Solving a directed profitable rural postman problem using two meta-heuristic algorithms

Masoud Rabbani, Fahimeh Taghiniam
School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Abstract

In transportation companies, the costs are highly dependent on the routes in which the customers are served. We investigate the problem where a company that should serve a set of customers has two internal vehicles and also delegates some services to external contractor. The problem is to determine the set of customers to outsource with the goal of minimizing the total cost including traveling cost (routing cost) and outsourcing cost (penalty cost). In the other words, the problem can be stated as a maximization of the profit which can be calculated by the difference between a incomes gained from servicing customers and the total cost. We call this problem the Directed Profitable Rural Postman Problem (DPRPP). We solve the problem with two meta-heuristic algorithms including Tabu search (TS) and Genetic algorithm (GA). Then, computational experiments are conducted and two algorithms are compared. The obtained results show the good performance of proposed procedures.

Keywords: rural postman problem, directed graph, tabu search, genetic algorithm.

1. Introduction

The Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) are categorized as the most widely studied problems in operation research field (e.g., see Gutin and Punnen [18] and Toth and Vigo [22]). Both of these problems aim to serve the all customers in the optimally manner. To the best of our knowledge so limited number of papers have addressed the case in which a subset of the customers can be selected for servicing. Feillet, Dejax, and Gendreau [15] introduced the Profitable Arc Tour Problem (PATP). The objective function of the PATP is to maximize the profit gained by traversing on arcs minus the traveling cost applying a set of unlimited vehicles with no capacity limitation, but an upper bound is available on the length of routes. There is not any depot and the profit is available on each arc for a limited number of times. In [19] a routing problems in which a profit is associated with customers and there is no compulsion to serve all the customers is investigated. The objective function might be the maximization of the total gained profit, or the minimization of the traveling cost or the maximization of the difference of them. If only one vehicle is available, the problems will be named Orienteering Problem, Prize-Collecting TSP and Profitable Tour Problem.

In most cases with customer selection, customers are represented by vertices in a graph network. Only few authors specify customers by edges (arcs) of an undirected or directed graph. We refer to the book by Dror [13] for an overview on classical arc routing problems, in which all customers need to be served by vehicles. To the best of our knowledge, Malandraki and Daskin [20] was the first authors to address arc profits by introducing a directed version of the prize-collecting arc routing problem. In their problem, a prize can be gathered from each arc several times, and the size of the prize decreases with the number of collections. This is the case of the one-period Bus Touring Problem (BTP) proposed by Deitch and Ladany [6]. In the BTP the obtained profits are non-negative attractiveness values and the objective is to maximize the total attractiveness of the route by determining a subset of nodes which should be visited and arcs should be traveled with respect to constraints on time, cost and total distance. Feillet et al. [15] presented the profitable arc tour problem that wants to maximize the net profit considering constraints which is related to restriction in the number of times the profit is accessible on the arcs and the maximum length of tours. An unlimited set of uncapacitated vehicles are available for the servicing customers. There is not any depot; that is, each vehicle route can start and end at any vertex. The Prize-collecting Rural Postman Problem (PRPP) is introduced by Araoz et al. in [1] and its attributes are investigated. This problem can be seen as a generalization of the classical Rural Postman Problem (RPP) in which instead of required edges, a set of profitable edges are available and the objective is to find the tour maximizing the difference between the gained profit and the traveling costs. Moreover, The PRPP is categorized
as the undirected version of the DPRPP. Some papers are available on the literature such as the clustered prize-collecting ARP by Araoz, Fernandez, and Franquesa [2], and the windy clustered prize-collecting ARP by Corberan, Fernandez, Franquesa, and Sanchis [10]. Recently, Black, Eglese, and Wohlk [12] studied the time dependent prize-collecting Arc Routing Problem (TD-PARP) inspired from the PRPP, but with consideration of real constraints. The graph is directed and traveling costs depends on the time the arc are crossed. Moreover, the problem allows for more requests between the same two nodes. Finally, in the literature there are problems in which profits are occurred on nodes and also arcs. An exact algorithm for the PRPP is introduced and examined by Araoz et al. in [2], whereas its clustered version is solved by the same authors in [1]. Archetti et al. [3] investigated the undirected capacitated arc routing problem with profits. Traveling time is associated with each arc of the graph. A profit and a demands in this problem are associated with each edge of a set of profitable edges of the graph. A fleet of capacitated vehicles is considered to serve the profitable edges, and a maximum length of the route for each vehicle is also addressed. The objective is to find a set of routes in the way that maximize the total gained profit. In this paper, we propose the problem in which a subset of the arcs are representative for the customers and a penalty is associated with each customer. Also, a traveling cost is considered for each arc of the graph. Two uncapacitated vehicles are available. If the vehicles do not serve a customer, then the associated penalty will be paid and when two vehicles serve one arc, the penalty should be assigned. The problem aims to minimize the sum of the routing costs and the penalty costs. Since the problem can be expressed as the maximization of the difference between a profit gained from the served customers and the routing costs, we call this problem Directed Profitable Rural Postman Problem (DPRPP). The DPRPP is introduced by Guastaroba et al. in [17] in the field of truckload transportation service procurement called Shipper’s Lane Selection Problem (SLSP). In such problems, customers ask a truckload transportation service which should be provided from an origin point and directly sent to a destination point. This pair of origin–destination points is usually referred to as a lane. In this type of problem lanes can be defined in a graph as arcs to be traversed. In [17] the authors analyzed the problem in which a shipper should to decide which lanes to be served and which ones to be outsourced through an electronic auction. If the shipper vehicles are not enough to fulfill all the transportation requests, the shipper has to delegate some of customers’ servicing. Even, if the fleet is enough for servicing, it may be beneficial to delegate some of the lanes to external fleet, since other contractors may be able to satisfy some requests at a low level cost. Some authors have investigated the problems in which customers who are represented by vertices, can be served by one of the vehicles whether an internal fleet or an external contractor. In these problems, the objective is usually to minimize the sum of the traveling cost plus the total cost charged by the contractors. Chu [8] considered the case in which the total demand of customers is higher than the capacity of the internal fleet. The author presented a new mathematical model and one heuristic algorithm in order to minimize total cost function. Côté and Potvin [9] proposed a quite different mathematical formulation and applied a tabu search for solving the problem. The DPRPP is a generalization form of the Directed Rural Postman Problem (DRPP) (e.g., see Christofides et al. [7]) that aims to determine a shortest directed cycle including each arc associated with a customer at least once. The DRPP are different from the DPRPP since no selection among the customers is allowed and it is obtained from the DPRPP by setting the penalties to a enough large value. The DRPP and DPRPP are proven to be NP-Hard problem [14]. The most similar problem to the DPRPP proposed in the literature is the aforementioned PRPP which is introduced by [1]. The main difference between them is that in the PRPP transportation services are associated with a subset of edges of an undirected graph. The service associated with an edge is satisfied when the edge is traversed by vehicles. In the DPRPP transportation services are associated with a subset of arcs of a directed graph, and the service associated with an arc is fulfilled if and only if the arc is traversed. In the truckload application, the direction of the service is an important attribute of the problem. This difference implies that the mathematical model presented in [1] and the methodology proposed by Araoz et al. in [2] are not appropriate for applying in the DPRPP. Recently, the SLSP has been reformulated by Archetti, Guastaroba, and Speranza [17] and the named this problem Directed Profitable Rural Postman Problem and implemented an efficient Tabu Search algorithm with the addition of an ex-post ILP refinement. They also provided a set of benchmark examples for the problem which some of them have not been solved to optimally up to now. We solved the problem proposed in this paper by means of two algorithms. These algorithms are Tabu search (TS) and Genetic algorithm (GA). Then, the results obtained from these two algorithm are compared with each other. The remainder of this paper is organized as follows. In Section 2 the problem is introduced and the mathematical formulation is presented. In Section 3 descriptions about GA and TS algorithms are provided. The computational results are presented in Section 4. In Section 5, we compare the results between two algorithms. Finally discussion and conclusion remarks are provided.
2. The Directed Profitable Rural Postman Problem

We consider a directed graph $G = (V, A)$, in which $V = \{0, 1, \ldots, n\}$ is the set of vertices and $A$ is the set of arcs $(i, j)$ $(i \neq j)$. A set $R \subseteq A$ is given in a way that a transportation service is associated with each arc $(i, j) \in R$. The service is fulfilled if and only if a vehicle travels directly from node $i$ to node $j$. A nonnegative traveling cost $a_{ijk}$ associates with each arc $(i, j) \in A$. We assume that the traveling costs $b_{ijk}$ satisfy the triangle inequality. Vertex 0 is the depot and is the starting and ending point of any route. Two uncapacitated vehicles are available to traverse any subset of the arcs in $R$. If the arc from $i$ to $j$, with $(i, j) \in R$, is not traversed, a penalty $b_{ijk}$ will be paid. If both vehicles cross a common arc (not common vertices), a penalty $c_{ij}$ will be paid. The problem aims to find a route that minimizes the total cost which is the sum of the traveling cost and the penalties cost.

2.1. Mathematical formulation

Let $x_{ijk}$ be a non-negative integer variable representing the number of times the vehicle $k$ travels directly from $i$ to $j$, with $(i, j) \in A$. Moreover, let $y_{ijk}$ denotes a binary variable, associates with any arc $(i, j) \in R$ and it takes value 1 if the vehicle travels directly from node $i$ to node $j$; 0 otherwise. Let $Z_{ik}$ denotes a binary variable that takes value 1 if vertex $i \in V \setminus \{0\}$ is visited by the vehicle $k$. Let $W_{ij}$ be a binary variable that takes value 1 if both vehicles cross from common arc. Then, the DPRPP can be formulated as the following ILP model:

Minimize $\sum_{(i,j)\in A} \sum_k a_{ijk} x_{ijk} + \sum_{(i,j)\in R} \sum_k b_{ijk} (1 - y_{ijk}) + \sum_{(i,j)\in A} c_{ij} W_{ij}$ \hspace{1cm} (1)

Subject to:

$\sum_{j \in V \setminus \{i\}} x_{ijk} = \sum_{j \in V \setminus \{i\}} x_{ij}$ \hspace{1cm} (2)

$x_{ijk} \geq y_{ijk}$ \hspace{1cm} (3)

$\sum_{j \in V \setminus \{i\}} x_{ijk} \leq (|R| + 1) Z_{ik}$ \hspace{1cm} (4)

$\sum_{j \in P \setminus \{i\}} x_{ijk} \geq \sum_{j \in P \setminus \{i\}} Z_{ik}$ \hspace{1cm} (5)

$k \in E$ \hspace{1cm} (6)

$x_{ijk} \in N$ \hspace{1cm} (7)

$y_{ijk} \in \{0,1\}$ \hspace{1cm} (8)

$Z_{ik} \in \{0,1\}$ \hspace{1cm} (9)

$W_{ij} \in \{0,1\}$ \hspace{1cm} (10)

The objective function (1) minimizes the total cost including traveling cost and the penalties cost. Constraint (2) which is called the classical in-degree and out-degree constraints implies that the number of arcs entering into vertex $i \in V$ is equal to the number of arcs leaving vertex $i$ for two vehicles. For each arc $(i, j) \in R$, constraint (3) guarantees that a binary variable $y_{ijk}$ takes value 0 if the arc is not traversed by the vehicle $k$, (i.e., $x_{ijk} = 0$). Conversely, if the arc is traversed by the vehicle $k$, (i.e., $x_{ijk} > 0$), the minimization of the objective function forces variable $y_{ijk}$ to take value 1. Constraint (4) imposes that if the vehicle crosses from arc $(j, i)$, this causes that vertex $i$ is visited and $Z_{ik} = 1$. It can be concluded that $|R| + 1$ is the maximum number of arcs entering vertex $i$ in an optimal solution where $|R|$ denotes the cardinality of set $R$. Constraint (5) ensures the connectivity in the routes. Constraint (6) guarantees that if two vehicles visit common arc, $W_{ij}$ takes value 1. The objective function can be also expressed as follow:

Maximize $\sum_{(i,j)\in R} \sum_k b_{ijk} y_{ijk} - \sum_{(i,j)\in A} \sum_k a_{ijk} x_{ijk} - \sum_{(i,j)\in R} \sum_k b_{ijk} - \sum_{(i,j)\in A} c_{ij} W_{ij}$ \hspace{1cm} (11)

where the last term in this objective function is constant. This objective function represents the maximization of the difference between a profits resulted from crossing from the arcs in $R$ and the traveling cost. This form of the objective function makes the relation between the DPRPP and the arc routing problems with profits more evident. In order to validate the proposed formulation, we recall the necessary and sufficient conditions for a
directed graph to be Eulerian. A directed graph G is Eulerian if and only if G is connected and every vertex i ∈ V has equal in-degree and out-degree [4].

3. Methodology

Two metaheuristic algorithm are used in this paper to solve the proposed problem. Many papers in the literature has applied genetic algorithm to solve the problem like the problem considered in this paper, and GA showed good performance in solving of these problems. So we select GA and the results obtained from this algorithm are compared with TS.

3.1. Genetic algorithm

Genetic algorithm is stochastic adaptive global search and optimization algorithm, introduced based on the principles of natural selection and population evolution. The population includes several individuals that represent potential solutions to the problem. The algorithm initializes by generation and evaluation a first generation population. Subsequently, a main loop of operations including selection of the best individuals, applying crossover operator and the applying mutation on every locus (string position) is performed repeatedly. The population of solutions evolves within this loop and the best string found along these iterations is considered to be the solution for the problem.

The main computational effort in a GA is related to the evaluation of the solutions quality, (i.e., the calculation of the fitness function). Computational time is very important, as each generation of the genetic algorithm should be iterated many times in order to generate an appropriate results for a problem. Usually, the fitness function is a complicated mathematical formulation with many parameters, while the operators of algorithm such as selection, mutation, crossover and replacement have linear complexity, often work at constant rates for a given problem [19]. Our implemented genetic algorithm applies the followings operators:

- the greedy stochastic algorithm is used to produce a portion of the initial generation
- a modified version of the single point crossover is applied for mating
- an auxiliary operation based on neighbor exchanges is used after each fixed number of generations to enhance the fitness of the solution pool [24].

3.1.1. Chromosome representation

A permutation of numbers between 1 to N is used to represent the solutions, where N is the number of nodes. In this problem, we use two chromosome. The First chromosome is related to the first vehicle and second chromosome associates with the second vehicle. Because each vehicle starts from node zero and crosses from number of arcs and then comes back to the zero node, so the number of components related to chromosome is variable. We decide to restrict the search space to just the feasible space; hence, the adopted and simple permutation representation is used for the chromosomes. The fitness value of a permutation is considered the cost of the associated schedule [24].

<table>
<thead>
<tr>
<th>Chromosome for the first vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  2  4  6  7  9  5  1  -  -</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chromosome for the second vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  3  6  6  8  5  2  4  7  1</td>
</tr>
</tbody>
</table>

Figure 1. Example of solution representation

Figure (1) shows that solution representation consists of two chromosomes. The first Chromosome states that the first vehicle passes from nodes: 1-2-4-6-7-9-6-3-1 and second Chromosome expresses the second vehicle route is 1-3-6-9-4-5-7-3-6-8-1.
3.1.2. Initial population

It is common approach in genetic algorithms to establish the algorithm with a randomly generated initial generation. Our initial population is composed of two distinguish classes of schedules [24]. First component in each string starts by node 1 and finished by node 1. In other word, this string is complete when number 1 is repeated again. When two numbers are similar in one chromosome, then the second number is deleted.

3.1.3. Parent selection

Tournament selection which is well-known strategy for selection of parents is applied in this paper: In this selection process, two individuals are selected randomly from the population and the better one is selected for reproduction [5].

3.1.4. Crossover operator

Crossover is the operator which combines the pool of solutions to generate offsprings. If this operator effectively controlled, the process can be used to produce efficient. Typically, deterministic updating of the match will cultivate limitations only over the distance of one neighborhood with each iteration. Crossover can hasten this process by mix disconnected yet internally consistent sub graphs from the individual solutions in the mating pool. The standard crossover procedure involves selecting at random pairs of global matching configurations from the current population. Matches at the corresponding sites at randomly chosen locations in the two graphs are then interchanged with uniform probability. However, this uniform cross-over mechanism will not necessarily facilitate the merging of locally consistent sub graphs. Moreover, the process also ignores the underlying structure of the graphs. A better strategy is to combine the solutions by physically dividing the graphs into two disjoint sub graphs. In this way internally consistent portions of the individual solutions may be exchanged at the structural level [11].

Several mating pool schemes have been addressed for permutation based representations in the literature. Some examples among these are partially matched crossover (PMX), cycle crossover (CX) and Order based crossover (OX). In addition, we also considered a new single point crossover operation adapted from PMX. Given this procedure, a random crossover point is selected and the portion to the left of this point is first exchanged between the mating chromosomes. This process may results in infeasible schedule and an additional repair step is needed. The repair stage is exactly similar to the repair process in PMX method. An example for this procedure is shown in Figure 2 to illustrate the process with $N = 8$ (the crossover point is shown with a solid triangle in each chromosome). In this example, the first child gets the values 1, 6, 8 and 7, and loses the values 1, 3, 4 and 2. As a result of this, it has duplicate value for 4, and is missing the values 6 and 7. We leave the portion to the left of the crossover point intact and substitute 6 with 4 and 7 with 2 to the right of the crossover point. Child 2 is also repaired similarly. We find that the single point crossover and PMX have the great performers. Therefore, we look ahead with the single point crossover operation. We adopt the crossover operation that performed the best among all the crossover operations that we considered [24].

\begin{figure}[h]
\centering
\begin{tabular}{l|l}
Initial configuration: & \\
Chromosome 1 & 1,3,4,2,6,7,5,1 \\
Chromosome 2 & 1,6,8,7,9,5,4,1 \\
Immediately after exchange: & \\
Intermediate child 1 & 1,6,8,7,6,7,5,1 \\
Intermediate child 2 & 1,3,4,2,9,5,4,1 \\
Final configuration (after repair): & \\
Child 1 & 1,6,8,7,5,4,9 \\
Child 2 & 1,3,4,2,6,5,7,1 \\
\end{tabular}
\caption{Example of crossover operator}
\end{figure}

3.1.5. Mutation
A more randomization section is used to the individual to introduce new information into the population of global matches by means of mutation operator. This operator performs randomly changes by randomly rearranging the matches on individual sites [11]. We examine some mutation options. Unlike in a binary string representation for chromosomes in which a gene is simply reversed if chosen for mutation, in a permutation representation, maintaining feasibility is additional problem. We examined several distinguish mutation operators. 2-opt operator changes the position of two randomly selected component. Neighbor swap mutation choses an element for mutation and swaps it with its neighbor to the right. In the three-point-swap operator, one element is chosen and then two others are also randomly chosen and these three elements are cyclically swapped. Finally, random-two-swap operator selects chromosome level rather than at the individual element for mutation. In this method, if a chromosome is selected for mutation, two of its elements are randomly chosen and swapped [24].

3.2. Tabu search algorithm

Tabu search is a powerful local search algorithm (someone also knows it as a metaheuristic algorithm) for tackling many combinatorial optimization problems and was proved to be effective for solving the DPRPP problem. Unlike to the GA which explores a single neighborhood, our implemented tabu search applies several neighborhood structures based on the one-flip and exchange moves to improve neighborhood search strategy. The mixed neighborhood structure was proposed in variable neighborhood search (VNS), which switches among different neighborhoods to escape from trapping in local optimum. Our implemented tabu search procedure repeatedly switches between two complementary structures of neighborhoods. One of them is the well-known one-flip method and another one is an innovative constrained exchange move. The two aforementioned methods are described in detail as follows [25].

With starting from one initial solution, the proposed tabu search algorithm first performs the one-flip operator based on tabu search phase, in which each iteration conducts either the best one-flip move that is not available in tabu list or a tabu move satisfying the aspiration level criterion. One move is considered as a tabu if it was performed during the last t1 iterations. Although this tabu criterion is able to prevent the search from going back to previously visited solutions, it may fail to care some unvisited efficient solutions. As a consequence, an aspiration criterion is performed which permits to perform a move if performing it yields to a solution with a better objective function value than the best found so far [25]. Basically, tabu search algorithm starts from an initial solution and iteratively keeps on from the current solution to its best allowable neighborhood. The structure of a neighborhood for one solution is defined as a set of solutions that can be reachable from this solution by some predefined operations. Each type of operation associates to a neighborhood. The transition from the current solution to one of its neighborhoods is named a move [21]. Our tabu search (TS) algorithm is based on the standard tabu search method. The main operations at the beginning is so that a solution calculated by ARMSA (initial solution) is stored as a current solution and also as the best solution. In our TS, a solution is coded as a permutation of all demands and to obtain the value of the solution, the demands node are allocated one by one according to the order and using the First Fit approach. At next stage, TS iteratively searches through the solution space to improve solution by applying a move which is not tabu to the current solution. As a result of this process, in each iteration a candidate solution is obtained. If this candidate solution is better than the best found solution, it is stored as the current and best solution, (i.e., an improvement in the objective function is reached). If the candidate solution is better only than the current solution, then it will be considered as a new current solution, but an improvement is not performed in this iteration. Finally, if the candidate solution is worse than the incumbent solution, there is no improvement and all stored solutions remain without any change. In addition, the searching process is adjusted by some memory structures and some diversification processes. A stopping criteria for TS is considered as a combination of a maximum number of iterations and an execution time limit [16]. In our tabu search, we have 5 types of actions, they are presented in Figure (3).

<table>
<thead>
<tr>
<th>Action 1</th>
<th>1 0 A 0 1 0 0 B 0 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>= Swap</td>
<td>1 0 B 0 1 0 0 A 0 0 1</td>
</tr>
</tbody>
</table>
Figure 3. List of actions in tabu search

4. Experimental results

In Example 1 which is shown in Figures 4-6, a graph G consists of six nodes plus a depot and the directed graph G induced by the set of service arcs. The optimal solution for the DPRPP on graph G is the tour for first vehicle is \(0 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 0\) and the tour for second vehicle is \(0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0\) with cost equal to 8 obtained by solving problem on graph G. The tour of vehicle one and two are shown by red and blue arcs in Figure 5. If we assume that each arc in G cannot have overlap between two vehicles’ tours, then the best solution becomes \(0 \rightarrow 4 \rightarrow 5 \rightarrow 0\) for first vehicle and \(0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0\) for second vehicle with cost equal to 5 which is it optimal for the DPRPP.
The computational experiments is conducted on a PC Intel Core 5 with 2.40 GHz processor and 4.12 GB of RAM. Also, the heuristic algorithms are coded in Matlab language. The computational results are reported in Table (1). Table (1) refers to the set of instances. For this table, the first column denotes the name of the instance while the second and the third columns refer to the number of vertices in $V$ and the number of arcs in $R$, respectively. In the fourth column (labeled Opt.) the best solution computed by the Tabu search and Genetic algorithm is reported. Columns five and six concern with the performance of the tabu search. Column five (Opt) reports the optimality solution for Tabu search. Column six (CPU time) shows the computational time. The last three columns are reported to show the impact of the Genetic algorithm procedure. The first column in this section shows the percent of improvement with respect to the solution found by the tabu search. Column eight gives the best solution by Genetic algorithm. Finally, in the last column the computational time spent by the Genetic algorithm is reported.
<table>
<thead>
<tr>
<th>Instances</th>
<th>Instances details</th>
<th>Opt. Tabu search</th>
<th>Genetic algorithm</th>
<th>CPU time</th>
<th>imp%</th>
<th>best</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPRPP-1</td>
<td>5 [V] 20 [R]</td>
<td>776</td>
<td>878</td>
<td>10.1</td>
<td>11.6</td>
<td>776</td>
<td>18.5</td>
</tr>
<tr>
<td>DPRPP-2</td>
<td>10 30</td>
<td>3510</td>
<td>3576</td>
<td>2.2</td>
<td>1.8</td>
<td>3510</td>
<td>15.2</td>
</tr>
<tr>
<td>DPRPP-3</td>
<td>15 40</td>
<td>7670</td>
<td>7710</td>
<td>3.3</td>
<td>0.51</td>
<td>7670</td>
<td>18.7</td>
</tr>
<tr>
<td>DPRPP-4</td>
<td>20 40</td>
<td>13035</td>
<td>13436</td>
<td>5.8</td>
<td>2.9</td>
<td>13035</td>
<td>30.5</td>
</tr>
<tr>
<td>DPRPP-5</td>
<td>30 50</td>
<td>28788</td>
<td>28824</td>
<td>12.3</td>
<td>0.12</td>
<td>28788</td>
<td>39.9</td>
</tr>
<tr>
<td>DPRPP-6</td>
<td>40 50</td>
<td>49730</td>
<td>49730</td>
<td>331</td>
<td>-1.01</td>
<td>50326</td>
<td>61</td>
</tr>
<tr>
<td>DPRPP-7</td>
<td>40 65</td>
<td>50388</td>
<td>50539</td>
<td>334</td>
<td>0.29</td>
<td>50388</td>
<td>59</td>
</tr>
<tr>
<td>DPRPP-8</td>
<td>50 85</td>
<td>77967</td>
<td>78227</td>
<td>150</td>
<td>0.33</td>
<td>77967</td>
<td>88</td>
</tr>
<tr>
<td>DPRPP-9</td>
<td>50 95</td>
<td>77772</td>
<td>77836</td>
<td>40</td>
<td>0.08</td>
<td>77772</td>
<td>91</td>
</tr>
<tr>
<td>DPRPP-10</td>
<td>60 75</td>
<td>111578</td>
<td>111738</td>
<td>60</td>
<td>0.14</td>
<td>111578</td>
<td>118</td>
</tr>
<tr>
<td>DPRPP-11</td>
<td>60 100</td>
<td>111331</td>
<td>111331</td>
<td>180</td>
<td>-0.14</td>
<td>111497</td>
<td>121</td>
</tr>
<tr>
<td>DPRPP-12</td>
<td>70 100</td>
<td>151188</td>
<td>151332</td>
<td>190</td>
<td>0.09</td>
<td>151188</td>
<td>178</td>
</tr>
<tr>
<td>DPRPP-13</td>
<td>70 120</td>
<td>151058</td>
<td>151058</td>
<td>200</td>
<td>-0.08</td>
<td>151188</td>
<td>195</td>
</tr>
<tr>
<td>DPRPP-14</td>
<td>80 130</td>
<td>196778</td>
<td>196819</td>
<td>400</td>
<td>0.02</td>
<td>196778</td>
<td>221</td>
</tr>
<tr>
<td>DPRPP-15</td>
<td>80 150</td>
<td>196778</td>
<td>196957</td>
<td>600</td>
<td>0.09</td>
<td>196778</td>
<td>200</td>
</tr>
<tr>
<td>DPRPP-16</td>
<td>100 160</td>
<td>305988</td>
<td>306046</td>
<td>720</td>
<td>0.01</td>
<td>305988</td>
<td>301</td>
</tr>
<tr>
<td>DPRPP-17</td>
<td>100 195</td>
<td>305977</td>
<td>306129</td>
<td>750</td>
<td>0.04</td>
<td>305977</td>
<td>305</td>
</tr>
<tr>
<td>DPRPP-18</td>
<td>130 130</td>
<td>405988</td>
<td>406088</td>
<td>900</td>
<td>0.02</td>
<td>405988</td>
<td>350</td>
</tr>
<tr>
<td>DPRPP-19</td>
<td>130 140</td>
<td>405900</td>
<td>406050</td>
<td>1000</td>
<td>0.03</td>
<td>405900</td>
<td>355</td>
</tr>
<tr>
<td>DPRPP-20</td>
<td>130 150</td>
<td>405800</td>
<td>406032</td>
<td>1200</td>
<td>0.05</td>
<td>405800</td>
<td>349</td>
</tr>
<tr>
<td>DPRPP-21</td>
<td>130 160</td>
<td>405700</td>
<td>406000</td>
<td>2000</td>
<td>0.07</td>
<td>405700</td>
<td>340</td>
</tr>
<tr>
<td>DPRPP-22</td>
<td>140 150</td>
<td>596357</td>
<td>600372</td>
<td>3000</td>
<td>0.09</td>
<td>596313</td>
<td>735</td>
</tr>
<tr>
<td>DPRPP-23</td>
<td>140 200</td>
<td>596300</td>
<td>600400</td>
<td>5000</td>
<td>0.09</td>
<td>596300</td>
<td>750</td>
</tr>
<tr>
<td>DPRPP-24</td>
<td>140 300</td>
<td>596200</td>
<td>600290</td>
<td>6000</td>
<td>0.09</td>
<td>596200</td>
<td>740</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>1213.42</td>
<td>0.71</td>
<td>292.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Discussion and conclusion

From table 1 we can conclude that when the problem size is small, the best found result for GA and TS are more or less same. When the problem size is increased (number of nodes and arcs) then the best found solution for GA and TS have different and we can see the GA is better than TS algorithm. In Figure 7 the Blue and orange lines are related to the Genetic and tabu search best solutions’ objective functions respectively. From this figure, we conclude when the problem size increase (number of nodes and arcs increase) then the best result for GA is better than TS algorithm.
From Table 1, we conclude that when problem size is increased, the time for solving the problem increase. In Figure 8 the Blue line and orange line are related to the Genetic and tabu search computational time for solving the problems. We conclude when the problem size is increased then the time for solve the problem increase and computational time for TS is much more than GA. Therefore, it can be concluded that the GA algorithm is better than TS with respect to best found solution and computational time.

**Figure 7.** Comparison between the results of TS and GA

**Figure 8.** Comparison between two algorithms with respect to computational time

The field of profitable routing problems has attracted the researchers’ attentions in recent years because of the increasing number of real applications related to the opportunities proposed by IT to logistics companies. The literature of this field is almost rich in the class of routing problems with profits on vertices, on the other hands, this is still scarce in the class of arc routing problems with profits. In this paper, arc routing problems which is called with profits Directed Profitable Rural Postman Problem (DPRPP) were analyzed. We proposed a
mathematical formulation and two meta heuristic solution approaches to tackle the problem. Computational experiments were conducted and these experiments showed that GA in the large size can find better solution in less time. Future work will be devoted to the extension of the problem and solution methods to the case of multiple vehicles, limited capacity and giving priority to customer.

References


