# A comparison of different lower bounding procedures for the routing of automated guided vehicles in an urban context

Ezzeddine Fatnassi*

Institut Supérieur de Gestion,
University of Tunis,
41 Rue de la liberté, Le Bardo 2000, Tunisia
Email: ezzeddine.fatnassi@gmail.com
*Corresponding author

Jouhaina Chaouachi

Institut des Hautes Etudes Commerciales,
IHEC Carthage Présidence-2016 Carthage, Tunisia
Email:siala.jouhaina@gmail.com

**Abstract:** We propose different lower bounding procedures to solve the problem of routing a set of automated guided vehicles in an urban context. This problem consists of designing a set of low-cost roads starting and ending at a depot while satisfying a set of transportation demands in an urban area. The problem is treated under a set of different constraints, such as battery constraints and time window constraints. Our lower bounding approach consists of a decomposition method in which we first solve a relaxed problem. We then deal with the set of infeasible routes obtained from the relaxed problem. The different procedures developed in this paper allow us to find a set of good-quality lower bounds.

**Keywords:** automated guided vehicles; AGVs; routing; lower bound; vehicle routing problem.

**Reference** to this paper should be made as follows: Fatnassi, E. and Chaouachi, J. (2015) 'A comparison of different lower bounding procedures for the routing of automated guided vehicles in an urban context', *Int. J. Applied Nonlinear Science*, Vol. 2, Nos. 1/2, pp.120–135.

**Biographical notes:** Ezzeddine Fatnassi received his BS in Operational Research and PhD in Business Computing from the University of Tunis, Higher Institute of Management of Tunis. His research interests lie in intelligent transportation systems, evolutionary computation and operational research.

Jouhaina Chaouachi received her PhD in Computer Science from the University of Tunis. Her research interests lie in designing algorithms for optimisation problems with a special interest for Steiner tree, spanning tree problems and logistics fields.

This paper is a revised and expanded version of a paper entitled 'A lower bound for minimizing the total energy consumption in a statically routed Personal Rapid Transit system' presented at 2014 International Conference on Advanced Logistics and Transport (ICALT), 1–3 May 2014.

## 1 Introduction

### *1.1 Automated guided vehicles background*

In recent years, there has been growing interest in electric vehicular systems for transporting people. Electric vehicles have the advantage of being a clean technology, producing almost no air or noise pollution compared with combustion engines. This change affects public transportation. Consequently, we have seen a lot of new electric public transportation tools such as electric buses, electric rail transport, and personal rapid transit (PRT).

PRT is a novel mode of automated transportation that consists mainly of a set of electric automated guided vehicles (AGVs).

The main features of PRT are:

- On-demand service: Vehicles move only on passengers' request.
- Non-stop transportation service: Vehicles can move directly from the passengers' departure station to the destination station, as the stations are located off the main line.
- Privacy: Only passengers that know and choose to travel with each other can share the same vehicle. The PRT system does not allow strangers to be mixed in the same vehicle.

Automation: PRT vehicles are computer guided. Although early ideas and concepts related to PRT were developed in the '60s, only three PRT systems are operational: the Masdar PRT system in Abu Dhabi, UAE, the Heathrow PRT system at Heathrow Airport, UK, and the Morgantown PRT in West Virginia, USA. Additionally, in Uppsala, Sweden, there exists a 400 m test track that was opened in 2008 by the Vectus PRT company. In the literature, there are various PRT feasibility studies, almost all of which are favourable to PRT systems (Lichtenberg et al., 2010). However, a lack of financial and political support for the necessary technology has restricted PRT developments.

### *1.2 Related literature*

In the literature, a number of different issues have been treated using optimisation techniques. In 2010, Lie et al. (2010) considered the problem of minimising the fleet size of a PRT system. Won et al. (2006) used a multiobjective genetic algorithm to find the optimal guideway network design for a PRT system, and later presented four different optimisation problems involved in the design of a PRT system (Won et al., 2006). Lees-Miller and Wilson (2011, 2012) treated the problem of minimising passenger waiting times by presenting two different dynamic strategies.

## *1.3 Problem being considered*

As PRT transportation systems are stochastic, the routing of PRT vehicles requires dynamic methods and strategies. Hence, the assumption of deterministic and known passenger demand appears to be inconsistent. However, based on different works from the literature, the assumption of known demands is very useful for many practical cases of PRT system.

1 First, one should note that there is a lack of real PRT implementations. Therefore, PRT scientists need to use different simulation tools to validate their PRT initiatives. Consequently, the solutions obtained by specific deterministic PRT optimisation routing problems are useful for validating several PRT microsimulators (Schweizer et al., 2012). Solutions to deterministic PRT problems also help to benchmark dynamic PRT strategies.

2 Around the world, several PRT projects are in the planning stage, and thus require enhanced and specific PRT planning tools. One may wish to consider, for example, the station design and location (Won et al., 2006), maximum capacity for PRT guideways and networks, and so on. Therefore, an accurate estimation of the PRT network's maximum capacity could be made using deterministic PRT problems in order to identify any bottlenecks in the PRT network (Schweizer et al., 2012; Lees-Miller et al., 2010).

## *1.4 Focus of the paper*

Our focus in this paper is on the static problem of routing PRT vehicles. This problem was recently solved by an innovative method for routing PRT vehicles (Mrad and Hidri, 2015), and was also solved heuristically by Mrad et al. (2014). To the best of our knowledge, no study has focused on presenting a specific lower bounding procedure for this problem. In this paper, we aim to present new and different lower bounding strategies for the PRT static routing problem.

A lower bound is a value that is known to be less than or equal to the optimal value in an optimisation problem. The lower bound is said to be tight if it is close or equal to this optimal value. Generally, a lower bound is attained using a bounding procedure. For optimisation problems, lower bounds are very useful. Their primary use is to define the accuracy of the solutions obtained by non-optimal algorithms in the absence of an exact optimal value. Second, lower bounding procedures are useful in the process of implementing exact methods, such as branch-and-bound or branch-and-cut. In this paper, we present different types of lower bounds, two of which are based on the principle of partial integrality.

In fact, we aim to exploit solutions given by a linear relaxation program for our problem. More specifically, we foresee an interest in using and refining linear relaxation solutions, as they provide excellent lower bounds. We aim to add partial integrality constraints to the solutions obtained by the linear relaxation technique. By doing so, the quality of the obtained lower bound should be enhanced.

## *1.5 Contributions of this paper*

The contributions of this paper are two-fold.

1 We propose new lower bounds for the recently introduced problem of the static routing of PRT vehicles. Four bounding schemes are presented and compared. The first considers the relaxation of battery constraints for the electric PRT vehicles. The second scheme iteratively adds feasible constraints in order to obtain lower bounds. The third and fourth schemes produce lower bounds by adding partial integrality constraints to the solution obtained from the linear relaxation of two valid mathematical formulations of our problem.

2 The second contribution of this paper is to validate and compare our lower bounding principle using data from the literature through an extensive computational study and statistical tests.

### *1.6 Outline of the paper*

In this work, we focus on the static problem of routing empty PRT vehicles that are assumed to run on battery power. The remainder of this paper is organised as follows: in Section 2, we review the literature related to AGVs. Section 3 provides the definition and formulation of the PRT problem. Then, Section 4 describes the proposed lower bounds for our PRT problem. The experimental design used to test the different ideas presented in this paper, as well as an analysis of the results, is described in Section 5. Finally, our conclusions and ideas for future research in this area are provided in Section 6.

## 2 Literature review related to automated guided vehicles

As we mentioned in the previous section, PRT mainly uses a set of driverless electric vehicles. Therefore, it is interesting to study the literature related to AGVs. Generally, AGVs serve transportation demands as fast as possible and without conflict between the vehicles. Therefore, scientists have identified and studied different problems related to (Vis, 2006):

- route selection for AGVs
- scheduling of AGVs
- dispatch loads for AGVs
- dispatch of AGVs for parking places.

An AGVs working day starts when it is first assigned a transportation request. The route is then scheduled so that there is no conflict between AGVs. Finally, after serving its transportation request, a new transportation request is assigned to the AGV, or the AGV is routed to a free parking place to wait for a new assignment. The above-mentioned functions related to AGVs can be planned simultaneously or sequentially, and most of the literature about AGVs considers one or two of these problems in the same study.

Ebben et al. (2004) considered the use of AGVs in an underground transportation system. They focused on real-time control, and proposed a look-ahead heuristic and dynamic programming algorithm (Ebben et al., 2004). Wallace (2001) presented a decentralised agent-based controller system for large networks of guide paths (Wallace, 2001). Wijesoma et al. (2002) developed a stable lateral fuzzy controller (LAFC) for AGVs. Their algorithm was used to control an automated electrically powered golf car.

Dispatching problems related to AGVs have tended to focus on manufacturing areas rather than transportation. For instance, Tan and Tang (2001) presented a dispatching system for a fleet of AGVs based on a hybrid fuzzy approach, while Tiong and Randhawa (2001) developed a multi-attribute dispatching rule that uses an additive weighted method. They considered three attributes at the same time:

- the distance between an empty AGV and a workstation
- the remaining space in the input buffer of a delivery point
- the remaining space in the outgoing buffer of a workstation (Vis, 2006).

In Hall et al. (2001) considered the development of three widely used AGV dispatching strategies in the context of a manufacturing system served by AGVs. As for contexts other than manufacturing systems, scientists have paid much attention to AGVs in the context of container terminals. Chen et al. (1998) considered the development of a greedy heuristic for the assignment of AGVs to containers. Bish et al. (2001) examined the wider problem of assigning AGVs to containers and containers to storage areas in the terminal yard. Grunow et al. (2005) focused on a similar problem, but considered the option of vehicles carrying more than one container at a time. They developed a priority rule-based approach for online AGV control, and a mixed integer program (MIP) for the evaluation process. Koster et al. (2004) evaluated different real-time dispatching rules using the simulation models of three different companies. We should note that the issues related to AGV batteries have not been widely considered in the literature (Vis, 2006). This represents new and interesting feature that deserves further study.

## 3 Problem definition

In this section, we define the static problem of routing PRT vehicles. This has already been presented in Mrad and Hidri (2015), where a constraint generation technique was used to solve the problem studied in this paper. The PRT problem aims to find the optimal roads for electric PRT pods to satisfy an already-known set of passenger requests, bearing in mind the battery capacity of the PRT vehicles, time constraints, and so on. The vehicles are initially located at the depot. Each passenger request consists of origin and destination stations and a departure time. The PRT vehicles required to satisfy a given request should take the passenger(s) from the origin station at the exact departure time (no delay is permitted), and then proceed to the destination station. As the PRT pods should run on a dedicated guideway, we assume there is a network of PRT guideways, $M$ stations, and one depot. The network should guarantee the connectivity constraints. We assume that the PRT vehicles run on batteries that can only be charged at the depot. In our objective function, we must minimise the overall energy consumption of the system while satisfying all passenger requests.

We shall now develop a graph to address the above problem. Consider $n$ passenger requests. These requests can be taken as the vertices of an undirected graph $G = (V, E)$. The set of vertices also contains a node 0 that represents the depot. We also define $V^* / V / 0$. The set of edges $E$ is defined as follows:

- For each two nodes $(i, j) \in V^*$, we add an arc $(i, j)$ only if the arrival date of trip $i$ added to the time needed to go from the arrival station of trip $i$ to the departure station of trip $j$ is less than or equal to the departure time of trip $j$. This arc will have a combined time cost equal to the time cost of going from the destination station of trip $i$ to the departure station of trip $j$, plus the time cost of going from the departure station of trip $j$ to the destination station of trip $j$.
- For each node $i \in V^*$, we add an arc $(0, i)$ whose cost is the combined time cost of going from the depot to the departure station of trip $i$, plus the time cost of going from the departure station of trip $i$ to the destination station of trip $i$.
- For each node $i \in V^*$, we also add an arc $(i, 0)$ whose cost is the time cost of going from the destination station of trip $i$ to the depot.

For simplicity, we define the cost between stations as the travel time needed to go from one station to another. The battery capacity is represented as the time that the battery allows the vehicle to run. From the definition of graph $G$, note that routing vehicles under the battery constraints could be assimilated to the asymmetric distance-constrained vehicle routing problem (ADCVRP). The main difference between the two problems is the low sparsity rate of the graph $G$. In fact, if the arc $(i, j)$ exists, the opposite arc $(j, i)$ does not.

## 4 Lower bounds for the PRT problem

In this section, we present different valid lower bounds for the PRT problem proposed in this paper. These lower bounds are based on the principle of partial integrality (Yang and Lee, 2012). In this paper, we use the partial integrality principle to define a situation in which a subset of the integer decision variables retain their integrality constraints, while other subsets of integral decision variables are relaxed and transformed to a continuous domain. Generally, a lower bound that uses this technique is able to find results as good as those obtained from linear relaxation techniques. In the following, we first present two integer lower bounding strategies, then derive two partial lower bounds using different mathematical formulations.

### *4.1 First lower bound strategy*

Our first lower bound is based on the execution of Algorithm 1. This lower bound will solve the following simple linear program. We first introduce the following integer variable

**Algorithm 1** LB1 (list of trips)

```
solution ← Solve Relaxed LP(List of Trips)
Construct Roads (Roads, solution)
Final Cost ← 0
for all Roads R_i in the solution do
    if Cost(R_i) ≤ Battery Capacity then
        FeasibleRoads.Add(R_i)
    else
        InFeasibleRoads.Add(R_i)
    end if
end for
if FeasibleRoads.Size() ≠ 0 then
    Final Cost ← Cost(FeasibleRoads)
    NewListTrips.Add(Trips of the InFeasibleRoads)
    return Final Cost+LB1(NewListTrips)
else
    return Final Cost
end if
```

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

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

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

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

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

As it only concerns the assignment of trips to vehicles, this linear program will give some solutions that respect the battery constraint of the PRT vehicles, and others that do not. Our lower bounding strategy considers journeys that respect the battery capacity as part of the final lower bound solution. Then, our algorithm uses trips that include infeasible routes to construct a new problem to which the simple and relaxed linear program is applied. This process continues until the application of a relaxed linear program does not result in the use of feasible routes. Therefore, further decomposition of our problem is no longer possible. Our program will stop in this state, and return the obtained final value as the lower bound (called LB1) for our PRT problem.

**Figure 1** Basic concept of our second lower bounding procedure (see online version for colours)

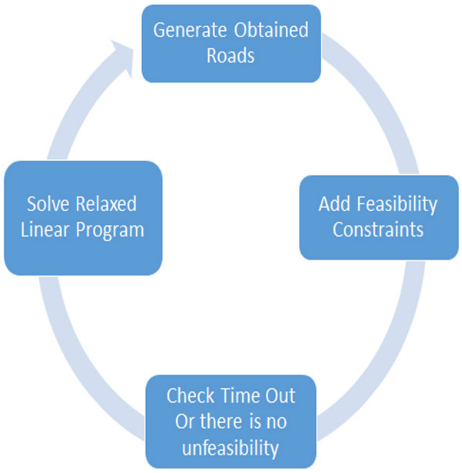
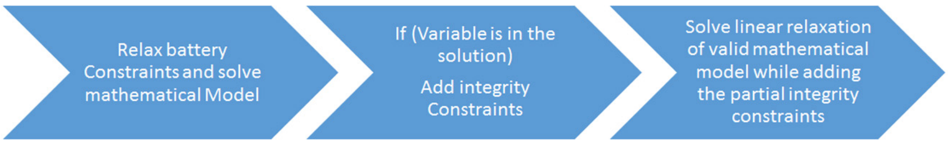



**Figure 2** Basic concept of our third and fourth lower bounding procedures (see online version for colours)




## *4.2 Second lower bound strategy*

Our second lower bounding strategy is based on the execution of Algorithm 2.

**Algorithm 2** LB2 (list of trips)

```
repeat
    solution ← Solve Relaxed LP(List Of Trips)
    Construct Roads(Roads,solution)
    Final Cost ← 0
    for all Roads R_i in the solution do
        Final Cost ← Cost(R_i)
        if Cost(R_i) > Battery Capacity then
            Add Feasibility Constraint to the relaxed linear program
        end if
    end for
until termination criterion
return Final Cost
```

Further details are shown in Figure 3. We should note that:

- A road is subset of trips connected by a set of edges that start and end at the depot, and define a specific road for the PRT vehicle.
- The function *Construct Roads* takes the solution obtained from the relaxed linear program and constructs different related roads.
- $Cost(R_i)$ defines the cost of a road ($R_i$).
- Return Final Cost defines the output of our method. This returns the cost of 14 the given solution, which defines the valid lower bound of our problem.

This lower bound will solve the relaxed linear program presented in the above section. The result of this relaxed linear program consists of a set of feasible and infeasible roads. This lower bound strategy adds feasibility constraints in order to cut infeasible roads from future iterations of the relaxed linear program. Then, our lower bounding strategy again solves the relaxed linear program while taking into account the added feasibility constraints. This process is repeated until we have no infeasible roads in the obtained solution, or until we reach a maximum time limit.

To present the feasibility constraints, we assume the following:

- $Road_{ab}$ is a sequence of selected edges that form an infeasible road from node *a* to node *b*.
- $|Road_{ab}|$ is the number of arcs in the sequence of edges $Road_{ab}$.
- $In\,fR$ is the set of all infeasible roads in *G*.

The feasibility constraints are written as follows.

$$\sum_{(i,j)\in Road_{ab}} x_{ij} \leq |Road_{ab}| - 1 \quad (1)$$

This constraint ensures that an infeasible road $Road_{ab}$ will not appear in future iterations of the linear relaxed program. Therefore, this infeasibility is cut from the search space. This tends to produce a tighter lower bound for our problem. Note that we set a maximum run time of 100 s or the ability to find a solution with no infeasibility as the termination criterion for our algorithm. This lower bound is called LB2.

### *4.3 Third principle for partial integer lower bounding strategy*

In this section, we adapt the relaxed linear program 4.1 to give tight partial integer lower bounds. In this third approach, we first solve the relaxed integer mathematical formulation 4.1. Then, we preserve the integrality of the variables that have values equal to 1. We transform all the other variables (those with values of 0) to make them continuous. After transforming the decision variables, we solve a valid mathematical formulation for our PRT problem. As in the first lower bounding strategy, we use two mathematical formulations.

The first mathematical formulation (see model 2) is derived from the work of Miller, Tucker, and Zemlin (MTZ) for the asymmetric travelling salesman problem (TSP) (Miller et al., 1960). Note that Miller et al. (1960) introduced new constraints to the TSP that eliminate any infeasible tours that do not contain the depot node. Such constraints have been applied in many works, especially in transportation problems where we have a depot node (Kara et al., 2004; Gouveia and Pires, 1999; Akgun and Tansel, 2011; Akgun and Tansel, 2010).

The lower bound that uses this particular linear relaxation is called LB3. The other lower bound, LB4, uses the linear relaxation developed by Kara for the ADCVRP (Kara, 2008) (see model 9). Note that a linear relaxation applied to an integer mathematical model relaxes the integer decision variables by allowing them to take real values.

Further details are presented in Figure 4.

More specifically, our third lower bounding procedure is based on Algorithm 3.

**Algorithm 3** LB3 (list of trips)

```
solution ← Solve Relaxed LP(List of Trips)
Construct Roads(Roads,solution)
Final Cost ← 0
for all decision variables x_ij do
    if (x_ij==1) || (x_ij==0)) then
        Add integrality constraints related to x_ij to the linear relaxation of the valid
        mathematical model of our problem
    end if
end for
Solve the linear relaxation of the valid mathematical model of our problem while
integrating the integrality constraints related to x_ij
return Final Cost of the solved model
```

We aim to retain the integrality of variables that are present in the final solution of the integer relaxation of our mathematical formulation, as this will retain the main and best features of this integer solution. In addition to the linear relaxation of the other variables, this should provide a very tight lower bound, which will be as close as possible to the optimal value.

$$\textbf{PRT}(1): \quad \text{Minimise} \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{2}$$

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

$$\sum_{j\in\delta^{-}(i)} x_{ji} = 1 \forall i \in V^{*} \tag{4}$$

$$z_i + c_{ij} \leq z_j + \left(b_i - a_j + c_{ij}\right)\left(1 - x_{ij}\right) \quad \forall (i,j) \in E^{*} \tag{5}$$

$$a_i \leq z_i \leq b_i \quad \forall i \in V^{*} \tag{6}$$

$$x_{ij} \in \{0,1\} \forall (i,j) \in E \tag{7}$$

$$z_i \geq 0 \forall i \in V^{*} \tag{8}$$

where

- $x_{ij} = 1$ if node j is visited immediately after node $i$, and $x_{ij} = 0$ otherwise
- $z_i$ is the charge used to reach node $i \in V^*$ from the depot
- $a_i$ is the energy used to reach node $i$ directly from the depot for $i \in V^*$
- $b_i$ is the remaining energy in the battery after going straight from the depot to node $i$ for $i \in V^*$.

The objective of (4.1) is to minimise the total charge used. Constraints (3) and (4) require that each node $i \in V^*$ must be present only once in the final solution, and that the solution consists of a set of roads. Constraints (5) and (6) ensure that the final solution respects the battery capacity of the vehicles.

$$\textbf{PRT(2)}: \quad \text{Minimise} \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{9}$$

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

$$\sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall i \in V^* \tag{11}$$

$$\sum_{(i,j)\in E'} z_{ij} - \sum_{(i,j)\in E'} z_{ij} - \sum_{j\in V^*} c_{ij} x_{ij} = 0 \quad \forall i \in V^* \tag{12}$$

$$z_{ij} \le \left(B - c_{j0}\right) x_{ij} \quad \forall (i,j) \in\in E^* \tag{13}$$

$$z_{ij} \ge \left(c_{ij} + c_{0i}\right) x_{ij} \quad \forall i \ne 0, \forall (i,j) \in E^* \tag{14}$$

$$z_{0i} = c_{0i} x_{0i} \quad \forall i \in V^* \tag{15}$$

where $z_{ij}$ is the shortest length travelled from the depot to node $j$ in graph $G$ after visiting node $i$. In this latter formulation, constraints (12) and (13) ensure that the final solution respects the battery constraints of the PRT vehicles.

## 5 Numerical results

The performance of the different lower bounds for our PRT system was evaluated on a large set of randomly generated instances. Algorithms were coded in C++ with Visual Studio 2008. CPLEX version 12.2 was used as an MIP solver for our mathematical models. All experiments were carried out on a PC with an Intel Core i5-2410QM CPU and 8 GB RAM, running Microsoft Windows 7. We used benchmark instances from the literature (Mrad and Hidri, 2015) to solve the PRT problem. Mrad and Hidri generated 100 instances based on five size classes, where $n$ (number of trips) $\in$ {20, 40, 60, 80, 100}. We extended their basic PRT instances by studying variations in battery capacity.

To simplify the presentation of our computational results, all columns in Table 1 have the following headings:

- $ADIS(LB_i) = \text{Max}(LB_i) - LB_i = \text{Max}(LB_i) \times 100$, where $i = 1, 2, 3, 4, 5$.
- $ERROR(LB_i) = UB - LB / UB \times 100$, where $UB = Min(UB_{CPLEX}, UB_{CG}) \cdot UB_{CPLEX})$ is the optimal output obtained by CPLEX from a valid mathematical formulation of this problem. $UB_{CG}$ denotes the results obtained by 1the constraint generation method of Mrad and Hidri for this problem (Mrad and Hidri, 2015).
- Time is the average, in seconds, needed to obtain the results.

Each value in our table represents an average within each size class.

The results for the ADIS metric are summarised in Figures 3, 4, and 5.

**Figure 3** Results of the ADIS metric for a battery capacity of 40 min (see online version for colours)

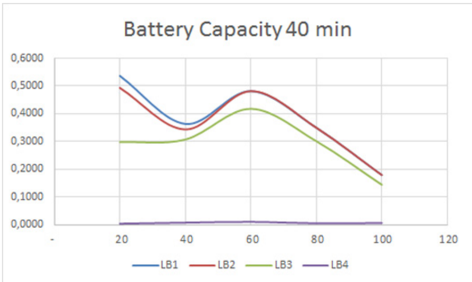



**Figure 4** Results of the ADIS metric for a battery capacity of 45 min (see online version for colours)

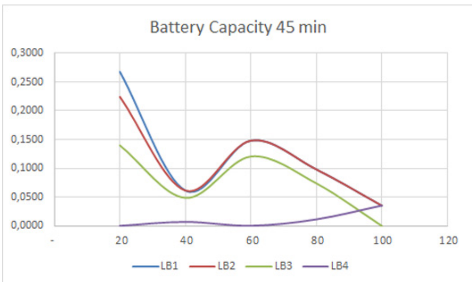



**Figure 5** Results of the ADIS metric for a battery capacity of 50 min (see online version for colours)

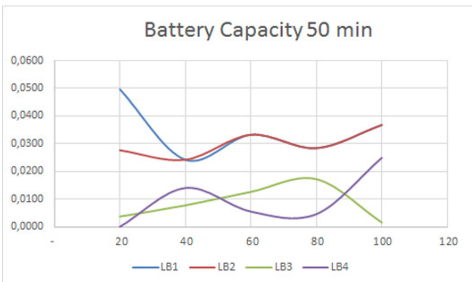


**Table 1** Computational results

| *Battery capacity* | *Problem size* | *LB1* | | | *LB2* | | | *LB3* | | | *LB4* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *ADIS %* | *Average time* | *Error %* | *ADIS %* | *Average time* | *Error %* | *ADIS %* | *Average time* | *Error %* | *ADIS %* | *Average time* | *Error %* |
| 40 | 20 | 0.5365 | 0.0516 | 1.1544 | 0.4932 | 5.1314 | 1.1096 | 0.2976 | 0.1499 | 0.8985 | 0.0037 | 0.1472 | 0.5869 |
| | 40 | 0.3632 | 0.0773 | 0.5977 | 0.3433 | 80.2247 | 0.5778 | 0.3069 | 0.2277 | 0.5406 | 0.0073 | 0.6555 | 0.2357 |
| | 60 | 0.4814 | 0.1350 | 0.5324 | 0.4814 | 90.3498 | 0.5778 | 0.4176 | 0.3913 | 0.4699 | 0.0103 | 2.1363 | 0.2695 |
| | 80 | 0.3496 | 0.2020 | 0.1509 | 0.3496 | 100.4157 | 0.1509 | 0.3010 | 0.5077 | 0.1189 | 0.0050 | 3.8684 | 0.0390 |
| | 100 | 0.1791 | 0.1719 | 0.0333 | 0.1791 | 95.5416 | 0.0333 | 0.1440 | 0.4677 | 0.0100 | 0.0061 | 2.8748 | 0.0020 |
| | Average | 0.3820 | 0.1275 | 0.4937 | 0.3693 | 74.3326 | 0.4808 | 0.2934 | 0.3489 | 0.4076 | 0.0065 | 1.9364 | 0.2266 |
| 45 | 20 | 0.2678 | 0.0516 | 1.1544 | 0.2243 | 0.1058 | 1.1096 | 0.1399 | 0.2184 | 1.0189 | 0.0000 | 0.2197 | 0.8693 |
| | 40 | 0.0613 | 0.0773 | 0.5977 | 0.0613 | 60.1840 | 0.5977 | 0.0486 | 0.2066 | 0.5847 | 0.0068 | 0.3877 | 0.5421 |
| | 60 | 0.1481 | 0.1350 | 0.5324 | 0.1481 | 80.2622 | 0.5324 | 0.1205 | 0.3102 | 0.5081 | 0.0000 | 0.7958 | 0.4499 |
| | 80 | 0.0981 | 0.2020 | 0.1509 | 0.0981 | 85.4783 | 0.1509 | 0.0738 | 0.4691 | 0.1284 | 0.0111 | 1.9837 | 0.1294 |
| | 100 | 0.0352 | 0.1719 | 0.0333 | 0.0352 | 75.5801 | 0.0333 | 0.0000 | 0.4677 | 0.0100 | 0.0352 | 2.5801 | 0.0333 |
| | Average | 0.1221 | 0.1275 | 0.4937 | 0.1134 | 60.3221 | 0.4848 | 0.0766 | 0.3344 | 0.4500 | 0.0106 | 1.193 | 0.4048 |
| 50 | 20 | 0.0496 | 0.0516 | 1.1544 | 0.0276 | 0.2026 | 1.1319 | 0.0037 | 0.2624 | 1.1055 | 0.0000 | 0.1455 | 1.1013 |
| | 40 | 0.0242 | 0.0773 | 0.5977 | 0.0242 | 45.1694 | 0.5977 | 0.0077 | 0.1761 | 0.5810 | 0.0140 | 0.2467 | 0.5874 |
| | 60 | 0.0332 | 0.1350 | 0.5324 | 0.0332 | 70.2397 | 0.5324 | 0.0126 | 0.3285 | 0.5158 | 0.0055 | 0.5712 | 0.5073 |
| | 80 | 0.0284 | 0.2020 | 0.1509 | 0.0284 | 60.4975 | 0.1509 | 0.0172 | 0.4355 | 0.1452 | 0.0046 | 0.9306 | 0.1482 |
| | 100 | 0.0368 | 0.1719 | 0.0333 | 0.0368 | 55.5293 | 0.0333 | 0.0016 | 0.4677 | 0.0100 | 0.0249 | 1.1015 | 0.0258 |
| | Average | 0.0345 | 0.1275 | 0.4937 | 0.0300 | 46.3277 | 0.4892 | 0.0086 | 0.3340 | 0.4715 | 0.0098 | 0.5991 | 0.4740 |

The results in Table 1 clearly reveal that LB4 dominates LB1, LB2, and LB3.

Indeed, LB4 exhibits better results for the two metrics on almost all problem sizes. Note also that LB4 requires more time to reach each value than the other lower bounds, except for LB2.

To better compare our different lower bounds, we performed a statistical analysis based on a non-parametric Friedman test followed by Dunn's multiple comparisons test. From Tables 2 and 3, we can see that LB4 dominates the other lower bounds. Indeed, the results obtained by LB4 are significantly different to those from LB1 and LB2. As for the comparison between LB3 and LB4, the results show that the differences are not significant. However, one could note that, on average, LB4 found better results than LB3.

**Table 2** Results of the Friedman test

| | |
|---|---|
| P-value | < 0.0001 |
| Are means signif. different? (P < 0.05) | Yes |
| Friedman statistic | 46.63 |

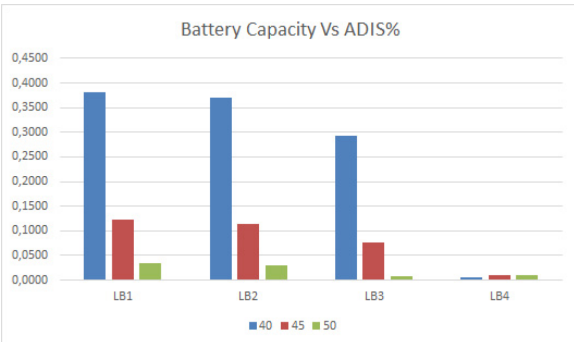
**Table 3** Results of Dunn's multiple comparisons test

| | *Rank sum diff.* | *Significant?* | *Adjusted P-value* |
|---|---|---|---|
| LB1 vs. LB4 | 43 | Yes | < 0.0001 |
| LB2 vs. LB4 | 36 | Yes | < 0.0001 |
| LB3 vs. LB4 | 9 | No | 0.736 |

The obtained results can be explained by the fact that the Kara formulation generates more constraints, and therefore requires a lot of time to find a solution. This results in a better bounding procedure, but a large computation time. The MTZ formulation uses a small number of constraints. This impacts on the quality of the solution, but reduces the time needed to find the final lower bound.

Finally, we studied the impact of the variance of the battery capacity on the results of the different lower bounding procedures. Figure 6 shows the results of this study. We can observe the impact of varying the battery capacity on the performance of each lower bounding procedure. In fact, as the battery capacity increases, the performance of the lower bounding procedures is enhanced. This is because, as the vehicles have more battery capacity, the mathematical models become much easier to solve. This results in the enhancement of the performance of our lower bounds.

**Figure 6** Effect of varying battery capacity on the ADIS metric (see online version for colours)



## 6 Conclusions

The design of efficient and powerful methods that are capable of solving vehicle routing problems has been an important subject that has attracted many scientists and researchers. It has many real-world applications, such as in logistics, transportation, and supply chain management. In this paper, we have solved a real world AGV routing application in an urban context. Our problem was related to PRT systems. Generally, the huge number of passenger requests combined with the real-time complexity of an urban context makes routing PRT AGVs a difficult task. This paper considered the static problem of routing PRT vehicles under different hard constraints. We have presented different valid lower bounding procedures for our routing problem. The main insight of this work is the combination of relaxed linear programming and partial integrality, as this allowed us to determine tight lower bounds. In fact, applying our methods to a set of PRT instances from the literature resulted in good quality lower bounds within a reasonable time. Future directions include the integration of more complex constraints to the studied problem, such as limited parking places in the station, the time needed for passengers to take a vehicle in a station, and so on.

## References

Akgun, I. and Tansel, B.C. (2010) 'Min-degree constrained minimum spanning tree problem: new formulation via Miller-Tucker-Zemlin constraints', *Computers & Operations Research*, Vol. 37, No. 1, pp.72–82.

Akgun, I. and Tansel, B.C. (2011) 'New formulations of the hop constrained minimum spanning tree problem via Miller-Tucker-Zemlin constraints', *European Journal of Operational Research*, Vol. 212, No. 2, pp.263–276.

Bish, E.K., Leong, T-Y., Li, C-L., Ng, J.W.C. and Simchi-Levi, D. (2001) 'Analysis of a new vehicle scheduling and location problem', *Naval Research Logistics (NRL)*, Vol. 48, No. 5, pp.363–385.

Chen, Y., Leong, Y.T., Ng, J.W.C., Demir, E.K., Nelson, B.L. and Simchi-Levi, D. (1998) 'Dispatching automated guided vehicles in a mega container terminal', *INFORMS*, Vol. 5, No. 2, pp.102–111, Montreal.

de Koster, R., Le-Anh, T. and van der Meer, J.R. (2004) 'Testing and classifying vehicle dispatching rules in three real-world settings', *Journal of Operations Management*, Vol. 22, No. 4, pp.369–386.

Ebben, M., van der Zee, D-J. and van der Heijden, M. (2004) 'Dynamic one-way traffic control in automated transportation systems', T*ransportation Research Part B: Methodological*, Vol. 38, No. 5, pp.441–458.

Gouveia, L. and Pires, J.M. (1999) 'The asymmetric travelling salesman problem and a reformulation of the Miller-Tucker-Zemlin constraints', *European Journal of Operational Research*, January, Vol. 112, No. 1, pp.134–146.

Grunow, M., Günther, H-O. and Lehmann, M. (2005) *Dispatching Multi-Load AGVs in Highly Automated Seaport Container Terminals*, Springer.

Hall, N.G., Sriskandarajah, C. and Ganesharajah, T. (2001) 'Operational decisions in AGV-served flowshop loops: fleet sizing and decomposition', *Annals of Operations Research*, Vol. 107, Nos. 1–4, pp.189–209.

Jeong, B.H. and Randhawa, S.U. (2001) 'A multi-attribute dispatching rule for automated guided vehicle systems', *International Journal of Production Research*, Vol. 39, No. 13, pp.2817–2832.

Kara, I. (2008) *Two Indexed Polonomyal Size Formulations for Vehicle Routing Problems*, Technical Report, Baskent University, Ankara/Turkey.

Kara, I., Laporte, G. and Bektas, T. (2004) 'A note on the lifted Miller Tucker Zemlin subtour elimination constraints for the capacitated vehicle routing problem', *European Journal of Operational Research*, Vol. 158, No. 3, pp.793–795.

Lees-Miller, J.D. and Wilson, R.E. (2011) 'Sampling for personal rapid transit empty vehicle redistribution', *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 16, No. 2216, pp.174–181.

Lees-Miller, J.D. and Wilson, R.E. (2012) 'Proactive empty vehicle redistribution for personal rapid transit and taxis', *Transportation Planning and Technology*, Vol. 35, No. 1, pp.17–30.

Lees-Miller, J.D., Hammersley, J.C. and Wilson, R.E. (2010) 'Theoretical maximum capacity as benchmark for empty vehicle redistribution in personal rapid transit', *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 46, No. 1, pp.76–83.

Li, J., Chen, Y.S., Li, H., Andreasson, I. and van Zuylen, H. (2010) 'Optimizing the fleet size of a personal rapid transit system: a case study in port of Rotterdam', *International Conference on Intelligent Transportation*, pp.301–305.

Lichtenberg, A., Guimares, P. and Podsedkowska, H. (2010) 'Planning for sustainable mobility with personal rapid transit in small European cities', in Rauch, S., Morrison, G.M. and Monzan, A. (Eds.): *Highway and Urban Environment, Volume of Alliance for Global Sustainability Bookseries*, pp.3–14, Springer Netherlands.

Miller, C.E., Tucker, A.W. and Zemlin, R.A. (1960) 'Integer programming formulation of traveling salesman problems', *J. ACM*, 27 October, Vol. 7, No. 4, pp.326–329.

Mrad, M. and Hidri, L. (2015) 'Optimal consumed electric energy while sequencing vehicle trips in a personal rapid transit transportation system', *Computers & Industrial Engineering*, Vol. 79, No. 1, pp.1–9.

Mrad, M., Chebbi, O., Labidi, M. and Louly, M.A. (2014) 'Synchronous routing for personal rapid transit pods', *Journal of Applied Mathematics*, Vol. 14, No. 1, pp.1–8.

Schweizer, J., Danesi, A., Rupi, F. and Traversi, E. (2012) 'Comparison of static vehicle flow assignment methods and microsimulations for a personal rapid transit network', *Journal of Advanced Transportation*, Vol. 46, No. 4, pp.340–350.

Tan, K.K. and Tang, K.Z. (2001) 'Vehicle dispatching system based on Taguchi tuned fuzzy rules', *European Journal of Operational Research*, Vol. 128, No. 3, pp.545–557.

Vis, I.F.A. (2006) 'Survey of research in the design and control of automated guided vehicle systems', *European Journal of Operational Research*, Vol. 170, No. 3, pp.677–709.

Wallace, A. (2001) 'Application of AI to AGV control? Agent control of AGVs', *International Journal of Production Research*, Vol. 39, No. 4, pp.709–726.

Wijesoma, W.S., Kodagoda, K.R.S. and Teoh, E.K. (2002) 'Stable fuzzy state space controller for lateral control of an AGV', *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, Vol. 32, Nos. 1–2, pp.189–201.

Won, J.M., Lee, K.M., Lee, J.S. and Karray, F. (2006) 'Guide way network design of personal rapid transit system: a multiobjective genetic algorithm approach', *2006 IEEE Congress on Evolutionary Computation*, Vols. 1–6.

Won, J-M., Choe, H. and Karray, F. (2006) 'Optimal design of personal rapid transit', *Intelligent Transportation Systems Conference*, September, pp.1489–1494.

Yang, Y. and Lee, J.M. (2012) 'A tighter cut generation strategy for acceleration of benders decomposition', *Computers & Chemical Engineering*, Vol. 44, No. 1, pp.84–93.