An Optimization Based Software Tool for Individual Automated Guideway Transit Systems

Ezzeddine Fatnassi

▶ **To cite this version:**

Ezzeddine Fatnassi. An Optimization Based Software Tool for Individual Automated Guideway Transit Systems. 15th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Sep 2016, Vilnius, Lithuania. pp.322-333, 10.1007/978-3-319-45378-1_29 . hal-01637474

HAL Id: hal-01637474

https://inria.hal.science/hal-01637474

Submitted on 17 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

An optimization based software tool for Individual automated Guideway Transit Systems

Ezzeddine Fatnassi[1]

Institut Supérieur de Gestion de Tunis
Université de Tunis
41, Rue de la Liberté - Bouchoucha - 2000 Bardo, Tunisie

Abstract. Automated Guideway Transit is a form of public transportation tools where fully driverless vehicles operate in order to move passengers between specific locations. Automated Guideway Transit includes a large variety of systems including mass transit systems and limited people automated guideway transit systems. In this paper, we focus on a specific limited people mover system called personal rapid transit. We develop in this paper an optimization based software which could manage efficiently the empty vehicles movements within the personal rapid transit system. Within this context, a branch and bound algorithm is proposed and its efficiently is proved on a set of instances taken from the literature. Our algorithm is shown to get good quality results.

Keywords: Automated guideway transit, Ondemand Transportation systems, Optimization, Branch And Bound

1 Introduction

Personal Rapid Transit(PRT) is a relatively new mode of specific transportation system. It falls under the automated guideway transit systems (AGT). In fact and as any classic AGT, PRT uses a set of automated vehicles which run on dedicated guideways. PRT is designed to move people in urban areas. PRT vehicles move people directly from origin station to destination station with a PRT'network of guideways without intermediate stops or transfers. The transportation service in PRT is done on-demand where one or a group of passengers ask for their transportation service. Typically, the main characteristics of PRT includes: i)Small electric driverless vehicles, ii) Direct origin-to-destination transportation service, iii) on demand transportation service and iv) Exclusive use of the PRT network by the PRT vehicles.

In summary, the main characteristics of PRT includes the offer of a taxi-like transportation service by the use of driverless vehicles on a set of dedicated guideways.

As the PRT is a complex, intelligent transportation service, it consists mainly on several advances components such as electric/electronic hardware, guideways, automated vehicles, stations, power sources and software[13].

Typically, a software is one of the core component of any PRT system. It includes and supervises all the other PRT'components in order to deliver a high level of mobility for its users. Central control system for PRT manages the vehicles movements, the requested trips scheduling, the response of irregular operations such as intrusion, accidents and so on. Consequently, implementing efficient softwares for PRT is of a high importance for such an intelligent transportation system. In this paper, we focus on implementing an optimization based software for PRT in order to effectively manage the empty vehicles management of PRT while reducing its total energy consumption. In fact for a on-demand transportation service such as PRT, we could get up for a high number of empty vehicles moving to take passengers from different stations in the PRT'network which represents a high level of wasted transportation capacity. Therefore including optimization module within PRT'management software is of a high importance.

In the literature, several operational optimization related studies to PRT were published such as dynamic routing [3], network design [19], simulation [5][11], optimized operational planning [6][9], energy minimization [18], total traveled distance [10][12][8], fleet size[4] and so on.

In this paper, we focus on the optimization problem related to PRT studied by Mrad and Hidri [18]. They proposed to study the routing electric vehicles with limited battery capacity in a static deterministic context. We propose a branch and bound resolution approach for that problem. We found that our branch and bound method outperforms the commercial solver Cplex' results[1] for solving the PRT'optimization problem. An specific integration of our optimization approach within a PRT'software is also proposed.

The remainder of this paper is organized as follows: Section 2 presents the formal problem definition. Section 3 develops our branch and bound optimization approach. Section 4 describes the computational results as well as the integration of our optimization approach within a PRT'software. Finally, Section 5 concludes the paper.

2 Problem formulation

We present in this section the problem formulation as presented in the work made by Mrad and Hidri[18]. The problem treated is based on the assumption that we have a predetermined list of trips to serve.

Let us suppose that we have a set of PRT stations S, a depot D, and a PRT network that makes it possible to journey between any pair of stations. Let us also suppose that we have an unlimited number of vehicles that are initially located in the depot and have battery capacity B.

$Cost_{i,j}$ is a cost matrix that defines the cost of traveling from station i to station j. The cost of moving between station i and station j in the PRT network will be the length of the shortest path calculated using a Floyd–Warshall algorithm.

[1] Details about Cplex could be found in `https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

Let us define a list of trips T that has cardinality $|T| = n$. Each trip i $(= 1, \ldots, n)$ will have its particular origin and arrival stations (OS_i, AS_i) and its particular departure and arrival times (OT_i, AT_i). We also suppose that the vehicle should return to the depot to charge its battery when necessary. The PRT problem is defined on an asymmetric graph $G = \{\boldsymbol{V}, \boldsymbol{E}\}$, where $\boldsymbol{V} = \{v_0, v_1, v_2, \dot{;} v_n\}$ is a set of nodes for which v_0 defines the depot and $v_1, v_2, \dot{;} v_n$ defines the n different trips that the PRT system must cover. We define $\boldsymbol{V}^* = \boldsymbol{V}/v_0$. Moreover, $\boldsymbol{E} = \{(v_i, v_j); v_i, v_j \in \boldsymbol{V}\}$ is a set of arcs that is defined as follows:

- If $v_i, v_j \in V^*$ with $AT_i + Cost_{(AS_i, OS_j)} \leq OT_j$, then the arc (i, j) exists and has cost c_{ij}, which represents the energy consumed from the arrival station of trip i (AS_i) to the arrival station of trip j (AS_j). Hence, each edge will have a combined cost that includes the cost of the movement from one trip to another and the cost of a trip itself.
- For each node $i \in V^*$ we add an arc $(0, i)$. The cost of this arc is c_{0i} and represents the energy used to reach the arrival station of trip i from the depot.
- For each node $i \in V^*$ we add an arc $(i, 0)$. The cost of this arc is c_{i0} and represents the energy used to reach the depot from the arrival station of trip i.

We also define $\boldsymbol{E'} = \{(v_i, v_j); v_i, v_j \in \boldsymbol{V}^*\}$.

Note that G is a direct incomplete graph, because if the arc between node v_i and node v_j exists, the opposite arc does not exist. Our problem is a typical node routing problem that can be assimilated into the asymmetrical distance-constrained vehicle routing problem (ADCVRP). The main purpose in our problem is to assign trips to vehicles with respect to the battery capacity of each vehicle in order to minimize the total consumption of electrical energy.

2.1 Flow-based mathematical formulation

In this section, we adapt and present a flow-based mathematical formulation of our problem. This formulation was presented in [16] for the asymmetrical distance-constrained vehicle routing problem. We first introduce the following integer variables:

$$x_{ij} = \begin{cases} 1 \text{ if node } j \text{ is visited after node } i, \\ 0 \text{ otherwise,} \end{cases}$$

$$z_{ij} = \begin{cases} \text{the shortest length traveled from the depot to customer } j, \\ \text{as } i \text{ is the predecessor of } j \\ 0 \text{ otherwise.} \end{cases}$$

Let c_{ij} be the cost of the journey from node i to node j. Let $\delta^+(i)$ is the set of edges that have v_i as a root. Let $\delta^-(i)$ is the set of edges that have v_i as a sink.

$$\textbf{PRT(1)}: \quad \text{Minimize} \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{1}$$

$$\sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i \in V^* \tag{2}$$

$$\sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall i \in V^* \tag{3}$$

$$\sum_{(i,j)\in E'} z_{ij} - \sum_{(i,j)\in E'} z_{ji} - \sum_{j\in V^*} c_{ij}x_{ij} = 0 \quad \forall i \in V \tag{4}$$

$$z_{ij} \leq (B - c_{j0})x_{ij} \quad \forall (i,j) \in E' \tag{5}$$

$$z_{ij} \geq (c_{ij} + c_{0i})x_{ij} \quad \forall i \neq 0, \forall (i,j) \in E' \tag{6}$$

$$z_{0i} = c_{0i}x_{0i} \quad \forall i \in E' \tag{7}$$

The objective (1) is to minimize the total charge used to make all the trips. Constraints (2) and (3) control the assignment of trips to routes. In fact, both constraints require that each node $i \in V^*$ be visited just once. Constraint (4) ensures that the distance from node v_i to any node v_j by road is equal to the distance between the depot and node v_j plus the distance from node v_j to node v_i. Constraint (5) ensures that the different routes are generated with respect to the battery capacity. In fact, it guarantees that the amount of charge the vehicle consumes to reach node v_j from the depot is less than its battery capacity minus the charge needed for returning to the depot. Moreover, in accordance with constraint (6), the total charge used to reach node v_j from the depot is greater than or equal to the charge needed to follow the direct link between the depot and node v_j. Finally, constraint (7) provides the initial value for z_{0i}, which should be equal to the distance from the depot to node v_i.

3 Branch-and-bound algorithm for static routing problem of personal rapid transit system

For the general VRP, there exists a different method of solution that utilizes an exact algorithm. The reader is referred to [2] for a more extensive review of the exact method for the VRP. In this paper, we propose a branch-and-bound algorithm for solving the PRT problem. We present first the general framework of our branch-and-bound procedure (B&B) [15][17][7]. The branch-and-bound algorithm is an exact method of finding optimal solutions for various distinctive optimization problems. This method was first introduced by Land and Doig in 1960. Its main idea consists in enumerating all possible solutions in a search tree. The branch and bound first solves a general global problem at the root of the search tree. Then, for all the other nodes the branch and bound will solve subproblems in which some constraints are relaxed and some variables are fixed.

The branch and bound normally starts from a feasible initial solution given by an upper-bounding procedure. However, if we do not have a heuristic approach, the initial upper bound is set to infinity. Once some constraints are relaxed, we need to define a search strategy that consists of the main rules for choosing the next node to branch off and treat. There are two well-known strategies for branching: the depth-first search (DFS), which consists in solving the most recently generated nodes first; and the best-first search, which consist in solving the most promising node first [14]. Our B&B is presented in Algorithm 1. Based on Algorithm 1, there are several components that need to be defined such as the lower bounding procedure and the branching rules. In the next sections, we present details of the specific components of the B&B method. We should note that a solution S in Algorithm 1 represents a set of roads that cover each node in graph G exactly once.

3.1 Computation of lower bound for PRT problem

The lower bound is one of the most important issues for the branch-and-bound algorithm. In fact, the branch and bound has to apply a lower-bounding procedure at each node of the search tree. The lower bound also has the advantage of giving a performance measure for heuristics. In the branch and bound, the lower bounds can be found by relaxing the mathematical formulation. There is a multitude of ways to relax these models. In our B&B, we decided to use the following linear model in order to generate the lower bounds.

We will first introduce the following integer variables

$$x_{ij} = \begin{cases} 1 \text{ if node of index } j \text{ is visited after node of index } i \\ 0 \text{ Otherwise} \end{cases}$$

c_{ij} define the cost of moving between nodes v_i and v_j in G.

$$\textbf{PRT}: \quad \text{Min} \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{8}$$

$$\sum_{j\in\delta^+(i)} x_{ij} = 1 \forall i \in V^* \tag{9}$$

$$\sum_{j\in\delta^-(i)} x_{ji} = 1 \forall i \in V^* \tag{10}$$

$$x_{ij} \in \{0,1\} \forall \quad (i,j) \in E \tag{11}$$

In fact to generate the lower bounds, we decided to remove the battery constraints from the mathematical model presented in [18] for solving the PRT problem. The quality of this lower bound is not high, but with this lower-bounding scheme we obtain integer solutions. Solving the relaxed mathematical model, we obtained two types of cycles (routes):

- A cycle that respects the battery capacity of the PRT vehicles.
- A cycle that does not respect the battery capacity of the PRT vehicles which must be eliminated in future resolution of the relaxed linear program.

Algorithm 1 B&B

```
Compute Lower Bound of the problem
if Obtained Solution S Is Feasible then
    Solution S is Optimal
    Return Objective value of S
else
    for all Infeasible Routes in Solution S do
        Compute the ratio for each route
        Choose the route with the largest ratio
        for all Arcs in the chosen Route do
            Relax the arc by putting its value to ∞
            if We get a feasible solution then
                Update the Upper Bound
            end if
            Add Node to the tree
        end for
    end for
    while ActualNode<MaxNodes do
        Choose Subproblem with the smallest value of objective function
        if There is more than one subproblem then
            choose one randomly
        end if
        Calculate Lower Bound
        for all Infeasible Routes in Solution S do
            Compute the ratio for each route
            Choose the route with the largest ratio
            for all Arcs in the chosen Route do
                Relax the arc by putting its value to ∞
                if We get a feasible solution then
                    Update the Upper Bound
                end if
                Add Node to the tree
            end for
        end for
        if List of infeasible nodes is empty then
            Return the upper Bound
        end if
    end while
end if
Return the upper Bound
```

3.2 Branching rules

As mentioned in the previous section, solving of the lower bound can result in cycles that do not respect the battery capacity. Such a route is infeasible and thus should be eliminated (i.e., the cycle should not appear in any future relaxation of the problem). Accordingly, we take inspiration from previous work on solving either the asymmetric traveling salesman problem (ATSP) [14] or

the ADCVRP [1]. Normally, to eliminate an infeasible route, one arc should be removed. In our branch-and-bound method, we assign a large value to the selected arc that should be removed and then solve the relaxed program again. Thus, in the future relaxations, the infeasible route will not reappear. The main disadvantage of this strategy is that the excluded edge may be present in the optimal solution, which can result in a failure of the method to find the optimal solution.

There are several ways to choose an arc to exclude from the infeasible route:

- The first way is to exclude the arc with the greatest cost.
- The second way is to exclude a random arc.
- A third way is to use the concept of the tolerance of an arc. This type of sensitivity operator will choose only one infeasible route to work on and branch from each time. Hence, for each infeasible route, the method will compute the ratio between the total electric charge used along the route and the number of arcs in that route. We then choose to work on the infeasible route with the largest ratio. For this route, we solve the different relaxation problems while eliminating one arc of the chosen route each time. We add to the tree the nodes that correspond to these different relaxations.

In this paper, we choose to work with the third way as a branching rule for our branch-and-bound algorithm.

3.3 Example

In this example, we explain the proposed B&B with the sensitivity operator. Suppose that we have 10 trips to serve and we have a battery capacity that permits the vehicle to run for 30 min. For simplicity, we assume that we have no upper bound. Therefore, the initial value of the upper bound is set to ∞. First, we solve the relaxed linear program. We obtain three routes $R1 = (D, 0); (0, 3); (3, 7); (7, D)$, $R2 = (D, 1); (1, 5); (5, 8); (8, 9); (9, D)$, and $R3 = (D, 2); (2, 4); (4, 6); (6, D)$ of 28, 45, and 31 min, respectively. Therefore, we have two infeasible routes $R2$ and $R3$. For each of these two routes we calculate the associated ratio ρ as follows:

- $$\rho(R2) = \frac{45(\text{Cost of the route R1})}{5(\text{number of arcs})} = 9 \tag{12}$$
- $$\rho(R3) = \frac{31(\text{Cost of the route R2})}{4(\text{number of arcs})} = 7.75 \tag{13}$$

The route $R2$ has the larger ratio ($\rho(R2) = 9$). Hence, we choose arcs from route $R2$ for branching. We add three nodes to the search tree, where each corresponds to solving the relaxed linear problem while excluding one arc (see Figure 1). The node counter is increased by 3. The upper bound is not updated, since we did not find a feasible solution.

We now have three subproblems on the tree, with costs 104, 107, and 105. We choose to branch from the node with the least cost (104). Solving the new

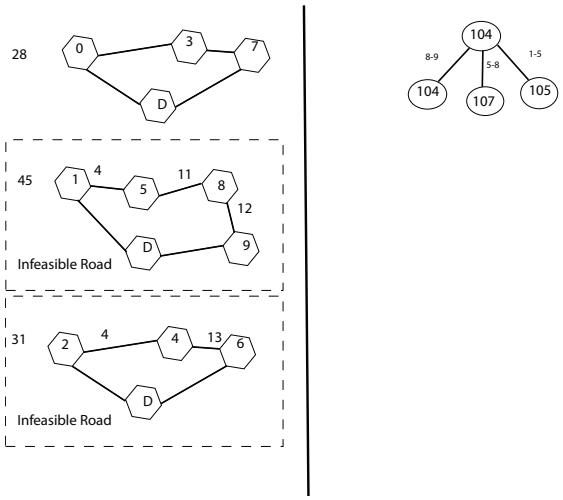

Fig. 1: First iteration of B&B

subproblem, we again have two infeasible routes $R2$ and $R3$ with costs 43 and 33. As in the previous iteration, we compute the ratio for each route:

–

$$\rho(R2) = \frac{43(\text{Cost of the route R1})}{5(\text{number of arcs})} = 8.6 \tag{14}$$

–

$$\rho(R3) = \frac{33(\text{Cost of the route R2})}{4(\text{number of arcs})} = 8.25 \tag{15}$$

Hence, we decide to branch from route $R1$. We add additional node subproblems to the tree, where each corresponds to excluding one of the arcs $(2, 4)$ $(4, 6)$, and $(6, 9)$ (see Figure 2).. The node counter is updated and increased by 3. The upper bound is not updated, since we again did not find a feasible solution.

This process is repeated until the maximum number of nodes is reached, the optimal solution is found, or the maximum running time is reached.

4 Computational Results

We have generated randomly 190 instances using the instances generator of Mrad and Hidri [18]. The size of the generated instances varies from 10 to 100 PRT'trips in steps of 5. For each trips'size, we generated 10 instances. Tests were made using a program coded in C++ and simulations are performed on a personal computer with Intel i5 2.3 GHZ processor and 6 GO of RAM(Windows 7).

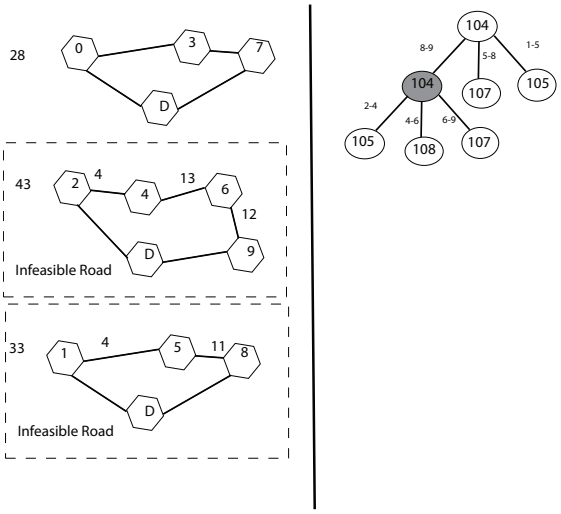

Fig. 2: Second iteration of B&B

We measure the performance of the tested algorithm by the GAP metric which is computed as follows:

$$GAP = (\frac{(SOL_{B\&B} - LB)}{LB}) * 100 \tag{16}$$

$SOL_{B\&B}$ represents the obtained solution for an instance using the $B\&B$ algorithm. The LB is the lower bound related to an instance of our problem. It is calculated based on the linear relaxation of the mathematical model given in [1]. All the mathematical models in this paper were coded using Cplex12.1.

Results of our algorithm are proposed in Table 1.

The B&B method founds an average GAP of 0.920 % in 356.717 seconds which could be described as good results. In fact and in many instances, the lower-bound technique used in our B&B can find integer solutions that are very close to the optimal solution. Also one could note from table 1 that the average computation time for B&B increases at a reasonably low rate.

We also compared the solutions obtained from B&B to those obtained by CPLEX for solving the valid mathematical model for the PRT problem [12].

Our B&B was able to find better solutions than CPLEX in 30 instances, and the results were equal in 160 instances. We also performed more extensive statistical tests to evaluate the significance of our results. In particular, we made a Wilcoxon matched-pairs signed-ranks test.

This test yielded a P-value less than 0.0001 and showed that B&B is superior to Cplex. Therefore, we can state that B&B outperforms CPLEX.

Table 1: Results of the B&B algorithm

Number of trips	Average Gap %	Average Time (s)
10	0	0.217
15	0.565	76.056
20	0.980	151.140
25	0.980	151.140
30	0.837	302.896
35	1.893	480.634
40	0.337	309.723
45	0.867	362.849
50	0.781	481.219
55	0.976	541.620
60	0.387	363.099
65	0.638	439.567
70	0.898	500.468
75	2.237	423.149
80	0.988	484.330
85	1.112	492.855
90	0.557	364.977
95	1.051	305.209
100	1.398	546.469
Average	0.920	356.717

4.1 Integration of the optimization module within a PRT software

There is several ways to integrate optimization resolution techniques within intelligent mobility softwares. In our PRT context and based on our problem, we propose a full integration procedure of our optimization technique which is presented in Algorithm 2.

Algorithm 2 PRT software

1: Get the information related to the PRT network and the List of Trips
2: Construct the matrix $Cost$ and $Time$.
3: Construct the graph G.
4: Use the B&B method to generate the optimal solution
5: The PRT control system uses the obtained data related to the different obtained roads from the B&B method to manage the PRT system.

Algorithm 2 includes several steps in order to manage efficiently the PRT system. It starts by obtaining the related input such as the metric matrix related to the cost of moving between the different PRT stations and the information related to the static list of trips to serve. Next and based on these input data, the two matrix $Cost$ and $Time$ are constructed. Then, the optimization phase of our PRT software starts by building first the graph G. The optimization phase then

,and using the proposed B&B method, would solve the constructed problem using the obtained input data. The last step of our algorithm includes the transmission of the obtained roads representing the solution of the B&B method to the central PRT control system in order to route efficiently the different vehicles available in the depot.

5 Conclusion

In this paper, we proposed an optimization approach to minimizing the energy consumption for PRT. While focusing on a static problem related to PRT, a B&B resolution approach was proposed. This method adapted to the context of PRT was capable to solve various sized PRT test instances. In fact, our algorithm was shown to get a good quality results as it handles to find an average gap of 0.920 in 356.717 seconds.

As future works, we could implement new lower bounding procedures based on different type of relaxation. A branch and cut algorithm could also be developed in order to speed up the performance of our algorithm. Also, one could note that the implementation of efficient new meta-heuristic and solution'representation seem to be very interesting in our context in order to enhance the solution quality. New implemented meta-heuristic could also be implemented within our PRT software to solve large instances'size.

References

1. Almoustafa, S., Hanafi, S., Mladenovi, N.: New exact method for large asymmetric distance-constrained vehicle routing problem. European Journal of Operational Research (2012)
2. Baldacci, R., Mingozzi, A., Roberti, R.: Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. European Journal of Operational Research 218(1), 1 – 6 (2012), http://www.sciencedirect.com/science/article/pii/S0377221711006692
3. Chebbi, O., Chaouachi, J.: Modeling on-demand transit transportation system using an agent-based approach. In: Computer Information Systems and Industrial Management, pp. 316–326. Springer (2015)
4. Chebbi, O., Chaouachi, J.: Optimal fleet sizing of personal rapid transit system. In: Computer Information Systems and Industrial Management, pp. 327–338. Springer (2015)
5. Chebbi, O., Chaouachi, J.: Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, chap. A Decentralized Management Approach for On-Demand Transit Transportation System, pp. 175–184. Springer International Publishing, Cham (2016), http://dx.doi.org/10.1007/978-3-319-29504-6_18
6. Chebbi, O., Chaouachi, J.: Reducing the wasted transportation capacity of personal rapid transit systems: An integrated model and multi-objective optimization approach. Transportation Research Part E: Logistics and Transportation Review 89, 236 – 258 (2016), http://www.sciencedirect.com/science/article/pii/S1366554515001647

7. Detienne, B., Sadykov, R., Tanaka, S.: The two-machine flowshop total completion time problem: Branch-and-bound algorithms based on network-flow formulation. European Journal of Operational Research 252(3), 750 – 760 (2016), http://www.sciencedirect.com/science/article/pii/S0377221716300224
8. Fatnassi, E., Chaouachi, J.: Vns as an upper bound for an exact method to solve a class of on-demand transit transportation systems. Electronic Notes in Discrete Mathematics 47, 101–108 (2015)
9. Fatnassi, E., Chaouachi, J., Klibi, W.: Planning and operating a shared goods and passengers on-demand rapid transit system for sustainable city-logistics. Transportation Research Part B: Methodological 81, 440–460 (2015)
10. Fatnassi, E., Chebbi, O., Chaouachi, J.: Discrete honeybee mating optimization algorithm for the routing of battery-operated automated guidance electric vehicles in personal rapid transit systems. Swarm and Evolutionary Computation 26, 35 – 49 (2016), http://www.sciencedirect.com/science/article/pii/S2210650215000619
11. Fatnassi, E., Chebbi, O., Siala, J.C.: Two strategies for real time empty vehicle redistribution for the personal rapid transit system. In: Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on. pp. 1888–1893. IEEE (2013)
12. Fatnassi, E., Chebbi, O., Siala, J.C.: Comparison of two mathematical formulations for the offline routing of personal rapid transit system vehicles. In: The International Conference on Methods and Models in Automation and Robotics (2014)
13. Furman, B., Fabian, L., Ellis, S., Muller, P., Swenson, R.: Automated transit networks (atn): A review of the state of the industry and prospects for the future. Tech. rep. (2014)
14. Germs, R., Goldengorin, B., Turkensteen, M.: Lower tolerance-based branch and bound algorithms for the {ATSP}. Computers & Operations Research 39(2), 291 – 298 (2012), http://www.sciencedirect.com/science/article/pii/S0305054811000980
15. Kadri, A.A., Kacem, I., Labadi, K.: A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems. Computers & Industrial Engineering 95, 41 – 52 (2016), http://www.sciencedirect.com/science/article/pii/S0360835216300183
16. Kara, I.: Two indexed polonomyal size formulationsfor vehicle routing problems. Technical Report. Baskent University, Ankara/Turkey (2008)
17. Lee, W.C., Wang, J.Y., Lin, M.C.: A branch-and-bound algorithm for minimizing the total weighted completion time on parallel identical machines with two competing agents. Knowledge-Based Systems 105, 68 – 82 (2016), http://www.sciencedirect.com/science/article/pii/S0950705116301010
18. Mrad, M., Hidri, L.: Optimal consumed electric energy while sequencing vehicle trips in a personal rapid transit transportation system. Computers & Industrial Engineering 79, 1–9 (2015)
19. Zheng, H., Peeta, S.: Network design for personal rapid transit under transit-oriented development. Transportation Research Part C: Emerging Technologies 55, 351–362 (2015)