The Minimization of the Number of Stops Problem

Samuel Deleplanque, Ana Luisa Marques Fernandes, Luiza Bernardes Real, Ricardo Saraiva de Camargo, Alain Quilliot

THE MINIMIZATION OF THE NUMBER OF STOPS PROBLEM

S. Deleplanque[13], A.L.M. Fernandes[2], L. Bernardes Real[2], R. Saraiva de Camargo[2]and A. Quilliot[1 1]

Abstract. In this paper two optimum models to study the transport of passengers and goods through the VIPA (Automatic Individual Public Vehicle) are proposed. The objective is to minimize the number of times the vehicles stop. Since those are autonomous vehicles (they have no driver), the operations of deceleration, acceleration, stopping, opening and closing doors and going out of the circuit to go to the station implies a loss of reliability. Therefore, minimizing the number of times the vehicles stop is crucial to ensure their fleet works efficiently. Moreover, doing this we will achieve less energy expense, the minimization of the total time and the maximization of the number of served clients.

Keywords: Autonomous Vehicle, Vehicle Scheduling, ILP, Reliability

Introduction

Nowadays, studying a way of improving the transportation system is very important. For this reason, automated guided vehicles (AVGs) have been constantly developed. AVGs are able to perform their task without any intervention of a human operator. One of those types of cars, the VIPA (Automatic Individual Public Vehicle), was developed by the *Institut Pascal* research laboratory (Clermont-Ferrand, France), the design office APOJEE, and the car manufacturer Ligier in Clermont Ferrand, France.

...

[1] LIMOS, UMR CNRS 6158, Labex IMOBS3, Blaise Pascal University, Clermont-Ferrand, France.
[2] Departamento de Engenharia de Produção, Belo Horizonte, Universidade Federal de Minas Gerais, Brazil.
[3] Institut Pascal UMR CNRS 6602, Labex IMOBS3, Blaise Pascal University, France.

The partnership‘s goal was to develop a fully automatic vehicle, capable of moving over short distances (from 0.5 to 3.0 km) without any requirement for changes in infrastructure. The purpose is to allow the transport of persons or goods in a limited area, such as hospitals, airports and industrial sites. Although the VIPA is functional, its functions are limited. The vehicles do not overtake one another and the reliability‘s constraints require an evolution of the fleet on the circuit. The model must be ready to be used for fleet management problems of the VIPA in the next ten years.

This study's objective is to minimize the fleet‘s operating costs of a VIPA that is implemented to carry passengers or goods in an industrial site or in a limited area. Here, the focus is on minimizing the number of times that each vehicle stops. Several benefits might be obtained: the spending of less energy, a faster route, the minimization of the total time and the maximization of the number of served clients. However, it is important to remark that the problem might be applied to other types of technology. One example is the Private Rapid Transit which operates at London's Heathrow Airport for passengers' transport. Moreover, the problem might be used in any system that has a long service time, and that is not proportional to the size of the charge, or even in some production systems.

In this problem, it is crucial to provide power management and to optimize the consumption, predicting the vehicle‘s return to the depot. With energy management, with a faster course and with a larger number of clients served, a reliable transportation system might be achieved. In a first moment, it is assumed that the transport requests will be known in advance. Thus, the costs must be minimized, taking into account these demands, passenger safety and the reliability of transportation. In this problem, the AGV routes are assumed to be predefined regardless of the demands, hence are not subject to optimization. So we are interested in minimization of the number of the vehicle stops. Furthermore, indirectly, it also minimizes the number of vehicles requested in the fleet.

The vehicles can be requested either by the stations at the terminals or directly by the passenger' cellphones. The passengers specify the departure station and the arrival station. It is considered that the network through which the vehicle passes is fixed, and at a first time, it is a loop circuit. These networks have deviations that allow a vehicle to get out of the circuit to post a client, to get a new client or to return to the depot. These deviations, the load/unload area, are composed by stations that can be of the type "users" or "garage". The first type represents the stations in which users will be taken or left. The second shows the stations that will work as a depot where the vehicles are kept or reloaded. The profile of the speed of a vehicle in the neighborhood of a load/unload area may be represented as in Figure 1. The vehicles decelerate to make these operations outside the common track. Therefore, the more the vehicles stop, the more they lose time.

To achieve the objective of minimizing the number of times that each vehicle stops, once the fleet of the VIPA is already running, it must call a real-time algorithm to allocate a given group of clients to a group of vehicles. In other words, this algorithm must say which one is the best way to distribute a set of

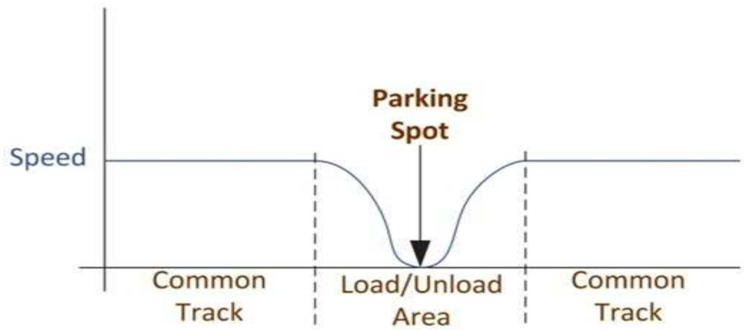

FIGURE 1. Speed Profile

requests (each one with a departure station, an arrival station and a specific load) among the available vehicles.

The structure of the paper is as follows. In Section 1, we provide a literature review. Section 2 presents the definitions and formulations of two exact models developed to reduce the number of stops in a circuit. In Section 3, the results obtained for each model are compared. Section 4 concludes and discusses future work.

1. Bibliographical Review

A serious research has been made about the subject. This is the first work that propose a minimization of the number of stops. Therefore, in the bibliographical review, a focus was done for the problems which aim at the task allocation, the path planning and the scheduling.

Having the correct AGVs to perform the tasks in the right places and times is seen as the mean problem in AVG's planning and scheduling. An approach for integrating task allocation, path planning and collision avoidance is presented. The metaheuristic algorithms of the simulated annealing and of the ant colony and an auction algorithm are investigated and applied [2].

There are some others studies that focus on the task allocation. In these cases, a rule is established to select a vehicle to perform a task. The rule can be, for example, the random vehicle rule, the nearest vehicle rule or the least utilized vehicle rule. In [3], some heuristic rules for dispatching AGVs in a job shop environment are presented. In [8], two empty vehicle dispatching policies are proposed. The first one is 'the nearest vehicle in time' whereas the second one is 'the nearest vehicle in distance' which dispatches the vehicle in the nearest position to the earliest arriving item even if this vehicle is not available before any other vehicle in the system as long as the increase of the waiting time of the item is less than a certain value.

Another part of the problem is the path planning, which is the selection of the path for the AVGs after that the tasks have been assigned and before that the movements starts. In [5], exact and approximate methods are investigated for estimating the minimization of vehicular movements in road network models where link speeds vary over time. A condition of FIFO is assumed and several adaptations of Dijkstra's algorithm are used for path-finding purposes. In [1] a genetic algorithm (GA) is applied to the basic vehicle routing problem (VRP). In this problem, the known demands are supplied by a single depot, vehicles are subject to a weight capacity and, in some cases, to a limit on the distance that they can travel. A GA is also proposed in [7]. This study treats the problem of simultaneous scheduling of machines and of AGVs in a flexible manufacturing system aiming at minimizing the makespan. Zhan and Noon evaluate 15 shortest path algorithms using a variety of real road networks [9].

Another point that has been studied in AVGs systems is the scheduling, which determines arrival and departure times of the vehicles at each path segment, station and intersection. The simulation methodology is used to compare the performance of three AGV configurations under a variety of experimental conditions in flexible manufacturing systems aiming to reduce congestion [4]. The objective of [6] is to develop an efficient deadlock prediction and avoidance algorithm for AVGs systems. An algorithm is developed to predict the occurrence of cyclic deadlock and the other kinds of deadlocks are avoided by using control logic.

2. Definitions and Formulation

To solve the problem two models were developed. The first one considers that the stations are disposed in a horizontal line. In contrast, the second one considers that the stations are disposed in a circular circuit, which allows the vehicles to make several cycles to meet the requests. Nevertheless, both models are utilized in transportation systems disposing of a circular circuit.

2.1. Model in a line circuit

In this model, the circuit by which the vehicle moves has been stretched and it can be traversed only once after optimization. That means we have a new node each time the vehicle passes through a station. Then, if the circuit has N stations, after stretching, it will have $2N-1$ nodes. A circuit with four stations can be represented by the figure 2.

We consider a line in which $2N-1$ stations are arranged, a set D of requests d and a set of K homogeneous vehicles whose capacity is γ. Each request d consists of a 3-tuple $o(d)$, $e(d)$ and $w(d)$ where $o()$ and $e()$ are the origin and the destination of d, respectively, and $w()$ is the number of persons in the request d or its charge.

We also defined the variable $x_i^k \in \{0,1\}$ which assumes the value 1 if the vehicle k stops in the node i; the variable $y_d^k \in \{0,1\}$ which assumes 1 if the vehicle k serves the request d; and the variable z_i^k that represents the charge of the vehicle k when it leaves the node i.

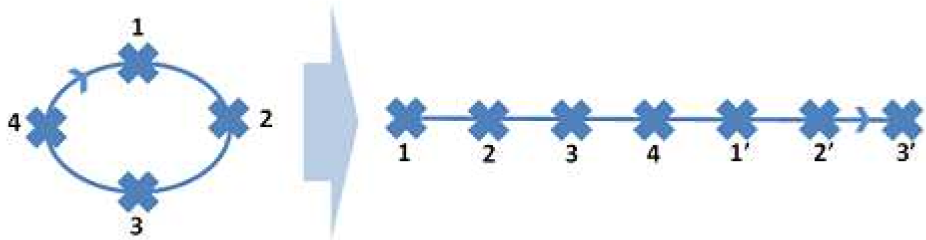

Figure 2. Linear circuit after been stretched

$$\min \sum_{k\in K} \sum_{i=1}^{2N-1} x_i^k \tag{1}$$

s.t.

$$\sum_{k\in K} y_d^k = 1 \qquad \forall\ d \in D \tag{2}$$

$$y_d^k \leq x_{o(d)}^k \qquad \forall\ d \in D,\ k \in K \tag{3}$$

$$y_d^k \leq x_{e(d)}^k \qquad \forall\ d \in D,\ k \in K \tag{4}$$

$$z_i^k = z_{i-1}^k + \sum_{\substack{d\in D:\\ o(d)=i}} w_d y_d^k - \sum_{\substack{d\in D:\\ e(d)=i}} w_d y_d^k \quad \forall\ i = 1..2N-1\ k \in K \tag{5}$$

$$z_i^k \leq \gamma \quad \forall\ i = 1..2N-1,\ k \in K \tag{6}$$

The objective is to minimize the number of times the vehicles stops. Constraints (2) assure that every request is served. Constraints (3) ensure that if the vehicle k serves the request d, whose origin is i, then the vehicle k stops at the node i. Constraints (4) ensure that if the vehicle k serves the request d, whose destination is i, then the vehicle k stops at the node i. Constraints (5) make the mass balance. The charge of a vehicle when it leaves a node i is always its charge when it left the node before i, plus the charge that it picked up in i, less the charge that it left in i. Finally, constraints (6) assure that the capacity of each vehicle is always respected.

As in this model the vehicles do not have the possibility of doing several cycles in only one optimization, they cannot always satisfy all the requests. In the case where the aggregated capacity of all vehicles is not enough to answer all the set of requests this problem will be infeasible. Thereby, aiming to construct a model which will always be feasible, the next model was developed.

2.2. Model with the circle circuit

In this new model, in order to meet all the requests, each vehicle can make a maximum of C cycles, where $C = \lceil \sum_{d\in D} w_d/K\gamma \rceil + 1$. We can assume that this number of cycles will always be enough for a certain number of vehicles and for a certain set of requests, each one with a different charge.

Now, we shall consider a circle in which N stations are arranged. The others parameters still being the same.

We defined the variable $x_{ic}^k \in \{0,1\}$ which assumes the value 1 if the vehicle k stops in the node i in the cycle c; the variable $y_{dc}^k \in \{0,1\}$ which assumes 1 if the vehicle k serves the request d in the cycle c; and the variable z_{ic}^k that represents the charge of the vehicle k when it leaves the node i in the cycle c.

The problem formulation is presented below.

$$\min \sum_{k\in K}\sum_{i=1}^{N}\sum_{c=1}^{C}(1+c/C)x_{i,c}^k \tag{7}$$

s.t.

$$\sum_{k\in K}\sum_{c=1}^{C} y_{d,c}^k = 1 \qquad \forall\ d\in D \tag{8}$$

$$\sum_{k\in K} y_{d,C}^k = 0 \quad \forall\ d\in D: e(d)<o(d) \tag{9}$$

$$y_{d,c}^k \le x_{o(d),c}^k \qquad \forall\ d\in D,\ c=1..C,\ k\in K \tag{10}$$

$$y_{d,c}^k \le x_{e(d),c}^k \qquad \forall\ d\in D: e(d)>o(d),\ c=1..C,\ k\in K \tag{11}$$

$$y_{d,c}^k \le x_{e(d),c+1}^k \quad \forall\ d\in D: e(d)<o(d),\ c=1..C-1,\ k\in K \tag{12}$$

$$z_{1,1}^k = \sum_{d\in D:o(d)=1} w_d y_{d,1}^k \qquad \forall k\in K \tag{13}$$

$$z_{i,1}^k = z_{i-1,1}^k + \sum_{\substack{d\in D:\\ o(d)=i}} w_d y_{d,1}^k - \sum_{\substack{d\in D:\\ e(d)=i,\\ e(d)>o(d)}} w_d y_{d,1}^k \quad \forall\ i=2..N,\ k\in K \tag{14}$$

$$z_{1,c}^k = z_{N,c-1}^k + \sum_{\substack{d\in D:\\ o(d)=1}} w_d y_{d,c}^k - \sum_{\substack{d\in D:\\ e(d)=1,\\ e(d)<o(d)}} w_d y_{d,c-1}^k \quad \forall\ c=2..C,\ k\in K \tag{15}$$

$$z_{i,c}^k = z_{i-1,c}^k + \sum_{\substack{d\in D:\\ o(d)=i}} w_d y_{d,c}^k - \sum_{\substack{d\in D:\\ e(d)=i,\\ e(d)>o(d)}} w_d y_{d,c}^k$$

$$- \sum_{\substack{d \in D: \\ e(d)=i, \\ e(d)<o(d)}} w_d y^k_{d,c-1} \quad \forall\ i = 2..N, c = 2..C,\ k \in K \tag{16}$$

$$z^k_{i,c} \leq \gamma \quad \forall\ i = 1..N,\ c = 1..C,\ k \in K \tag{17}$$

The objective is to minimize the number of times the vehicles stop. In order to guarantee that the requests will be answered as soon as possible, we multiply $x^k_{i,c}$ by c. Constraints (8) assure that every request must be served exactly once. Constraints (9) do not allow a request to be served in the last cycle if $e(d) < o(d)$, because otherwise this request would not be delivered, once this kind of request is always served in a cycle c and delivered in a cycle $c + 1$. Constraints (3), (11) and (12) guarantee that the vehicle k stops in the node i in the cycle c in the cases where: i is the origin of the request d and k serves d in the cycle c; i is the destination of d, with $e(d) > o(d)$, and k serves d in the cycle c; i is the destination of d, with $e(d) < o(d)$, and k serves d in the cycle $c - 1$. Constraints (13), (14), (15) and (16) make the mass balance. The charge of a vehicle when it leaves a node i is always its charge when it leaved the node before i, plus the charge that it picked in i, less the charge that it left in i. Finally, constraint (17) assures that the capacity of each vehicle is always respected.

Aiming to find a lower bound to the problem, the constraint (18) was added to it. We can infer that the number of stops made in a station will be at least the maximum between the biggest integer of the division of the number of persons, who have this station as origin by the capacity of the vehicles and the biggest integer of the division of the number of the persons, who have this station as destination by the capacity of the vehicles.

$$\sum_{k \in K} \sum_{c=1}^{C} x^k_{ic} \geq max(\lceil \sum_{d \in D: o(d)=i} w_d/\gamma \rceil, \lceil \sum_{d \in D: e(d)=i} w_d/\gamma \rceil) \forall\ i = 1..N \tag{18}$$

3. Computational Analysis

We present here the tests performed and their results, in relation to the two models, on an Intel I5, 2.5GHz and with 6Gb of RAM. Since there has been no previous test for this problem, we ran experiments on our own data, generated as realistically as possible. Sixteen scenarios were tested. For each scenario, five different instances were randomly generated. The difference from one instance to another is the origin and the destination of its requests. A program in C++ was created which takes as input the number of stations and the number of requests, to randomly generate the origins and destinations for each instance. The number of vehicles K ranged between 2 and 4 and their capacities assumed values 3, 4 and 6. The number of stations N was either 5 or 10. The number of requests $|D|$ was either 20, 30 or 40. All requests were considered to have one as their load. To perform these tests, the exact models were developed in AMPL and tested in CPLEX.

The table 1 compares the results obtained for each model and the computational time spent. S_{line} is the number of stops obtained by CPLEX on the model in a line circuit, cpu_{line} and $\text{cpu}_{circuit}$ denote the execution times which where necessary for solving each problem, f is the objective function (7) and Cycles is the number of cycles which was necessary to satisfy all the requests. The values below are an average of the instances of each scenario. For the second model, the objective function and the number of cycles utilized were also noted. This is due to the fact that the calculation of the objective function is different for the two models and that the second model allows the vehicles to do more than just one cycle. What matters here is the number of total vehicle stops.

Table 1. CPLEX results for both linear programs

Inst.	K	N	$\|D\|$	γ	S_{line}	cpu_{line}	$S_{circuit}$	f	Cycles	$\text{cpu}_{circuit}$
1	3	5	20	6	11.6	0	9.2	14.3	2.4	1
2	2	5	20	4	-	0	11.2	15.8	2.4	1
3	3	10	20	6	17	0	14.2	21.9	2.2	1
4	2	10	20	4	-	0	15.4	22	2.4	1
5	4	5	20	6	11.6	0	10.6	17.9	2	0
6	2	5	20	3	-	0	12.6	17.2	2.4	13
7	4	10	20	6	17	0	15.8	26.7	2	0
8	2	10	20	3	-	0	16.6	22.5	2.8	4
9	4	5	30	6	13.8	1	11.8	18	2.2	5
10	2	5	30	6	-	0	11.6	16.3	2.2	2
11	4	10	30	6	23.2	1	19	29.1	2.8	20
12	2	10	30	6	-	0	18.6	26.4	2.8	11
13	4	5	40	6	13.2	1	13.2	20.7	2.4	74
14	2	5	40	6	-	0	12.6	17.7	3	65
15	4	10	40	6	26.4	3	22.4	27.1	2.4	730
16	2	10	40	6	-	0	17.4	23.7	2.2	561

As stated earlier, the second model developed obtains presents feasible solutions even for instances for which the first model finds no answer. This happens because while the second model allows the vehicle to make a number of cycles that makes possible the fulfillment of all demands, the first model allows each vehicle to make just one cycle.

Another issue to note is that in some situations the second model finds a better solution than the first one. It can be understood by this example: if we have 2 vehicles with capacity 1, and 2 requests (one wants to go from the station 1 to the station 4 and the other wants to go from the station 2 to station 1). The first model finds the solution in which we utilize the two vehicles and each vehicle stops twice. Then, we have a total of four stops. If we utilize the second model, just one car is enough to answer the two requests and we have a total of three stops (the vehicle will first answer the second request, and subsequently the first one).

For scenarios 15 and 16, the second model spent a long time to solve some of the instances. Therefore, we stipulated a maximum time of 20 minutes to allow the model to function. The limited time was determined because we assumed that more than 20 minutes is not viable to await a solution. Then, the averages for these two scenarios shown above contemplate some solutions that are not the optimal ones. We accounted the solutions the model found after 20 minutes running. It is crucial to emphasize that the first model found the best solution for all the five instances of scenario 15. We can notice that when the first model finds viable solutions, it spends much less time than the second model.

Finally, analyzing the results, we can conclude that the number of stations in the circuit is more important to determine the number of vehicle stops than the number of requests.

4. Conclusion and future work

The VIPA is an important new technology to improve the transportation systems in cities and in industrial cities. Increasing its reliability is crucial, given its constraints due to the fact it is an autonomous vehicle. Therefore two exact models were developed aiming to minimize the number of times the fleet stops. The first model is faster than the second one, however it does not always find a feasible solution, because it limits to just one the number of cycles a vehicle can make. On the other hand, the second model allows the vehicles to make as many cycles as necessary, always finding a viable solution. Nevertheless, this last model can take too much time to find the best solution for larger instances such as those with 40 requests and 10 stations. Hence, a future work will implement a heuristic method to solve the problem in as little time as possible. This heuristic might be an insertion method (the demands are considered one after the other) or a successive matching method (the origin nodes are scanned and the related demands are distributed among the vehicles).

Another model for the variant of the problem involving time windows has to be implemented. The use of the time windows allows us to deal with real service times, which have influence on the duration of the tours which are performed by the vehicles. It requires dealing with another decisional variable.

Acknowledgments

We wish to thank the *Conseil Regional d'Auvergne*, the FEDER of the European Union.

References

[1] Baker, B. M., Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. Computers and Operations Research, 30(5), 787-800.

[2] Liu, D. K., Kulatunga, A. K. (2007). Simultaneous planning and scheduling for multi-autonomous vehicles. In Evolutionary Scheduling. Springer Berlin Heidelberg. 437-464.

[3] Egbelu PJ, Tanchoco JMA (1984) Characterization of automatic guided vehicle dispatching rules. International Journal of Production Research 22: 359-374.

[4] Farling BE, Mosier CT, Mahmoodi F (2001) Analysis of automated guided vehicle configurations in flexible manufacturing systems. International Journal of Production Research, 39: 4222-4239.

[5] Horn, M. E. (2000). Efficient modeling of travel in networks with timevarying link speeds. Networks, 36(2), 80-90.

[6] Moorthy RL, Hock-Guan W (2000) Deadlock prediction and avoidance in an AGV system. Master of science, Sri Ramakrishna Engineering College, National University of Singapore.

[7] Ulusoy GU, Sivrikaya SF, Bilge U (1997) A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Computers and Operations Research, 24: 335-351.

[8] Yamashita H (2001) Analysis of Dispatching Rules of AGV Systems with Multiple Vehicles. IIE Transactions 33: 889-895.

[9] Zhan FB, Noon CE (1996) Shortest Path Algorithms: An Evaluation using Real Road. Transportation Science, 32: 65-73.

[10] Vis, I. F. (2006). Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research, 170(3), 677-709.

[11] Le-Anh, T., De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. European Journal of Operational Research, 171(1), 1-23.