Order Sequencing in an Automated Warehouse System

Kai Gutenschwager[1,2], Sven Spieckermann[1,2] and Stefan Voß[1]

[1]**Technische Universität Braunschweig,**
Institut für Wirtschaftswissenschaften,
Abt-Jerusalem-Straße 7,
D - 38106 Braunschweig, Germany

[2]**SimPlan Gesellschaft für Simulation betrieblicher Abläufe mbH,**
Siemensstraße 10 - 12,
D - 63165 Mühlheim am Main, Germany

Abstract

In this paper we present part of a simulation study in the field of automated warehouse management. The aim of the study was to provide results about the actual processing time for the daily work load supposed to be completed in an eight hour shift. The study focussed on the order sequencing problem having major influence on the results obtained. The problem was divided into two sub-problems, one being the vehicle routing problem with time windows, and tackled using tabu search. Additionally, the integration of the model into a decision support system is proposed.

Keywords
simulation, tabu search, warehouse systems, order sequencing problem, vehicle routing problem with time windows

1 Introduction

The intelligent management of warehouse systems has become a major factor with respect to competitiveness and customer service in a variety of industrial branches. We present a part of a simulation study in the field of warehouse management. The simulation study was carried out for a German supplier of the automotive industry. Within a more general project of restructuring the logistic services, one specific topic was to extend the capacities of the existing warehouse at the main site of the company. The simulation model was developed using SIMPLE++, an object-oriented simulation package. Some of the features of this package are given by Levasseur [4].

The automated warehouse under consideration is supposed to store about 50,000 different articles. The daily work load is about 8700 articles that are retrieved in containers from 25 aisles. About 10,000 containers need to be handled within the daily work shift of eight hours. Customer orders are usually known at least one day ahead and consist of different articles to be picked from one to over one hundred (with an average of 4.5) locations in the warehouse. As each article is stored in only one location it is most likely that not all containers belonging to an order will be found in one aisle. Each aisle is served by a storage and retrieval unit (SRU), that can store up to 40 containers. All picked articles are transported on a conveyor system to the shipping area, where articles belonging to one order are brought together in special boxes.

The main emphasis of the simulation study was to provide results about the actual processing time for the daily work load, i.e., for the time it takes to complete the entire orders of a day. The objective was to minimize the actual processing time, i.e. the makespan. Given that the makespan exceeds the work shift this results in respective overtimes for the employees. The processing time depends on the performance of the overall system which is determined by the time the SRUs need to retrieve the containers from the aisles and the time the employees in the shipping area need to get the commissions ready, i.e., to gather the articles of the respective orders and to put them in the respective boxes.

A significant and quite characteristic restriction to be considered within the shipping area is that a customer order must not be interrupted on a so-called *packing table* by another order, which means that an order occupies the packing table until the last container belonging to the customer order has been handled. If customer orders have been started by the SRUs, but no packing tables are available the respective containers have to be stored in a buffer. Thus, the so-called order spread, i.e., the time between the delivery of the first and the last container of a customer order in the shipping area, has a significant impact on the utilization of the packing tables and on the total processing time. As the average order-spread depends on the schedules of the SRUs these schedules have to be determined considering both an acceptable (or scheduled) order-spread and the retrieval times of the SRUs. Accordingly, with respect to the overall objective of minimizing the total processing time we are facing two scheduling problems, the derivation of acceptable order-spreads, in terms of consequences for the utilization of the packing tables, and the minimization of the SRUs retrieval times.

In the following we present a two step algorithm to compute schedules for the SRUs. We first calculate acceptable order-spreads and use the results for defining the restrictions of the second problem, a well known operations research problem, the vehicle routing problem with time windows (VRPTW, cf. Savelsbergh [8]). This approach is promising as research has provided a variety of algorithms to provide satisfying solutions for instances of the VRPTW, and these can be applied directly. Therefore, we focus on presenting an algorithm for tackling the first problem. As a result of the two algorithmic steps we obtain a schedule for the SRUs that might be considered as an initial solution. (Note, that this solution must not necessarily be feasible in terms of definitely avoiding overtimes.)

Consequently, an algorithm to improve the initial solution has to be provided. This improvement procedure consists of first analyzing the determined schedule by performing a simulation experiment. Depending on the achieved order-spread in the experiment the results are then used to either “tighten” or “loosen” certain restrictions and to tackle the concluding instance of the VRPTW anew. Additionally, a meta heuristic, which is based on the tabu search paradigm, is applied to guide the search process.

In the sequel we describe relevant parts of the simulation model and define the scheduling problem in more detail, showing the connection to the vehicle routing problem. Afterwards, in Section 3, algorithms are presented to provide solutions for the outlined problem. In a concluding section an outlook explains opportunities of integrating the simulation model in a decision support system.

2 Problem definition

2.1 Minimizing the retrieval time of the storage and retrieval units

The retrieval time of a SRU for a set of containers depends on the traveling distance the unit needs to visit all locations of the containers in the set. Thus, the problem of minimizing the retrieval time can be considered as a problem of scheduling the sequence of locations to be approached by the SRU in the respective aisle. One way to determine this sequence of locations is to define tours, on which each SRU has to collect the ordered items and return them at the head of the aisle, with respect to traveling distance only. Minimizing the time needed to perform a specified amount of retrievals can be understood as an application of the capacitated vehicle routing problem (VRP). The VRP can be defined as determining an assignment of vehicles to customers that require (or provide) a certain amount of some commodities, such that the capacity of each vehicle is not exceeded and the total travel distance is minimized. An introduction to the VRP is given by Laporte [3].

This version of the VRP may be adopted to our problem assuming that more than one SRU serves one aisle. If we set the number of available vehicles for a single aisle equal to the number of locations (customers) to be served by the SRU, all possible assignments of locations to tours can still be chosen. We only have to perform the resulting tours successively (because we actually have only one SRU per aisle), which stands in contrast to other applications of the VRP as, e.g., the delivery of mail or the collection of garbage [8], where all vehicles start at the same time. In Figure 1 an aisle is shown, with all locations from where articles are to be retrieved illustrated as squares. If the capacity of the SRU equals four, all containers can be retrieved within three tours. In the given example, the chosen tours are illustrated by the connecting lines between the locations. As the tours are performed successively the time to retrieve all containers is equal to the sum of the length of all single tours.

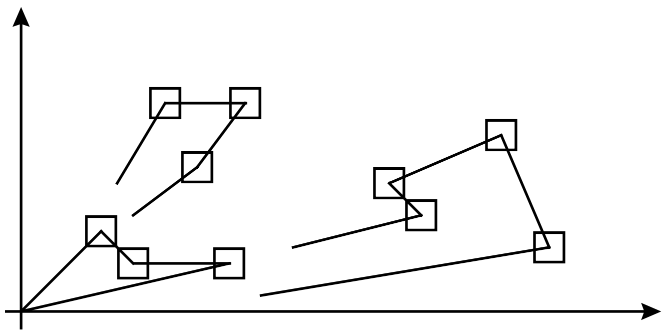

Figure 1. Sequence considering the traveling distance alone

If we define the problem of retrieving all ordered containers as a VRP, we minimize the retrieval time of the SRU. This policy leads to an optimal schedule, if no further restrictions, e.g., space limitations concerning started orders or deadlines for orders, have to be taken into account. Introducing restrictions concerning deadlines for customer orders lead to a special case of the VRP, the vehicle routing problem with time windows (VRPTW). In addition to the VRP sketched above all customers are to be visited within a customer dependent time window.

Mathematical formulations of the VRPTW can be found in [1, 3]. The time window for a customer i is defined by two points of time $tw_b(i)$ and $tw_e(i)$, with $tw_b(i) < tw_e(i)$. If, e.g., a deadline for visiting customer i is to be modeled we can define the respective time window as $tw_b(i)$ equal to the beginning of the planning horizon and $tw_e(i)$ equal to the due date.

In the previous section we have introduced an acceptable (or scheduled) order spread, which strongly relates to the efficiency of the shipping area. In the next sub-section we present the problem of obtaining an acceptable order spread. The resulting schedules are used to introduce time windows for the retrieval of containers belonging to customer orders. We can then formulate one instance of the VRPTW per aisle and minimize the traveling times of the SRU with respect to the restrictions concerning the order spread.

2.2 Minimizing the order spread

In the preceding section we have given a problem formulation for minimizing the retrieval times of the SRUs ignoring the time windows set up through an acceptable order spread. As containers of a large amount of different orders are retrieved simultaneously and the number of packing tables is restricted, a large number of containers has to be buffered before space is available at the packing tables again, and the total processing time is increased due to waiting times for still incomplete orders at the packing tables. This was an early finding of the simulation study.

Therefore, another problem has to be considered before solving the VRP as described in Section 2.1. The focus now is on synchronizing the retrieval of containers belonging to a specific customer order from different aisles. Orders spread over more than one aisle are called *critical*. A *sub-order* is defined as all containers belonging to the same customer order that need to be picked in the same aisle. Whereas for the minimization of the traveling distance the objective is to minimize the length of the tours the SRUs have to perform, we do now assume a given (average) tour length for each SRU. Furthermore we assume that the capacity of each SRU is used up completely on each tour. Let $t_{arr}^{first}(i)$ be the minimal point of time that a tour ends which retrieves containers belonging to order i, and $t_{arr}^{last}(i)$ the maximal point of time that a tour ends which retrieves containers belonging to order i. The order spread of order i is defined as $t_{arr}^{last}(i)$-$t_{arr}^{first}(i)$.

In Figure 2 an example is given for four aisles. Tours are indicated as rectangles, with the capacity illustrated by number of sub-rectangles, in which the order number of the assigned container is stated. In aisle 1 the capacity of the SRU is equal to three and order 1 is served on the first tour. The end of this tour defines $t_{arr}^{first}(1)$, while the third tour in aisle 2 defines $t_{arr}^{last}(1)$. The objective is then to minimize the average order spread of q orders. Formally, this may be stated as:

$$Min \quad \sum_{i=1}^{q}(t_{arr}^{last}(i) - t_{arr}^{first}(i)) \qquad (1)$$

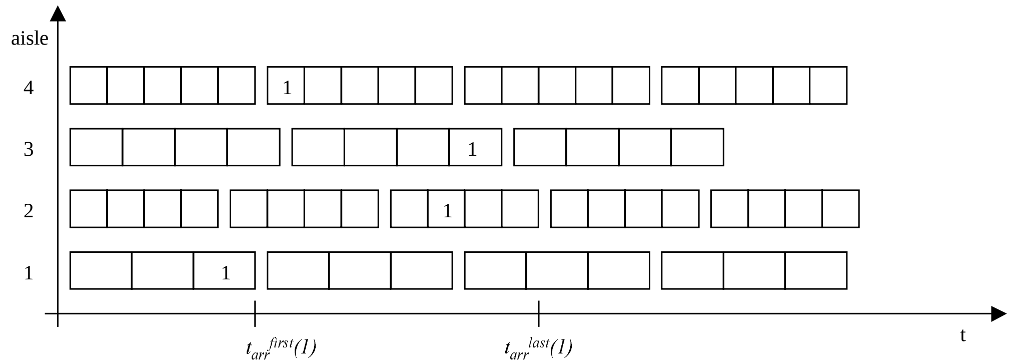

Figure 2. Example: Order sequence as scheduled tours

3 Computing an initial solution

In the last section we have introduced two separate problems that both appear relevant for the objective function value of the overall problem, minimizing the total processing time. In this section we present a two-step algorithm to compute a schedule for all SRUs. The influence of the schedules on the overall system, i.e. the warehouse, are analyzed by executing a simulation experiment.

In the first step the objective is to minimize the order spread for critical orders. The resulting schedule is applied to define the time windows for instances of the VRPTW, such that the time windows reflect the acceptable order spread computed in this initial part of the solution process. We focus on the first step of the algorithm, as the second step mainly consists of applying adaptations of known heuristics to instances of the VRPTW. Recent surveys on modern heuristic approaches for the VRPTW are given by Potvin and Bengio [6] and Potvin et al. [7].

In the first step, the lengths of the tours are assumed to be fixed. The maximal amount of tours is computed for each SRU, such that the last tour ends before the maximal estimated processing time of the SRUs. The result is a time scheme as illustrated in Figure 3.

The orders are sorted according to the size of the largest sub-order. Beginning with the customer order containing the largest sub-order, the orders are planned successively. Scheduling all sub-orders directly after the sequence of containers assigned (for the aisle) so far, defines $t_{arr}^{last}(i)$ for the customer order i. The containers are then scheduled as late as possible without violating the assumed threshold t_{arr}^{last}. Tours, to which containers are assigned to, are called *open*. All preceding tours of open tours within an aisle still to be filled with containers are also set *open*, even if no containers are assigned yet. In Figure 3 an example is given. The orders are numbered according to the largest sub-order. Order 1 is scheduled being served by two tours in aisle 1. The resulting open tours are indicated as gray rectangles.

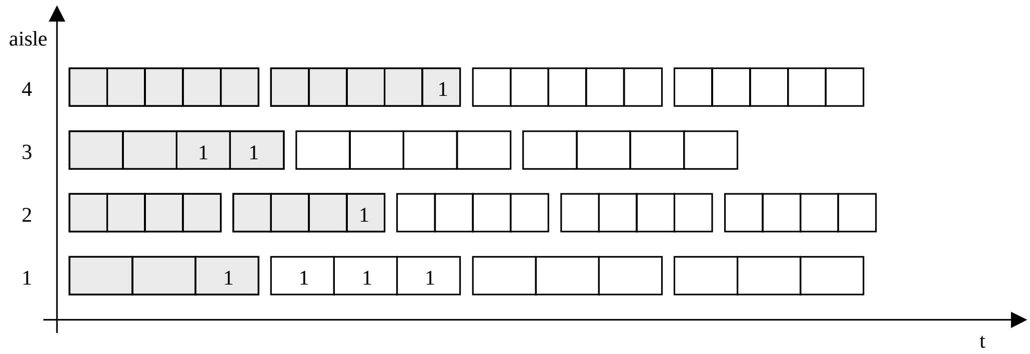

Figure 3. Scheduling of critical orders (a)

In the next step of the heuristic all open tours are taken under consideration. For these tours critical orders are determined that may be scheduled completely on open tours. Here, we successively determine the open tour with minimal arrival time t_{arr}^{open} not considered yet and assign all orders *i* that can be scheduled with $t_{arr}^{open} = t_{arr}(i)$. In the example t_{arr}^{open} is defined by aisle 2 in the first iteration. Eventually, no critical customer orders can be scheduled in the first iteration. In the second iteration t_{arr}^{open} is defined by aisle 1 (and 4) and customer order 5 is scheduled (cf. Figure 4).

After all open tours have been considered they are *closed* by assigning *dummy* containers (indicated by a "*d*"). The number of dummy containers for each aisle is defined by the capacity of the SRU times the number of tours to be planned minus the number of containers belonging to critical orders within the aisle. If the number of dummy containers is not enough to fill up an open tour it remains open (as well as the following open tours of the same aisle). In the example given in Figure 4 critical orders 5 and 6 may be scheduled and tours are closed by assigning dummy containers. In aisle 4 the number of remaining dummies is not enough to complete the second tour.

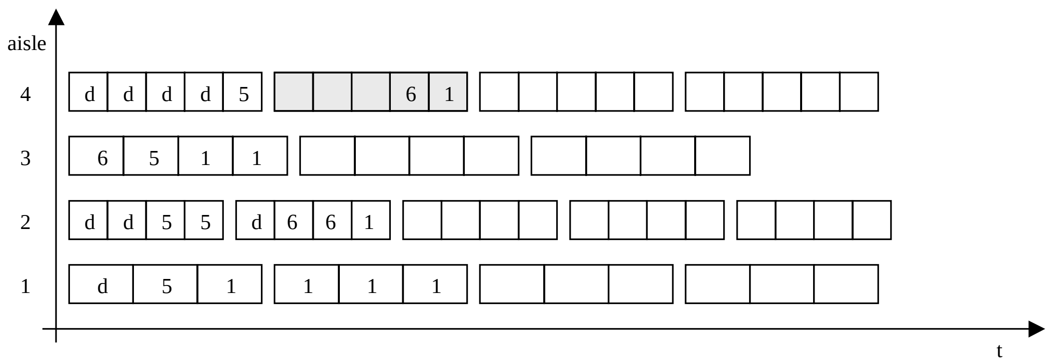

Figure 4. Scheduling of critical orders (b)

The algorithm starts again with the next order that is not scheduled yet and contains the largest sub-order. It terminates after all critical orders have been scheduled. After all orders are scheduled, instances of the VRPTW are formulated for each aisle separately. Here, all orders are taken under consideration, including uncritical orders. Note, that all tours are planned anew.

However, the scheduled order spreads of all critical orders are used to define the time window restrictions for all containers belonging to critical orders. Formally, $tw_e(i)$ for visiting location i is set to $t_{arr}^{last}(j)$ for all locations i that provide articles for (critical) customer order j. Additionally, $tw_b(i)$ for visiting location i is set to $t_{arr}^{first}(j)$ minus the average tour length for all locations i that provide articles for (critical) customer order j. The obtained instances of the VRPTW were tackled applying tabu search [7].

After a schedule has been computed for all aisles a simulation experiment is performed to analyze the solution including the determination of the actual processing time. The results of the experiment may then directly be used for changing the underlying solution within the context of an improvement procedure, as introduced in the next section.

4 Improvement procedure

Improvement procedures are usually local search heuristics, which means that a given solution is changed locally (in means of a limited number of solution attributes to be altered) and the objective function value of the new solution is computed. In our case, this is again obtained by a simulation experiment in each iteration.

The changes within the original solution schedule depend on the degree to that the SRU's capacity has been used on the one hand and the degree to that the capacities in the shipping area have been used on the other hand. In the simulation experiment we record the "actually" achieved order spreads at the packing tables. Comparing these with the scheduled order spreads, that were used for defining the time windows for the instances of the VRPTW, is part of the improvement procedure.

Here, tours are defined as they were recorded in the experiment in the aisles, defining the order spreads anew. If the order spreads were chosen to wide, which is indicated by relatively long (compared to older experiments) blocking times at the packing tables and a large number of containers stored in the buffer, while the SRUs still had capacity available, we try to reduce the average order spread within the improvement procedure and thereby tighten the time windows for the resulting instances of the VRPTW.

If, on the other hand, the SRUs were the bottleneck in the experiment, while containers were retrieved almost continuously, indicated by a small number of containers that had to be stored in the buffer, we loosen the restrictions concerning the time windows defined for the VRPTW of the original formulations, in order to obtain a better solution when solving the modified problem instances of the VRPTW. After the schedule has been modified another simulation experiment is performed and the improvement process starts again. In Figure 5 an outline of the improvement procedure is given.

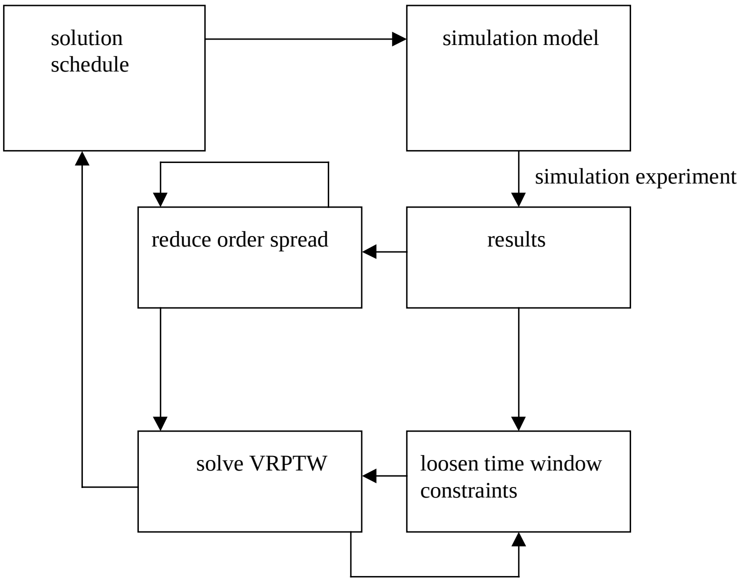

Figure 5. Improvement procedure

In the sequel we will define the local search approach for the sub-routine of reducing the order spread. The approach applied in the improvement procedure is based on the following idea. Consider a given solution with all tours scheduled as illustrated in Figure 6. If we exchange two containers, we can only obtain a change in the objective function value, if the order spread of at least one of the considered orders changes. Two requirements need to be fulfilled: The tour has to define t_{arr}^{first} or t_{arr}^{last} of (one of) the considered customer orders, and the container to be *swapped* with another one has to be the only one of the respective order within that tour.

The so-called *neighbourhood* of a given solution is therefore defined by a *swap move*, where all containers of the considered order are exchanged with containers belonging to critical orders from another tour. In Figure 6 an example is given, where order 1 is taken under consideration and the containers in aisle 1, tour 1 are exchanged with containers from tour 2. By performing this swap move the order spread of order 1 is reduced.

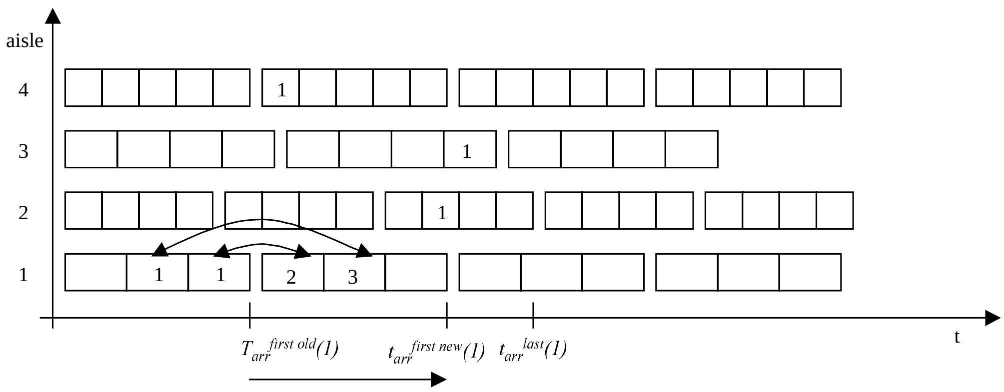

Figure 6. Swap move

In the improvement procedure all swap moves that may reduce the overall order spread are considered. Choosing the move that will decrease the sum of all order spreads the most is known as *steepest descent*, which selects the best neighbour in each iteration until no improving move is possible. After performing swap moves to reduce the order spread of critical orders, the instances of the VRPTW are modified (concerning the time window restrictions) and a new schedule is computed.

If the time windows were chosen too narrow, such that the SRUs turned out to be the bottleneck in the simulation experiment, a different optimization routine is applied. Again, we use a local search technique to loosen time window constraints by iteratively modifying $tw_b(i)$ and $tw_e(i)$ by a predefined step size δt until a significant improvement in the average processing time of the SRUs is achieved or a certain number of re-formulations of the VRPTW has been tested without a significant improvement.

The overall improvement procedure is guided using a meta heuristic called tabu search (cf. Glover and Laguna [2]). The basic idea of tabu search is to use information about the search history to guide local search approaches to overcome local optimality and not to visit solutions again. Primarily, this is done by either statically or dynamically prohibiting moves, that would lead back to solutions already visited. In our case, tabu search is applied solely to both subroutines of the improvement procedure and the VRPTW. For the VRPTW approaches based on tabu search can be found in Potvin et al. [7]. As the number of simulation experiments is restricted due to the time available for the runs, all resulting schedules of an iteration of the improvement procedure are stored in a separate list, and also set tabu.

5 Conclusion

In this paper we have presented an algorithm for determining schedules for storage and retrieval units in an automated warehouse system, which have been developed within the context of a simulation study. The schedules need to meet two requirements: The traveling distance of the units in the aisles should be minimized and customer orders should be handled nearly at the same time in all affected aisles due to space restrictions in the shipping area. We have provided both a two-step algorithm to compute an initial solution and an improvement procedure consisting of two different sub-routines.

For both parts of the overall solution process the objective function value (as a result of the schedules) is obtained by performing a simulation experiment. The result of each simulation experiment is used to decide which sub-routine is to be applied in the improvement procedure. Furthermore, the improvement process is guided using tabu search.

The study might also be regarded as an example to what extent simulation models, if detailed enough, can be used. The usage should not be restricted to the phase of the layout design of warehouses. Furthermore, the model should be used through early production and ramp up (cf. Swain [10]). Other examples and prospects for the extended use of (large scale) simulation models, particularly for their utilization in a real-time environment, can be found in, e.g., Spieckermann and Gutenschwager [9] or Manansang and Heim [5]. In the presented study, the simulation model is an integrative part of the solution concept for scheduling customer orders on a daily basis.

Furthermore, the information obtained by the simulation experiment about the number of employees in the shipping area needed can be used within a more general decision support system. The integration of the simulation model and the presented algorithms in a decision support system will be the next step in the overall project.

References

1. Bodin, L., Golden, B.L., Assad, A.A., Ball, M., “Routing and scheduling of vehicles and crews: The state of the art”, Computers & Operations Research 10, p. 62-221, 1983.

2. Glover, F. and M. Laguna, “Tabu search”, Kluwer, Dordrecht, 1997.

3. Laporte, G., “The vehicle routing problem: An overview of exact and approximate algorithms”, European Journal of Operational Research 59, p. 345-358, 1992.

4. Levasseur, G., ”An object-oriented phone center model using SIMPLE++”, Paper presented at the 1996 Winter Simulation Conference, 1996.

5. Manansang, H., Heim, J., “An online, simulation based patient scheduling system”, Paper presented at the 1996 Winter Simulation Conference, 1996.

6. Potvin, J.-Y., Bengio, S., “The vehicle routing problem with time windows. Part II: Genetic search", INFORMS Journal on Computing 8, p. 165-172, 1996.

7. Potvin, J.Y., Kervahut, T., Garcia, B.-L., Rousseau, J.M., “ The vehicle routing problem with time windows. Part I: Tabu search”, INFORMS Journal on Computing 8, p. 158-164, 1996.

8. Savelsbergh, M.W.P., “Computer aided routing”, CWI, Amsterdam, 1992.

9. Spieckermann, S., Gutenschwager, K., “Discrete event simulation in the automotive industry: Applications and prospects”, in Heller, M. (Ed.), “Automotive Simulation. 5th European Engineering Simulation Symposium”, Society for Computer Simulation, p. 65-74, 1997.

10. Swain, J.J., “Simulation goes mainstream”, OR/MS Today 24 (5), p. 35-46, 1997.