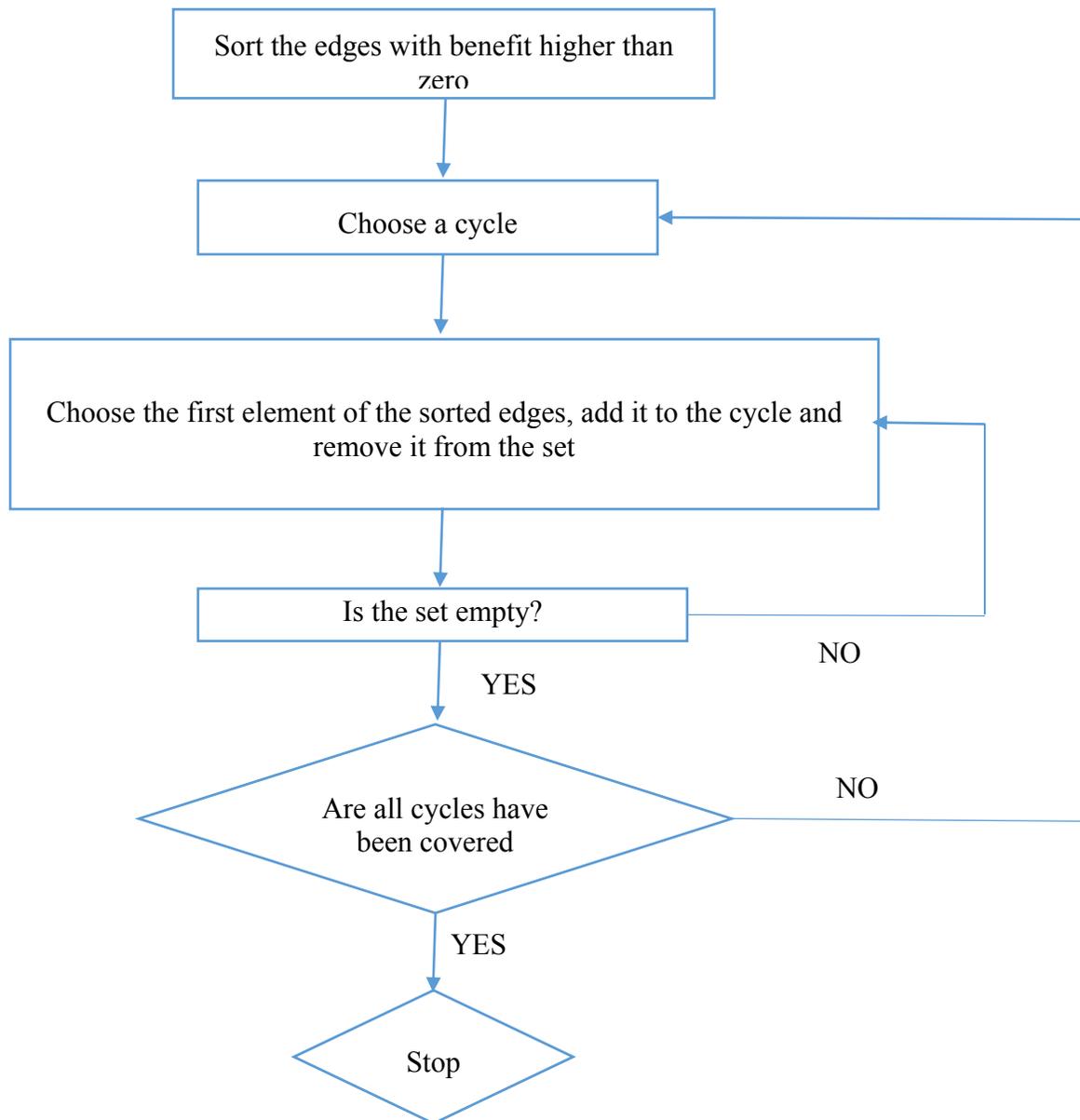


Figure 5. Initial Solution Improvement flowchart

based on the value of the objective function. The goal is to change the current solution including different cycles for finding a neighborhood solution. The first step is to choose a cycle randomly in the current solution. An edge will then be selected randomly from the set of edges of the cycle. The selected edge need to be evaluated at this step. The criteria is to see whether the edge has been covered more than once in the current solution or not. If the criteria is met, then this edge can be removed from the solution. Afterwards, the next edge in the row should be checked for removal. This procedure continues until the criteria for removal has not been met or the end of the cycle has been reached. At this point, the algorithm tries to connect the two ends of the remaining edges in the current solution with an appropriate path.

The shortest path is selected in the current solution by Dijkstra's algorithm with respect to the time constraint. The algorithm evaluates the benefits of all paths. The appropriate path with highest possible benefit is selected by Dijkstra's algorithm for connecting the two ends of the removed cycle in the current solution.

An algorithm for finding neighborhood solutions is proposed below.

Step 1: Choose a cycle randomly

Step 2: Choose an edge in the cycle randomly

Step 3: Add another edge to the previous edge if and only if it is covered at least once in the current solutions.

Step 4: Continues until reaching an unavailable edge or the end of the cycle.

Step 5: Remove all edges that are already added to the cycle

Step 6: Sort the edges by benefit in descending order

Step 7: Choose the first edge from the list

Step 8: Add it to the cycle if it connects the two ends of the removed cycle in the current solution. If not remove the edge from the sorted list of edges and go to step 7

Algorithm 3: Generating neighborhood solutions:

Input: a solution S , k : number of vehicles

Output: a neighborhood solution

$cycle = random(k)$

$edge = random(S^{cycle})$

$j = location\ of\ edge\ in\ S^{cycle}$

if (edge is repeated in other locations)

$available = true$

add edge to R

while ($j < size\ of\ S^{cycle}$ & available)

$j ++$

if (edge is repeated in other locations)

$available = true$

add edge to R

remove edges of R from cycle

profitedges = sorted edges based on decreasing of profit

$h = 0$

While ($h < size\ of\ profitedges$ && notfound)

$(e_1, e_2) = R(h)$

$h++$

Add path including edge (e_1, e_2) to the cycle in the place of removed edges if it meets the feasibility condition.

if (found a path)

notfound = false

Return (S solutions)

Figure 6 shows the neighborhood generation procedure.

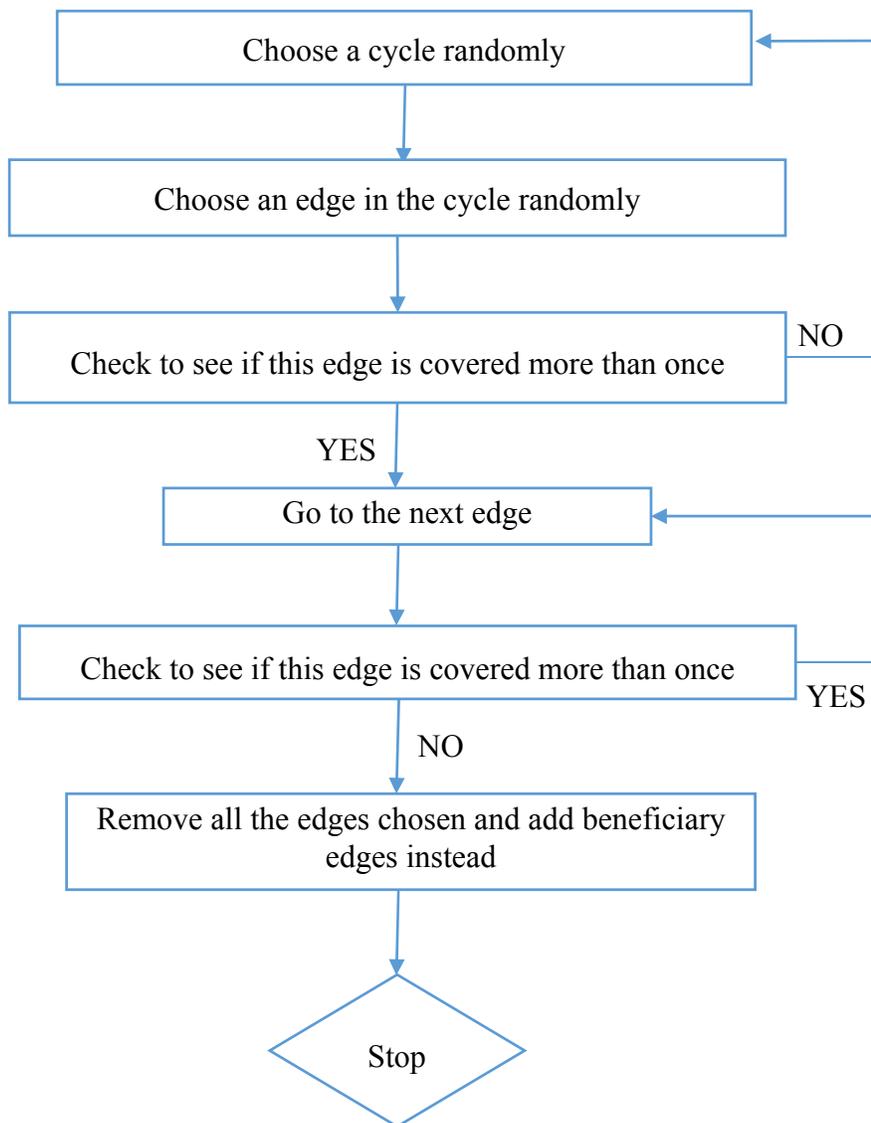


Figure 6. Generating Neighbor Solution flowchart

The next part is allocated to the conditions which are considered for the tabu search and stopping criteria.

iii. Tabu search and stopping criteria

As it was stated in the previous sections, at each iteration, moves become tabu for the specific number of iteration, namely tabu tenure. In other words, these moves in the previous iterations, which is actually the inverse of the moves, cannot be performed. This occurs because these moves will result in the previously visited solutions again. There is one exception for allowing these moves to be performed. The exception is that, this move is performed if and only if the result of such a move would produce a better value for the objective function.

There are many different ways to introduce the stopping criteria. Some studies focus on the improvements and some other consider the number of iterations or time as the stopping criteria. In the current study, time has been considered for the stopping the search for the best solution.

Chapter 5: Results

In this chapter, first, the solutions of the exact method and the metaheuristic approach for three small size networks are demonstrated. The solution to the larger size problem, which are obtained by the developed metaheuristics method, is then presented. To the best of the author's knowledge, data are not available for similar Maximum Benefit k-Chinese Postman Problems in the literature to compare the results. However, in order to validate the metaheuristic method, the results of the exact method, which is based on mathematical modeling, and tabu-search metaheuristic method are compared with each other. The results of the metaheuristic method are found to be in a good agreement with the exact method for the arbitrary small size networks. One of the instances of the problem is the network of the University of Maryland. The results of the metaheuristic algorithm and exact method for this network are provided in the following sections.

Network 1

The network of the University of Maryland, which is considered here for testing the problem, has been also used in one of the recent studies (Shafahi & Haghani, 2015). The overall view of the network is shown in Figure 7. This network has 12 nodes and 18 edges.

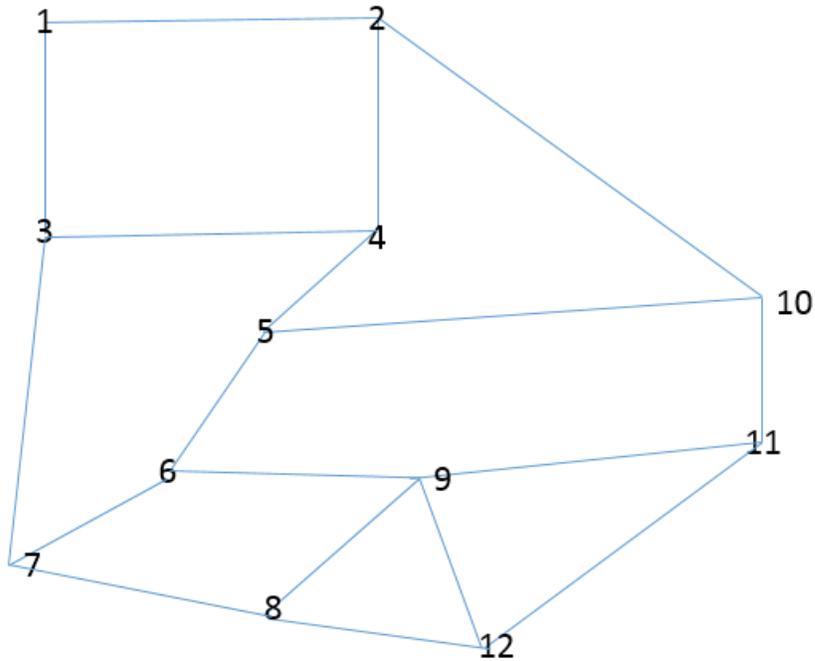


Figure 7. Small network 1 (city of College Park)

The machine used in solving the problem is a desktop computer with a 3.1 GHz CPU and 4.00 GB of RAM. The optimization software is Xpress.

Figure 8 depicts an overall view of the aforementioned networks with two values on each edge. First value is the cost of traversing each edge, which is the time of traversal in seconds. The second number shows the benefit of the edge, which is calculated based on the crime rate of each edge of the network. The depots are shown by circling some of the nodes.

Before proceeding further, how the values of the cost and benefit are achieved is represented here.

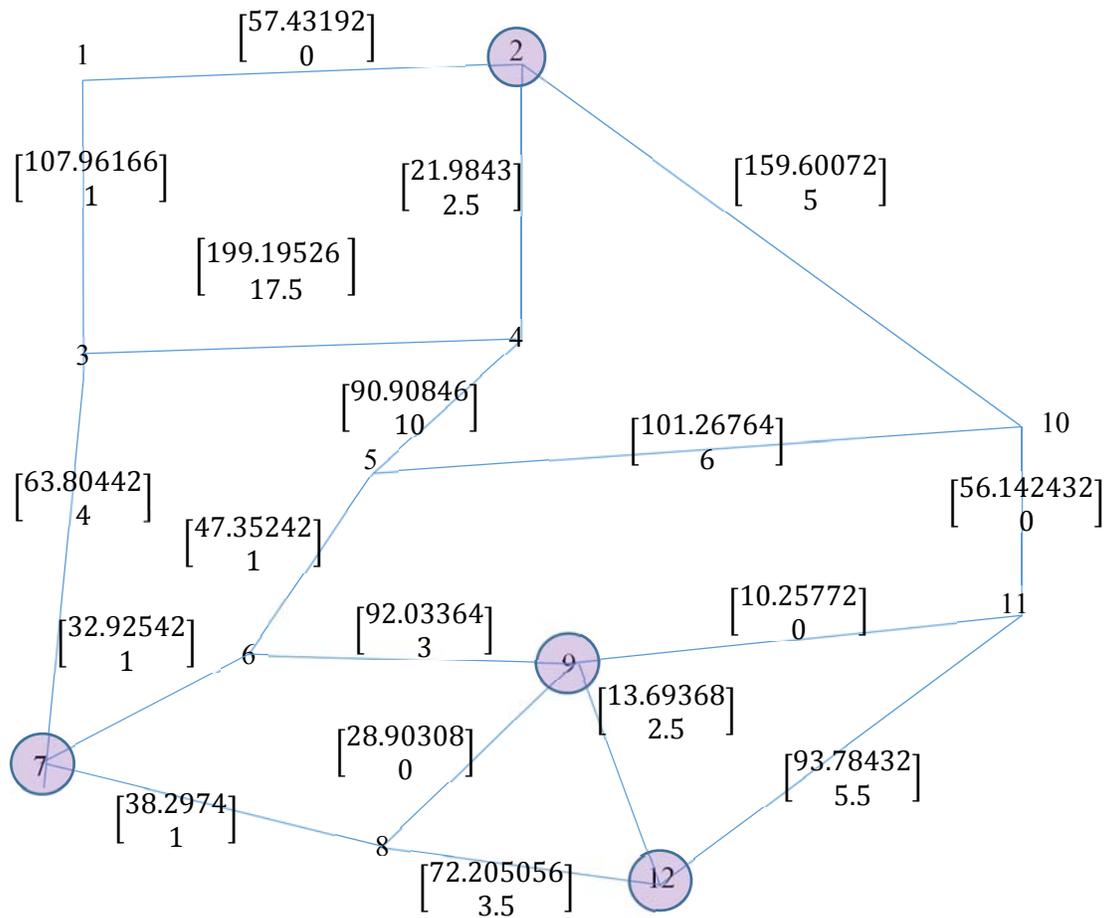


Figure 8. Costs and benefits of edges in network 1

The values of the cost and benefit are calculated based on Table 1, which is given by Shafahi and Haghani (2015). The costs are the travel times required for traversing edges. The travel times have been found based on the division of the length of the edges of the network and the speed limits. However, it would be better for patrolling vehicles to have lower speed than the speed limit in order to increase the efficiency. The benefit is calculated based on the crime rate of each edge of the network, which is a function of the weights assigned to each type of the crime. The types of crimes considered here and the weight assigned to each different crime are given in Table 1.

Table 1. Crime types and weights (Shafahi & Haghani, 2015)

Type of the crime	Weight
Assault	3
Other Sexual Offense	2.5
Robbery	2
Burglary	1.5
theft from vehicle	1.5
Theft	1

The benefit of each edge is calculated based on the values given in Table 1 and the crime rate obtained from [crime reports](#) website.

Figure 9 shows the routes found for four vehicles. The starting point and ending point of each vehicle is different. The nodes that have been considered as the depot are circled in Figure 9.

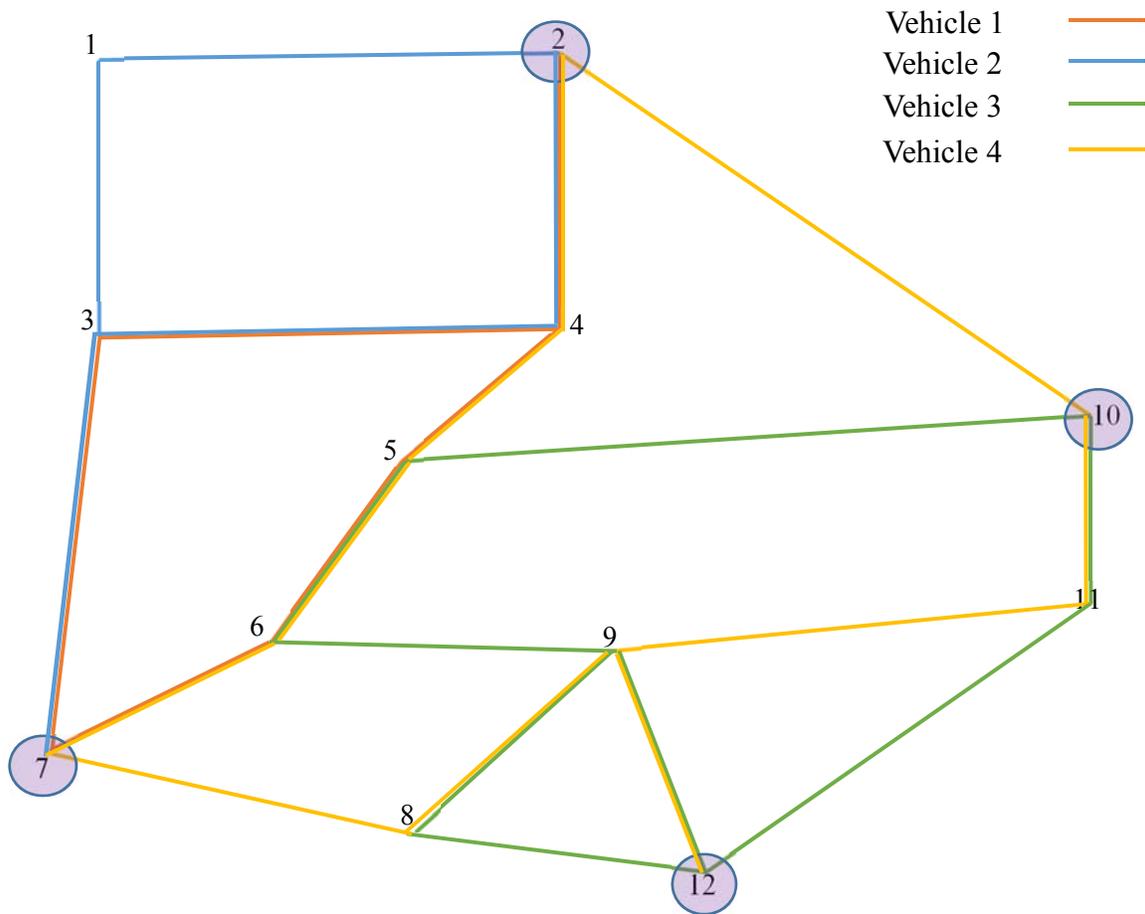


Figure 9. Routes of vehicles

The value of the objective function and the time needed for solving this problem in Xpress is shown in Table 2.

Table 2. Results of network 1

Objective function	Time (sec)
117	7205.5

The duration of each cycle are given in Table 3. It should be noted that the maximum duration of each cycle for the network is 540 seconds.

Table 3. Duration of each cycle

Cycles	1	2	3	4
Duration of cycles (sec)	478.2	450.4	505.4	513.8

Figure 10 shows the variation of the best solution and best bound with time. The red line represents the absolute values of the solutions found and the yellow curve shows the best (lower) bound obtained from the LP relaxations of the remaining open nodes. At the end, the curve reaches the value of the best solution. This means that optimality of this solution has been proven (we may have chosen to stop the search, for example, after a given number of nodes, in which case it may not be possible to prove optimality or even to find the best solution).

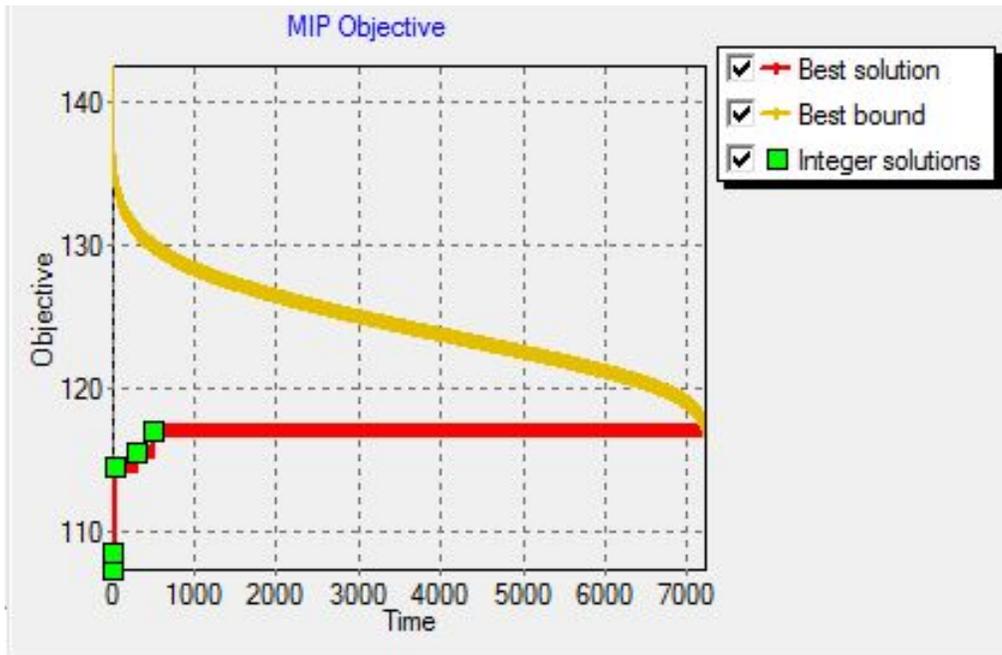


Figure 10. Pace of finding optimal solution (Xpress)

The result of the tabu search for this network is also investigated. Based on the randomness feature of the proposed tabu search, several replications of the algorithm

have been applied to the problem and the results are provided here. The results of the metaheuristic shows that this network has multiple optimal solution, which also provides the goal of having different solutions for preventing creating routine path. The first result of the problem is given in the Table 4.

Table 4. Solution to the network 1

Vehicle 1:	2	→	4	→	5	→	6	→	7	→	3	→	4	→
	2													
Vehicle 2:	7	→	3	→	4	→	2	→	1	→	3	→	7	
Vehicle 3:	10	→	5	→	6	→	7	→	8	→	9	→	12	→
	8	→	9	→	12	→	11	→	10					
Vehicle 4:	12	→	9	→	11	→	10	→	2	→	4	→	5	→
	6	→	9	→	12									

Figure 11 shows the variation of the objective function values based on tabu search with respect to time. A good agreement has been obtained between the objective function and the optimal solution. As it is shown, the tabu search reached the optimal solution in less than 45 seconds.

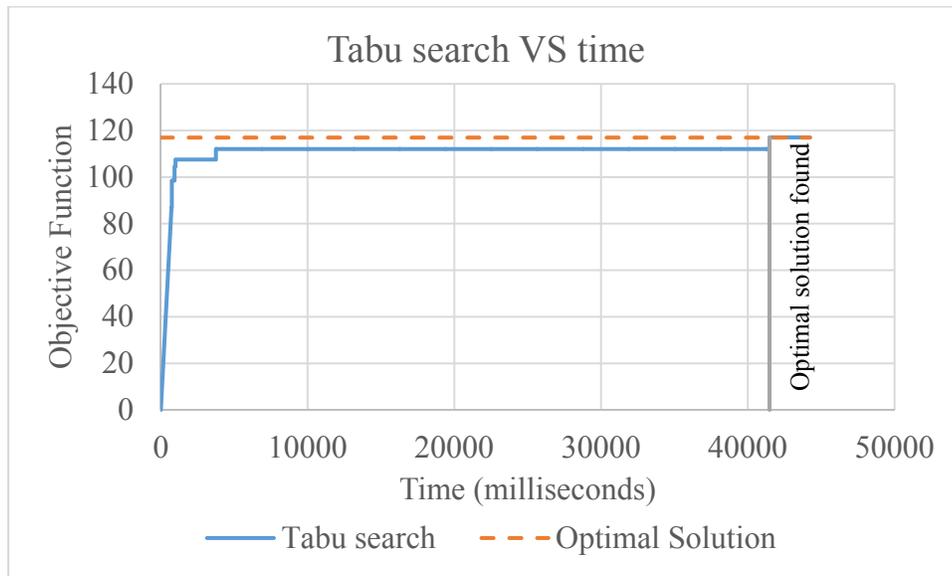


Figure 11. Tabu search objective value VS time

Table 5. length of each cycle of the tabu search solution

Cycles	1	2	3	4
Duration of cycles (sec)	478.2	514.2	539.6	513.8

The second result of the problem is as follows:

Table 6. Solution to the network 1.

Vehicle 1:	2	→	4	→	5	→	6	→	7	→	3	→	4	→
	2													
Vehicle 2:	7	→	3	→	4	→	2	→	1	→	3	→	7	
Vehicle 3:	10	→	5	→	6	→	9	→	12	→	8	→	9	→
	12	→	11	→	10									
Vehicle 4:	12	→	9	→	11	→	10	→	2	→	4	→	5	→
	6	→	7	→	8	→	9	→	12					

Table 7. Length of each cycle of the tabu search solution

Cycles	1	2	3	4
Duration of cycles (sec)	478.2	514.2	519.1	513.8

Based on the value of the objective function, this solution is also optimal. The algorithm has found different solutions for the problem with optimal objective functions.

The results of the metaheuristic presented show that the problem has multiple optimal solutions and the algorithm found different optimal solutions by using multiple replications of the model, which is helpful for having random different solutions for patrolling vehicles.

Network 2

The second network that has been considered as a proof for showing two important characteristics of the metaheuristic provided including quality of the solution and CPU time is shown in the Figure 12. This network, which is not a real network, has

13 nodes and 20 edges. For constructing this network, the real network of the city of College Park is modified by adding and removing some edges arbitrarily.

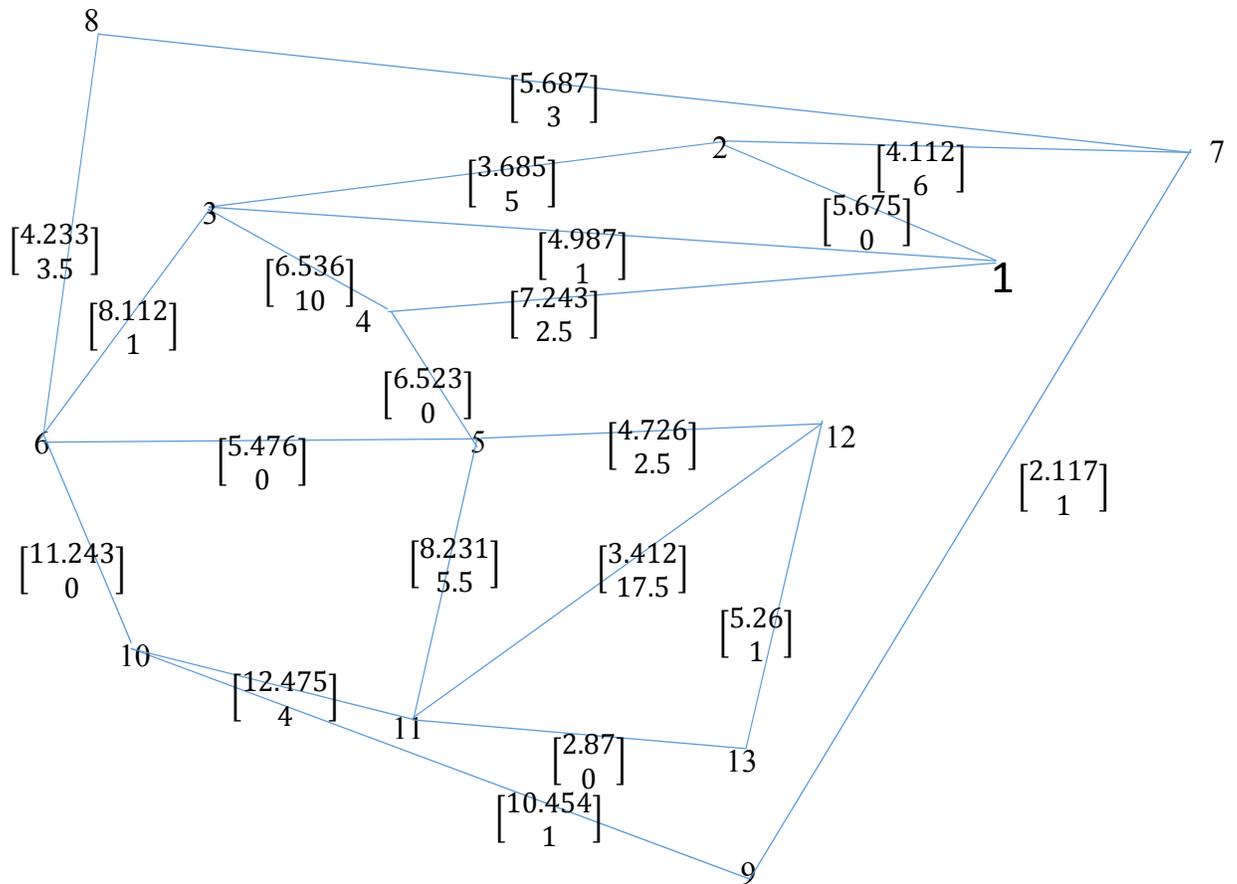


Figure 12. Cost and benefits of edges of network 2

Table 8 shows the value of the objective function and time needed to solve the problem by Xpress. Duration of each cycle is given in Table 9. It should be noted that the maximum length of each cycle for the network is 50 minutes.

Table 8. Results of network 2

Objective function	Time (sec)
150	8186.9

Table 9. Duration of each cycle

Cycles	1	2	3	4
Duration of cycles (sec)	48.968	48.58	48.181	45.462

Figure 13 shows the pace of finding solutions.

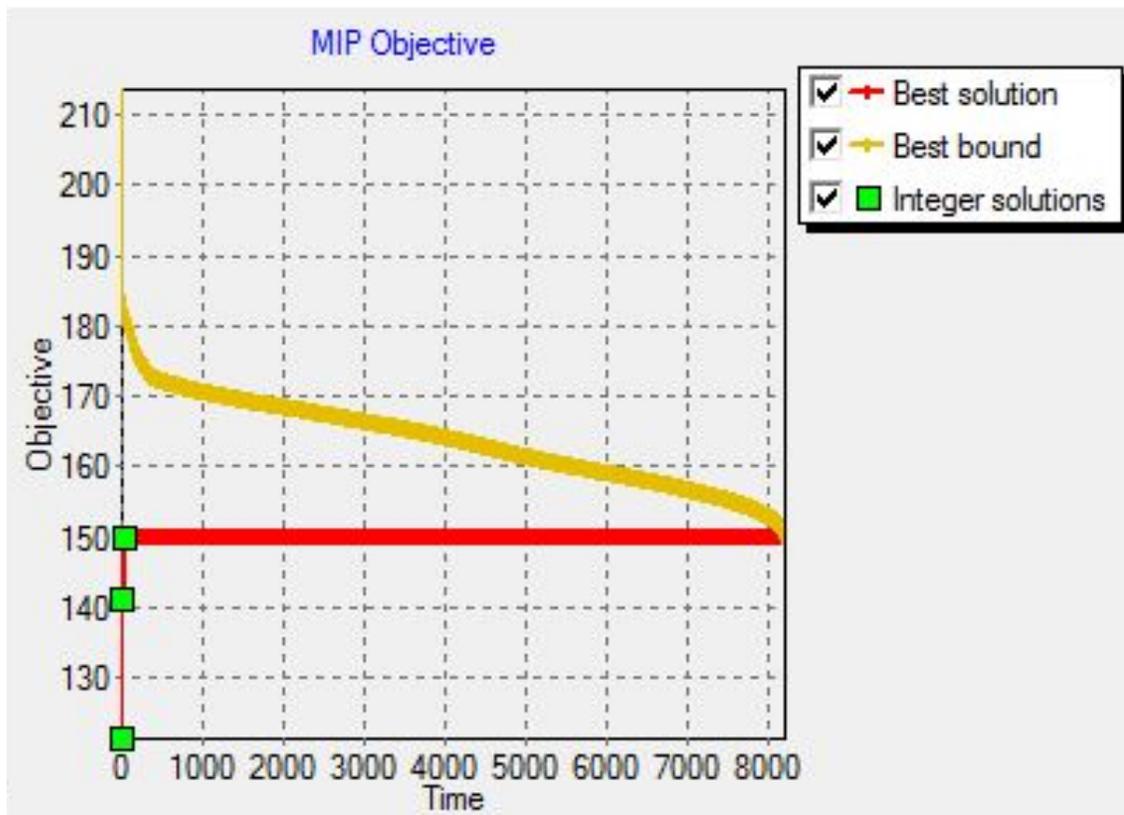


Figure 13. Pace of finding optimal solution (Xpress)

The results from the Xpress are shown in table 10.

Table 10. Solution to the network 2

Vehicle 1:	2	→	3	→	4	→	1	→	2	→	7	→	8	→
	6	→	3	→	2									
Vehicle 2:	7	→	8	→	6	→	5	→	12	→	11	→	10	→
	9	→	7											
Vehicle 3:	11	→	5	→	4	→	1	→	3	→	4	→	5	→
	12	→	11											

Vehicle 4:	13	→	12	→	11	→	10	→	6	→	5	→	12	→
	11	→	13											

Tabu search results in the optimal value of the objective function for this network too.

Figure 14 shows the pace of finding solution for one replication of the algorithm.

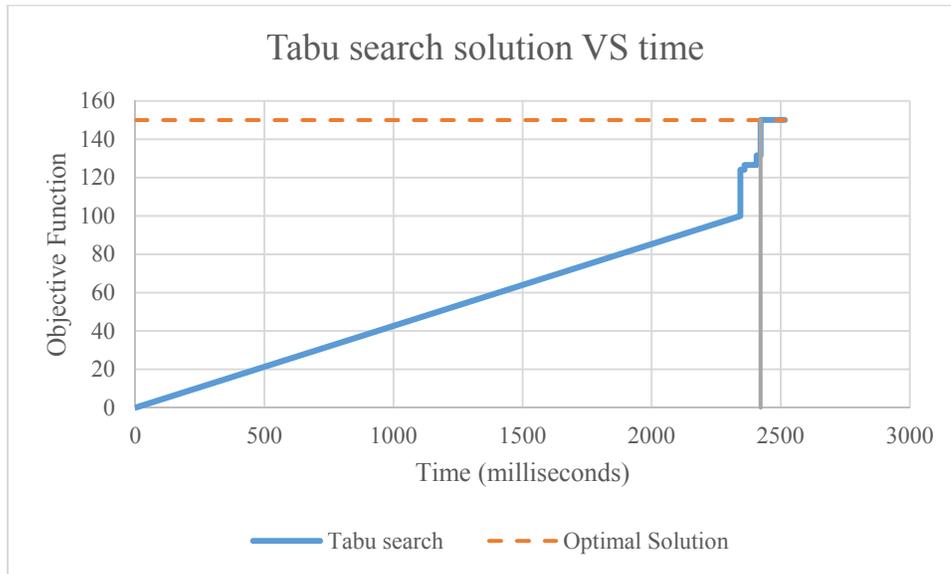


Figure 14. The objective function variation with respect to time.

In almost all cases, the algorithm reaches the optimal solution in less than 10 seconds.

Network 3

The third network that has been considered for testing the algorithm is constructed based on the real network of the city of College Park. The network is modified by adding and removing some edges arbitrarily. This network has 14 nodes and 21 edges. The network with the corresponding values of the benefit and costs (in minutes) is shown in Figure 15. The CPU time required for obtaining the optimal solutions by Xpress is given in Table 11. The maximum length of each cycle is 45

minutes. The results demonstrate that the CPU time for solving this network is 34 times the CPU time of the first one. This occurs because Xpress requires longer time to find the optimal solution for a larger feasible region. Figure 16 shows the pace of converging to the optimal solution by Xpress.

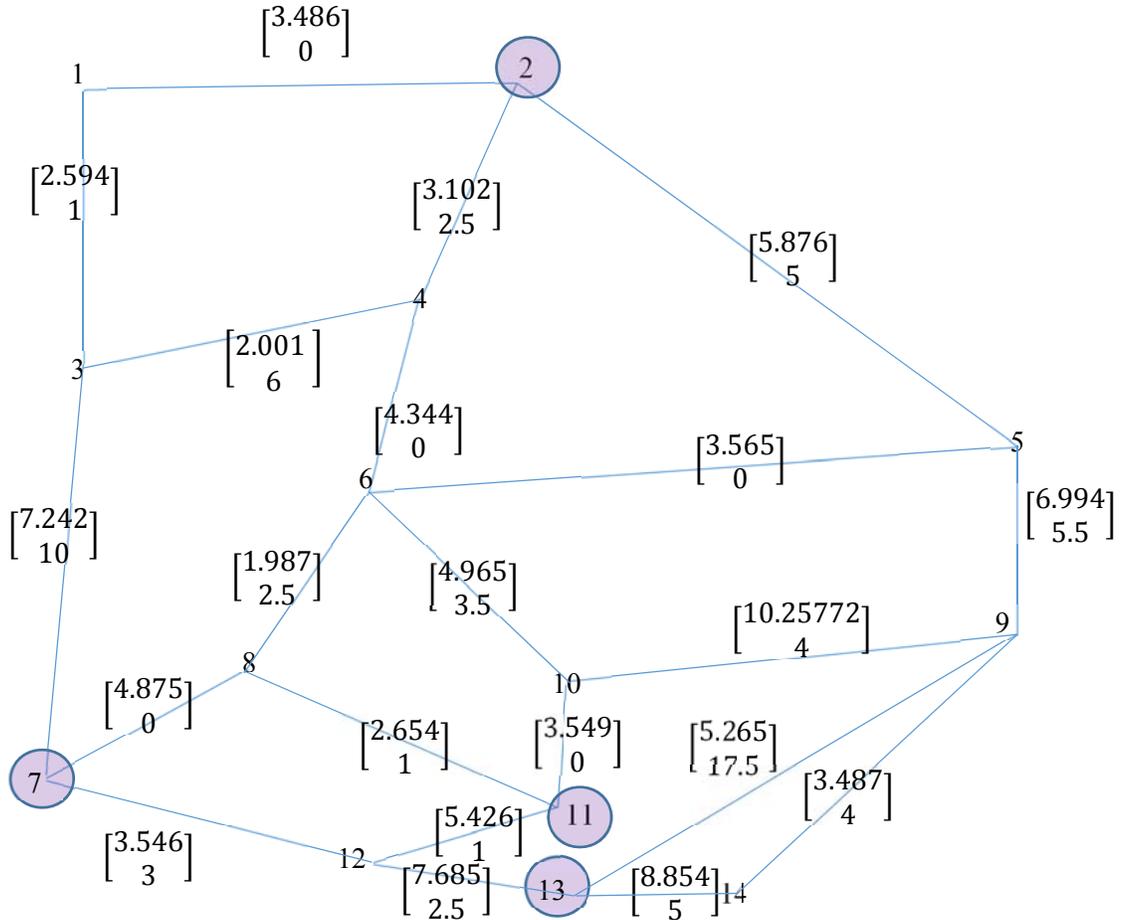


Figure 15. Cost and benefits of edges of network 3

Table 11. Results of network 3

Objective function	Time (sec)
196.5	491259.4

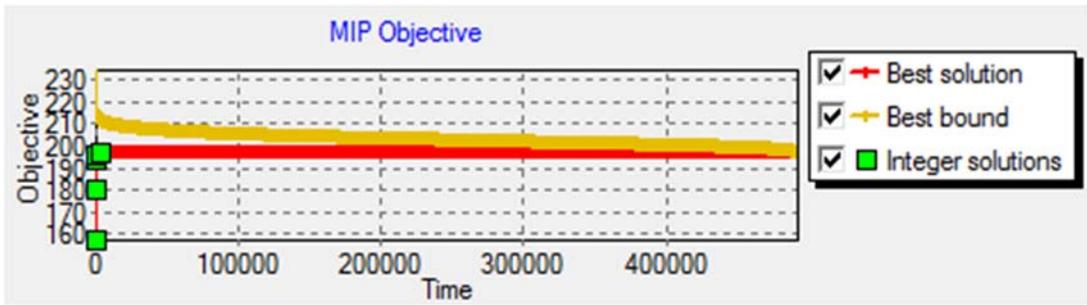


Figure 16. Pace of finding optimal solution (Xpress)

Several replications are made to find the best and worst solutions that the algorithm has found for this particular problem which may help to see the consistency of the result and the quality of the solutions.

A graph of improvement in the objective function with respect to time is plotted in Figures 17 and 18 to show the accuracy of the metaheuristic algorithm for a few replications.

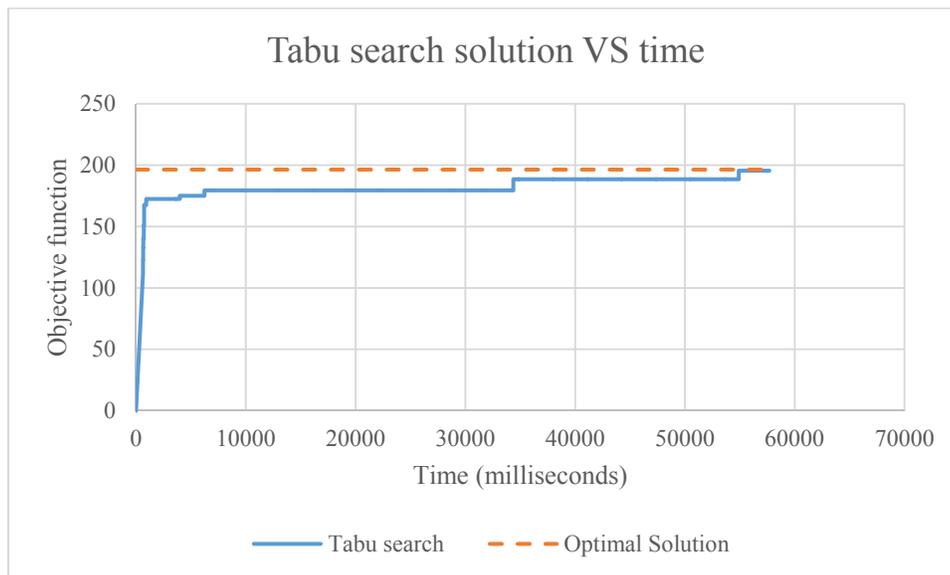


Figure 17. The objective function values with respect to time for tabu search and the optimal solution.

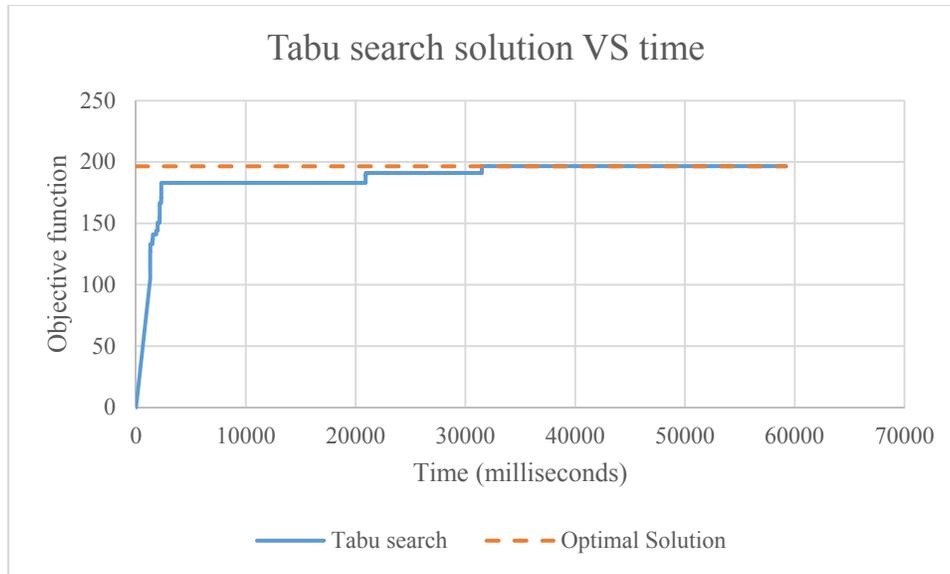


Figure 18. The objective function values with time for tabu search and the optimal solution.

It is found that the best solution has the optimal value of the objective function and the worst one (184) has the error of approximately 6%, which shows the good performance of the algorithm.

Table 12 shows the statistical parameters for 40 replications of the algorithm. Although, we have considered 450 seconds as the maximum time for running the algorithm, the average time needed to find the best solution in these 40 replications is 170 seconds.

Table 12. The statistical parameters for 50 replications of the algorithm

Number of replications	Average	maximum	minimum	mode	Standard Deviation	Coefficient of variance
50	190.7	196.5	184.5	193.5	3.9	0.020278

Figure 19 shows the variation of the objective function values with respect to different replications. The results of the developed algorithm for this problem is found to be consistent.

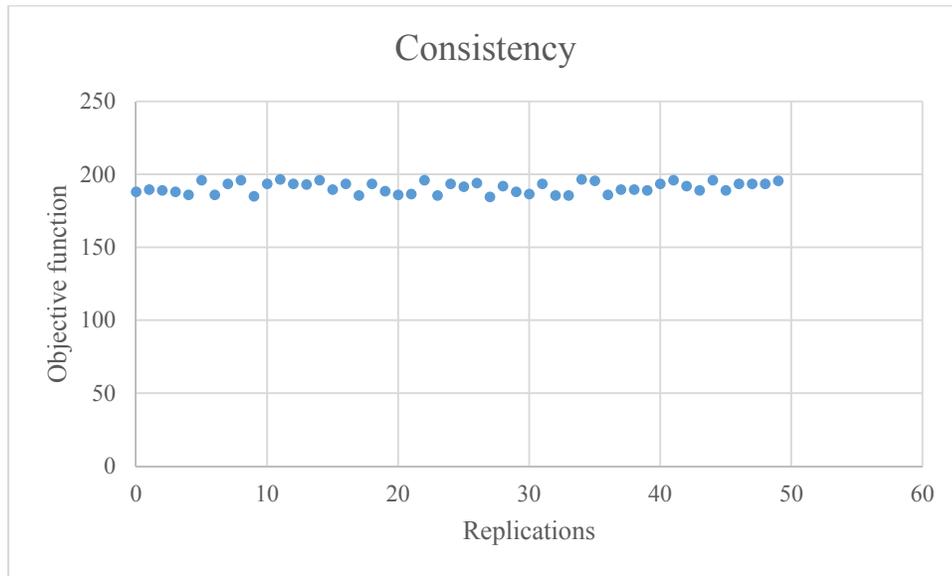


Figure 19. The consistency of the result for this problem.

One of the assumptions that we have considered in this study is that all edges of the network need to be covered at least once. However, the mathematical model presented is also flexible for covering some mandatory edges of the networks. For example, if our goal is to cover only those edges of the network with positive value of crime rate, we just need to make an exception for edges with zero crime rate in constraint (2) in the mathematical model.

In order to show the result, we have tested this assumption on the new network. The objective function and time needed to solve the problem is shown in the Table 13.

Objective function	Time (sec)
118	7205.5

Table 13. Objective function value and running time of Xpress

The routes found by Xpress is shown in the Table 14.

Vehicles	Routes											
Veh 1	2	4	3	7	6	5	4	2				
Veh 2	7	3	1	2	4	3	7					
Veh 3	10	5	4	2	10							
Veh 4	12	9	11	12	8	7	6	9	12	11	9	12

Table 14. Routes for vehicles (nodes in order)

Moreover, for solving this problem with tabu search the algorithm can determine the required edges of the network based on the value of the crime rate. In this way, the algorithm tries to cover the edges with positive crime rate and considers the others as the way for reaching the required edges of the network.

Network 4 (Large size)

The exact method, however, is not applicable for larger size networks such as the city of College Park. Finally, we move to the last part, which is examining the algorithm on the real size network. The network that has been considered for this problem is the network of the city of College Park in Maryland. The boundaries of the city of College Park have been shown in Figure 20.

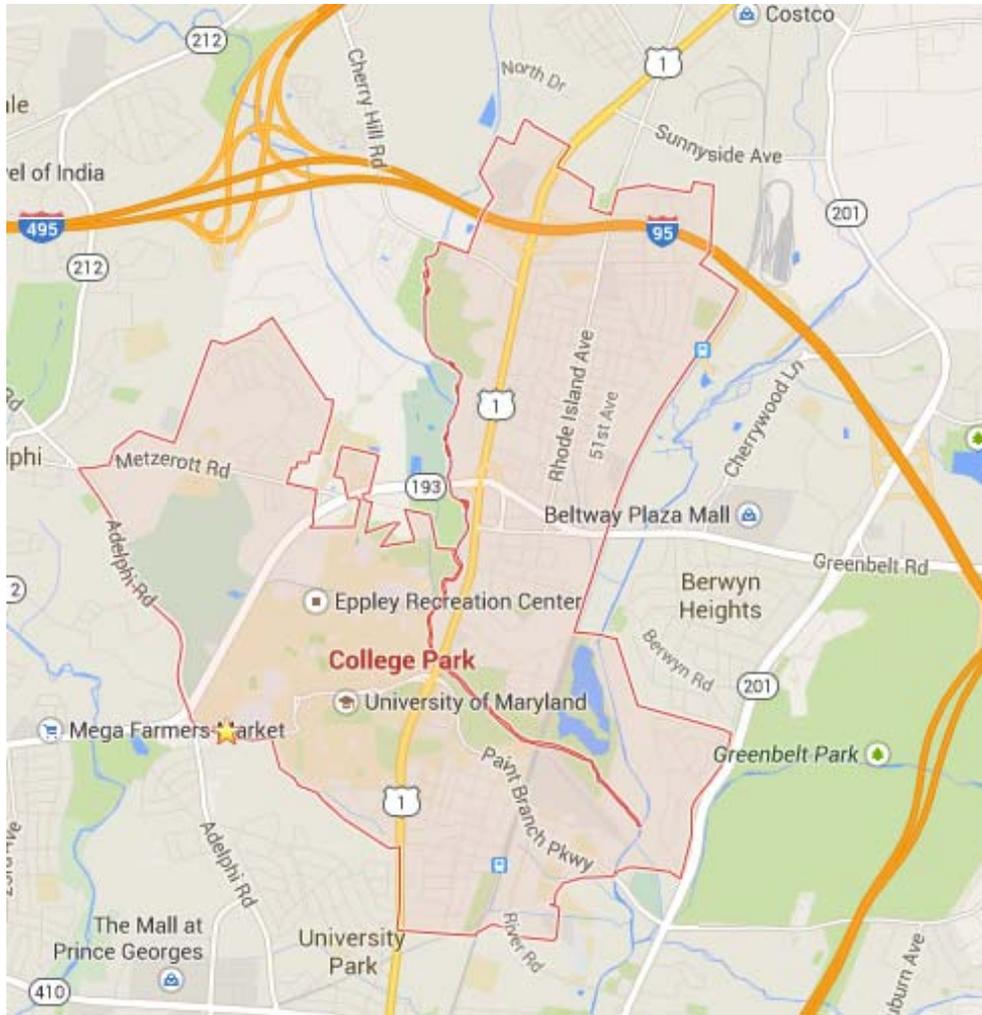


Figure 20. City of College Park

The overall view of the network and the streets that have been considered for covering are shown in Figure 21.

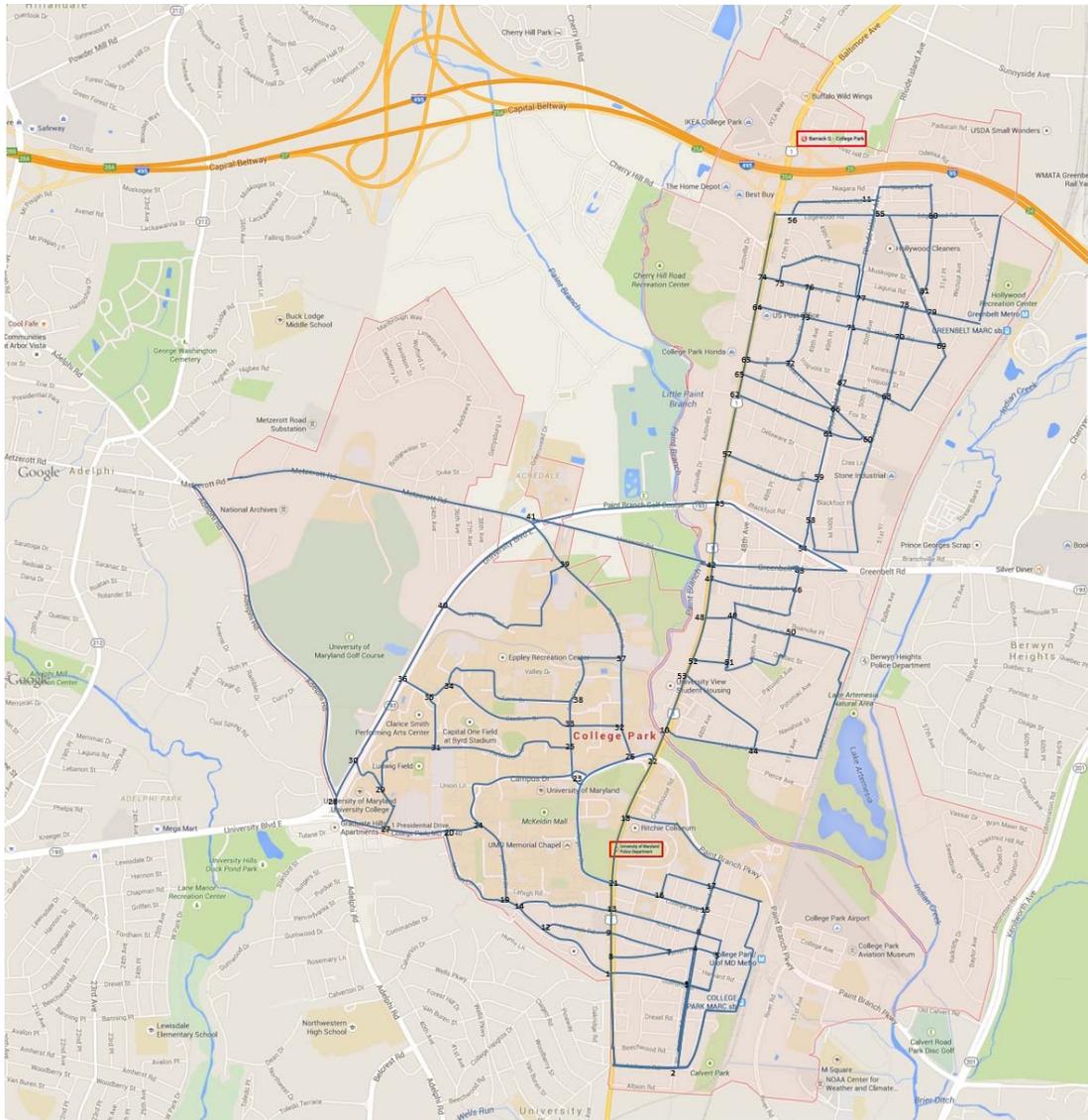


Figure 21. The overall view of the map to be covered by patrolling vehicles

For this network, 21 vehicles and 7 different depot are considered. These depots have been considered to show the performance of the tabu search and they are not based on the real starting point of patrolling vehicles. The costs of the edges are travel times and the benefit is achieved based on the crime rates reported on [crime reports](#) of two months February and March 2015. The travel times are based on the travel times reported by google maps. 45 replications with the maximum time of 20,000 seconds

has been considered for finding the solution to the problem. The duration for each cycle for this network is 20 minutes, which results in at least 72 visits during a day. The statistical analysis of the result provided by the tabu search algorithm is given in Table 15.

Table 15. The statistical parameters for 45 replications of the algorithm

Number of replications	Average	maximum	minimum	mode	Standard Deviation	Coefficient of variance
50	356.2	380	347	357	6.3	0.017719

The best result, which is found by the algorithm, has the objective function with the value of 349. In the following, two solutions of the algorithm for covering the whole network have been shown in Tables 16 and 17. Having various solutions by several replications results in more reliable procedure for increasing security and safety. In order to show the randomness feature of the solution, two solutions shown in Tables 16 and 17 are compared. Different parts of similar routes in each solution have been colored. The last column of Table 17 shows the similar route number of solution 1 for each route of solution number 2.

Table 16. Solution 1

Vehicle number	Routes																		
Veh 1	10	22	26	32	33	34	35	31	25	33	38	37	32	26	22	10			
Veh 2	10	22	26	32	37	39	40	36	35	34	38	33	32	26	22	10			
Veh 3	10	22	18	21	13	9	12	1	2	17	15	16	21	18	22	10			
Veh 4	10	22	26	23	24	20	27	28	41	39	37	32	26	22	10				
Veh 5	19	24	20	19	14	12	1	3	4	7	9	8	1	12	14	19			
Veh 6	19	14	12	9	13	6	15	16	21	13	9	7	8	1	12	14	19		
Veh 7	19	14	12	1	2	3	5	4	6	15	16	21	23	24	19				
Veh 8	29	27	28	30	36	35	34	38	33	34	35	31	29						

Veh 9	29	27	20	24	19	14	13	6	5	3	1	12	14	19	20	27	29		
Veh 10	29	31	25	33	38	34	33	38	34	35	36	30	29						
Veh 11	34	35	36	40	41	42	47	48	52	53	10	22	26	32	33	34			
Veh 12	34	33	32	26	22	18	17	15	16	17	18	22	26	32	33	34			
Veh 13	34	35	36	40	41	43	57	62	65	63	64	74	56	11	21	23	25	33	34
Veh 14	50	51	52	48	49	46	47	48	49	50	51	49	50						
Veh 15	50	44	53	10	44	50	51	52	48	49	50								
Veh 16	50	44	10	22	26	32	37	39	41	43	54	45	46	49	50				
Veh 17	67	68	60	61	66	72	73	71	67	66	61	59	57	62	61	66	60	68	67
Veh 18	67	68	60	58	54	45	42	43	54	58	59	61	62	65	66	67			
Veh 19	67	68	69	70	68	67	66	72	73	76	77	78	79	81	55	77	71	67	
Veh 20	79	69	70	71	73	72	63	64	73	76	75	77	71	70	78	79			
Veh 21	79	80	55	77	78	79	81	80	11	55	56	74	75	77	78	79			

Table 17. Solution 2

Vehicle number	Routes																			Similar vehicle number in Sol 1	
Veh 1	10	22	18	21	13	6	4	3	1	12	14	13	21	23	26	22	10			veh 3	
Veh 2	10	22	26	32	33	34	35	31	25	23	26	22	10							veh 1	
Veh 3	10	22	18	17	2	1	12	9	13	21	23	26	22	10						veh 2	
Veh 4	10	22	18	21	16	17	15	16	17	18	22	10	44	53	10					veh 4	
Veh 5	19	24	20	19	14	12	9	8	7	4	3	5	4	7	9	12	14	19		veh 5	
Veh 6	19	24	23	26	32	33	38	37	32	33	25	23	24	19						veh 6	
Veh 7	19	14	12	1	2	3	5	6	15	16	21	23	24	19						veh 7	
Veh 8	29	30	28	27	29	30	36	35	34	33	25	31	29							veh 8	
Veh 9	29	31	35	36	30	29	31	35	34	38	33	34	35	36	30	29				veh 10	
Veh 10	29	27	20	19	14	12	9	8	1	12	14	19	20	24	19	20	27	29		veh 9	
Veh 11	34	35	36	40	41	43	57	62	65	63	64	74	56	11	21	23	25	33	34	veh 13	
Veh 12	34	35	31	25	33	34	35	36	30	29	31	35	34							veh 12	
Veh 13	34	35	36	30	28	41	39	40	36	35	34									veh 11	
Veh 14	50	51	52	48	47	42	43	57	59	58	54	45	42	47	46	49	50			veh 14	
Veh 15	50	49	51	52	53	10	22	26	32	37	39	41	43	54	45	46	49	50		veh 16	
Veh 16	50	44	53	10	22	26	32	37	39	41	42	47	48	49	50					veh 15	
Veh 17	67	66	61	62	65	66	60	58	59	61	62	65	66	60	68	67				veh 17	
Veh 18	67	66	72	73	64	74	56	55	77	71	67	68	69	70	71	67	68	70	71	67	veh 19
Veh 19	67	68	70	78	79	80	55	77	76	75	74	64	73	72	63	64	73	71	67		veh 18
Veh 20	79	69	70	71	67	68	60	61	62	65	63	64	73	76	75	77	78	79			veh 20
Veh 21	79	69	70	78	79	81	80	11	55	81	79	69	70	78	79						veh 21

Table 18 shows the min, max and average time for finding the best solution in 45 replications.

Table 18. Running time for the best solutions

Time(seconds)	Min	max	average
Running time	94.172	1979.469	1069.157

Figures 22 and 23 show the variation of the objective function VS time for the network.

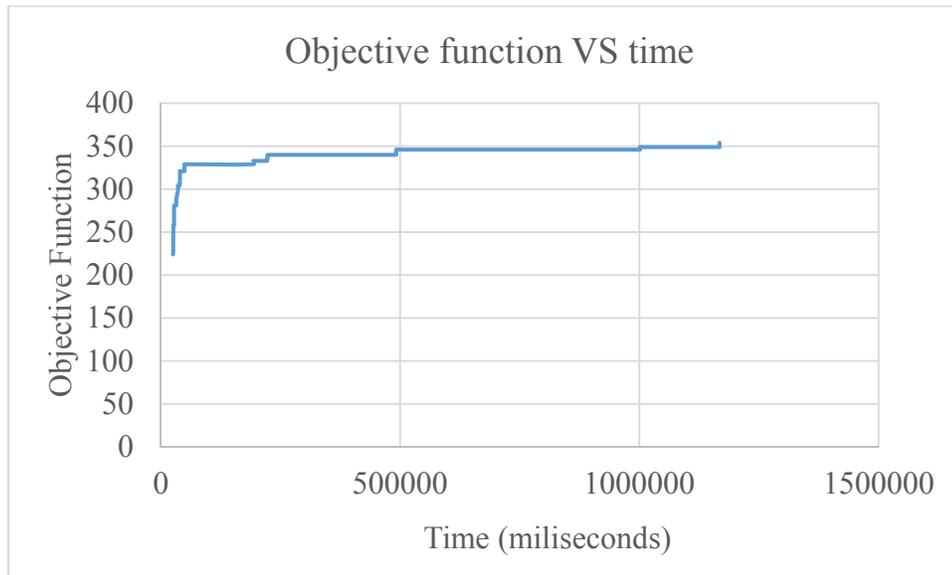


Figure 22. The objective function values with respect to time for tabu search

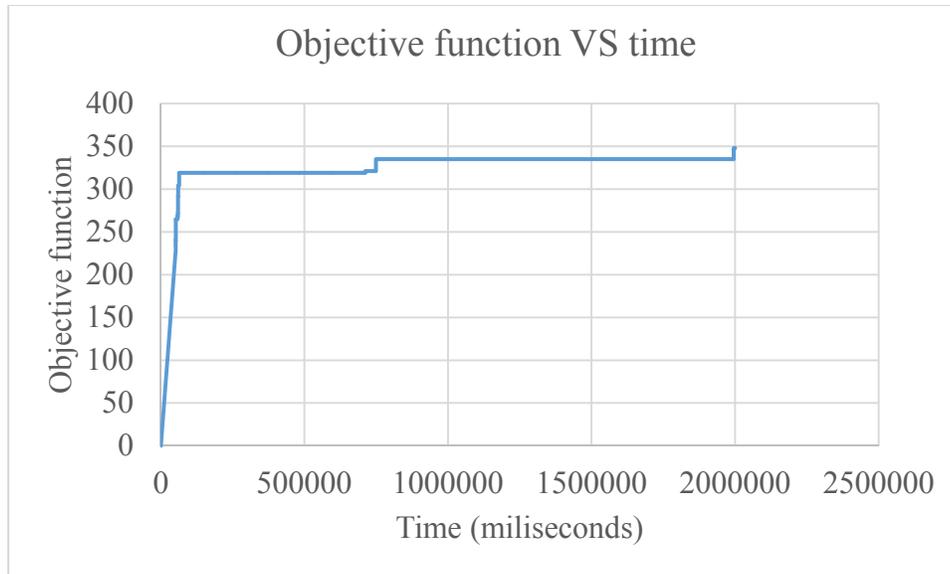


Figure 23. The values of the objective function with respect to time for tabu search

Figure 24 shows the consistency of the result for this problem.

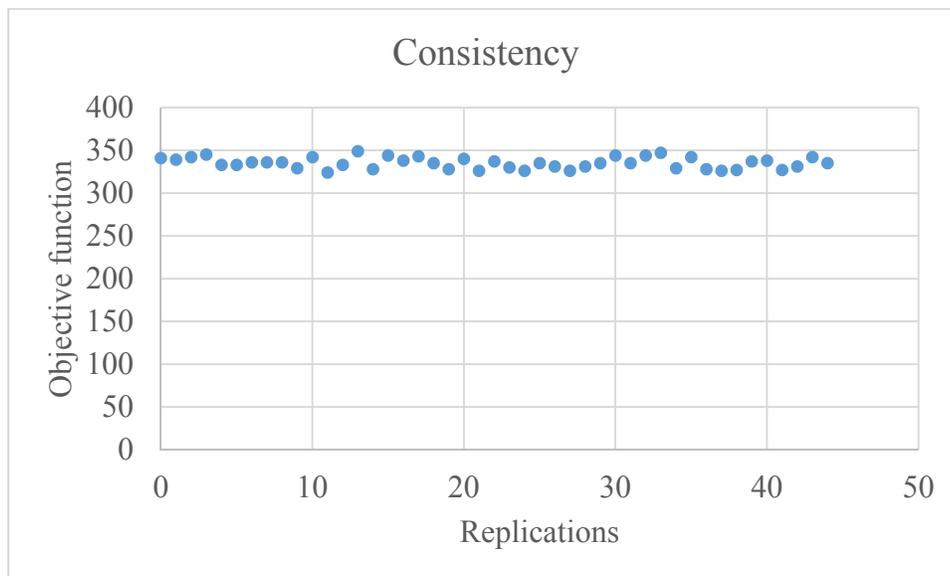


Figure 24. The consistency of the result for this problem.

In this section, the proposed heuristic approach to solve the Maximum Benefit k-Chinese Postman problem was discussed. In addition to the differences stated in the

section 3 for the developed formulation, the heuristic approach results in random solutions. This characteristic makes this algorithm very special for the patrolling vehicles in case of increasing safety. In other words, having random solutions impede constructing routine routes, which are predictable by potential criminals. Moreover, the algorithm provides high quality solutions in reasonable time. Although, we have tested the algorithm on undirected graphs, it is applicable to directed graphs too. We present our work in the context of routes for police patrols on a road network. However, our methods are applicable to many other networks that are a function of surveillance and situation - response routes to maximize the benefit of the selected routes.

Chapter 6: Conclusion

The goal in this research was to find suitable routes for patrolling vehicles in the network with various depots to increase the safety of the residential area. This problem is a special case of Maximum Benefit k-Chinese Postman Problem. A mathematical formulation was introduced for solving the problem. The problem was tested on three small networks with different sizes using the presented formulation. It has been found that the Xpress needs much time for solving even small size networks. We also found that large size networks cannot be solved by mathematical modeling. The combinatorial optimization problem is NP-hard, which demonstrate the need of heuristic approach to obtain good solutions in reasonable required time for finding solutions. To overcome this issue, we proposed a heuristic approach (tabu search) to solve the problem.

The tabu search metaheuristic algorithm was proposed for solving the real size networks. It is known that, having the same routine routes for patrolling vehicles will not result in increasing the safety due to the potential determination of the schedule of patrolling vehicles by criminals. Therefore, random routes are required for patrolling vehicles. It is often impossible to determine random routes by hand. Therefore, a computer-based methodology is required to generate the random routes. The proposed algorithm serves this purpose well.

The developed metaheuristic algorithm gives appropriate solution in a reasonable time, which makes it usable for many applications and generate random solutions in different replications, which makes it very special for security purposes.

Although, for concreteness, the research is presented in the context of routes for police patrolling vehicles, the developed algorithm is applicable to many other environments having the objective of maximizing the benefit of the selected routes. Optimal solutions are also attainable for different objective functions for these problems. In other words, the same methodology can be applied to other definition of benefit in Maximum Benefit k -Chinese Postman Problem.

Crime history data helps us to assign the roads weights, which are proportional to their crime rates. The proposed algorithm can be used by police department to find the suitable cycles for patrolling vehicles, which gives the high value of benefit in terms of safety for residential areas.

In most of the previous studies, the aim is to find routes for a set of required edges. However, considering the real situation there are many cases in which we need to cover all the edges of the network. Although previous studies mostly focus on formulating the problem, these problems are non-deterministic polynomial-time hard (NP-hard) problem. Therefore, developing a heuristic algorithm is necessary to find solutions for the real networks.

One of the drawbacks of the previous studies is that an edge can be traversed several times in a row, which results in an unreal value of the objective function. In the current study, traversing an edge two times in a row is not allowed. As a result, realistic routes for patrolling vehicles have been found by the algorithm to increase the residential safety.

We consider a general network with more than one station for starting and ending routes. To the best of the author's knowledge, this feature rarely has been considered in previous studies.

Future Studies

The optimization of vehicles routes has great potentials for future work. For example, the total number of patrolling vehicles and the number of patrolling vehicles at each depot can also be optimized.

In this study, we have considered a linear relationship between total benefits and the number of times that each edge is traversed. However, a nonlinear functional form for benefit versus number of traversals can make the model more realistic.

The assumption of going back to the starting depot after each cycle is considered as a constraint in the current study. For the future work, this constraint can be relaxed and the best route can be obtained instead of having a loop for each round of covering the whole network.

For the future, other heuristic methods such as Genetic Algorithm (GA) and Simulated Annealing (SA) can be developed to evaluate the solutions based on the quality and running time.

Moreover, for the maximization problem we can find strategies for getting the upper bound for the solution, which help to see the quality of the result for the large size networks. Furthermore, a sensitivity analysis can be performed on the input data such as travel times of edges.

We demonstrated a large network to show the efficiency of the methodology. We have found routes for patrolling vehicles based on the constant benefit for all day.

However, for finding routes for patrolling vehicles, we may divide one day into several shifts based on the crime rate of the roads in daylight or at night.

The same methodology can be applied to different objective functions. For example, we can consider the crime rate reduction of each edge or geographic equity as the benefit. Population of each area of the network can be one of the other important factor for assigning patrolling vehicles and finding benefit of the edges.

References

- Ahr, D. & Reinelt., G., 2006. A tabu search algorithm for the min–max k-Chinese postman problem. *Computers & Operations Research* 33, no. 12 , pp. 3403-3422.
- Aráoz, J., Fernández, E. & Franquesa, C., 2009. The clustered prize-collecting arc routing problem.. *Transportation Science* 43, no. 3, pp. 287-300..
- Aráoz, J., Fernández, E. & Zoltan, C., 2006. Privatized rural postman problems. *Computers & operations research* 33, no. 12, pp. 3432-3449..
- Archetti, C. et al., 2014. A branch-and-cut algorithm for the orienteering arc routing problem.
- Archetti, C., Feillet, D., Hertz, A. & Speranza, G., 2010. The Undirected Capacitated Arc Routing Problem with Profits. *Computers & Operations Research*, 37(11), pp. 1860-1869.
- Assad, A. A. & Golden, B. L., 1995. Arc routing methods and applications.. *Handbooks in operations research and management science* 8, pp. 375-483..
- Benavent, E. et al., 2014. Profitable mixed capacitated arc routing and related problems. *TOP*, pp. 1-31.
- Brandão, J. & Eglese., R., 2008. A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research* 35, no. 4, pp. 1112-1126.
- Calvo, R. W. & Cordone., R., 2003. A heuristic approach to the overnight security service problem.. *Computers & Operations Research* 30.9, pp. 1269-1287.
- Corberán, Á., Fernández, E., Franquesa, C. & Sanchis, J. M., 2011. The windy clustered prize-collecting arc-routing problem.. *Transportation Science* 45, no. 3, pp. 317-334..
- Corberán, Á., Isaac, P., Antonio, M. R.-C. & José, M. S., 2013. A branch-and-cut algorithm for the maximum benefit Chinese postman problem.. *Mathematical Programming* 141, no. 1-2, pp. 21-48.
- Corberán, A., Plana, I. & Sanchis, J. M., 2006. Zigzag inequalities a new class of facet-inducing inequalities for Arc Routing Problem.. *Mathematical programming* 108, no. 1, pp. 79-96..

- Corberán, A., Plana, I. & Sanchis, J. M., 2008. The windy general routing polyhedron: a global view of many known arc routing polyhedra.. *SIAM Journal on Discrete Mathematics* 22, no. 2, pp. 606-628.
- Corberán, A. & Prins, C., 2010. Recent results on arc routing problems: An annotated bibliography. *Networks* 56.1, pp. 50-69.
- Dantzig, G. B. & Ramser, J. H., 1959. The truck dispatching problem.. *Management science*, 6(1), 80-91..
- Degenhardt, J., 2004. An ant-algorithm for the balanced k-chinese postmen problem. *Operations Research Conference*. .
- Feillet, D., Dejax, P., Gendreau, M. & Gueguen., a. C., 2005. The Profitable Arc Tour Problem: Solution with a Branch-and-Price Algorithm. *Transportation Science* 39.4, pp. 539-552.
- Frederickson, G. N., 1979. Approximation algorithms for some postman problems.. *Journal of the ACM (JACM)* 26, no. 3 , pp. 538-554..
- Grötschel, M. & Win., a. Z., 1992. A cutting plane algorithm for the windy postman problem. *Mathematical Programming* 55, no. 1-3, pp. 339-358..
- Kwan, M.-K., 1962. Graphic programming using odd or even points. *Chinese Math* 1.273-277 , p. 110.
- Lacomme, P., Prins, C. & Ramdane-Chérif, W., 2001. A genetic algorithm for the capacitated arc routing problem and its extensions. *In Applications of evolutionary computing*, pp. 473-483. *Springer Berlin Heidelberg*.
- Lysgaard, J., 1997. Clarke & Wright's Savings Algorithm.. *Department of Management Science and Logistics, The Aarhus School of Business*.
- Minieka, E., 1979. The Chinese postman problem for mixed networks.. *Management Science* 25, no. 7, pp. 643-648..
- Oliveira, W. A., Mello, M. P., Moretti, A. C. & Reis., E. F., 2013. The multi-vehicle covering tour problem: building routes for urban patrolling.. *arXiv preprint arXiv:1309.5502*..
- Osterhues, A. & Mariak., F., 2005. On variants of the k-Chinese Postman Problem. *Operations Research and Wirtschaftsinformatik, University Dortmund* 30.

- Palma, G., 2011. A Tabu Search Heuristic for the Prize-collecting Rural Postman Problem.. *Electronic Notes in Theoretical Computer Science 281*, pp. 85-100..
- Pearn, W. L. & Wang., K. H., 2003. On the maximum benefit Chinese postman problem. *Omega 31, no. 4*, pp. 269-273..
- Reis, D., Adriano, M., André, L. C. & Vasco, F., 2006. Towards optimal police patrol routes with genetic algorithms. *In Intelligence and Security Informatics, pp. 485-491. Springer Berlin Heidelberg, .*
- Rosenshine, M., 1970. Contributions to a theory of patrol scheduling.. *Operational Research Quarterly, 99-106.*
- Shafahi, A. & Haghani., A., 2015. Balanced Routing of Patrolling Vehicles Focusing on Areas with Historical Crime.. *Transportation Research Board 94th Annual Meeting, No. 15-4387. 2015.*
- Shafahi, A. & Haghani, A., 2015. Generalized Maximum Benefit Multiple Chinese Postman Problem. *Transportation Research Part C: Emerging Technologies 55*, pp. 261-272.
- Sherman, L. W. & Weisburd, D., 1995. General deterrent effects of police patrol in crime “hot spots”: A randomized, controlled trial.. *Justice Quarterly, 12(4)*, 625-648..
- Vincent, F. Y. S.-W. L. W. L. a. C.-J. T., 2010. A simulated annealing heuristic for the capacitated location routing problem.. *Computers & Industrial Engineering 58, no. 2*, pp. 288-299..
- Willemse, E. J. & Joubert, J. W., 2012. Applying min-max k postmen problems to the routing of security gaurds. *Journal of the Operational Research Society, 63(2)*, pp. 245-260.
- Zachariadis, E. E. & Kiranoudis, C. T., 2011. Local search for the undirected capacitated arc routing problem with profits.. *European Journal of Operational Research 210, no. 2*, pp. 358-367.