appeared in *Annals of Operations Research* **57** (1995), 29–44

Vehicle Routing in an Automated Warehouse: Analysis and Optimization

Rainer E. Burkard* Bernd Fruhwirth† Günter Rote*

January 1994

Abstract

This study concerns the design of an operating system for vehicles in an automated warehouse. The lay-out of the warehouse, and the number and properties of the vehicles are given. The objective is to maximize the throughput. Using a partial enumeration technique we simulate several alternatives for the control and interplay of the vehicles within a reasonable time horizon. A subproblem is solved by network flow techniques. The algorithm is implemented as part of an automatic control system, and it has lead to a satisfactory performance.

Keywords: automation, heuristic search, partial enumeration

OR/MS subject classification: *Production/scheduling*, planning: several vehicles contending for a single trail. *Manufacturing*, automated systems: automated vehicles in a warehouse.

Contents

---

*Technische Universität Graz, Institut für Mathematik, Steyrergasse 30, A-8010 Graz, Austria
†Salomon Automationstechnik, Friesach 67, A-8114 Stübing, Austria

1 Introduction

In this study we describe the ideas for an operating system for vehicles in an automated warehouse which was built for an Austrian company producing packing material. The company is interested in producing their articles in big charges to avoid to set up the production lines too often. On the other hand, their customers need the packing material only little by little and are not willing or capable to store big charges. Therefore the company decided to build an automated warehouse with high rack stackers.

The warehouse is basically run by three stacker cranes in seven dead-end gangways which are connected by the so-called switching gangway, see figure 1. A detailed description of the lay-out will be given in Section 2. In contrast to conventional warehouses the stacker cranes *are not fixed to disjoint regions* of the warehouse. Every stacker crane is able to use any gangway which implies the possibility of conflicts among the stacker cranes. The stacker cranes have to be coordinated with three further vehicles, one bringing new pallets from the production area to the stacker cranes and two vehicles taking pallets from the stacker cranes to the trucks.

The warehouse has two areas of interface with the "outside world": Pallets come more or less continuously from the production area, and they must be taken into the warehouse immediately, because there is very limited intermediate storage for them. Secondly, trucks wait at the exit of the warehouse to deliver their loads to the customers; they should get their pallets within a reasonable time.

As a measure of the throughput of the warehouse one simply counts the number of double moves per hour. A double move consists of storing one new pallet in the warehouse and of bringing one pallet from the warehouse to a truck. Our objective was to develop an operating system for the warehouse that maximizes the throughput by optimizing the movements of the vehicles. A solution will therefore be a motion plan containing a list of tasks for all the vehicles. We emphasize that the warehouse *design* was already specified. Although the design of a warehouse and the dimensioning of its elements is an interesting problem of its own (see for example the overview of Ashayeri and Gelders [AG85]), this is not the subject of our paper.

The effects of congestion in transportation systems can sometimes be analyzed analytically by methods from queuing theory, see Gudehus [G76, G93]. However, these methods are mainly valid for larger systems, where the traffic elements can be regarded as stochastically independent, and conflicts are resolved by simple priority rules. In our case, we have only three vehicles in the central storage area, and moreover, we use the freedom in assigning tasks to the vehicles in order to *avoid* conflicts and congestion as much as possible, rather than just resolving conflicts by letting one vehicle wait.

The operating system has to make several types of decisions and for most of the decisions a lot of alternatives are at hand. For instance for each new pallet one has several possible storage places and for each such storage place there are different ways to bring the pallet to that place. In Section 3 we list all those questions.

Clearly, one can associate a rooted tree with the set of all possible solutions. Any node corresponds to a specific state of the system where a decision has to be made. The children of a node correspond to the alternatives of this decision. The root corresponds to the first necessary decision under the current state of the system. Even for a time horizon of two minutes the number of decisions and the high number of alternatives per decision prohibit an exhaustive search of the tree. Making a motion plan for more than two minutes does not make sense, since after a few minutes one always gets new information about the incoming pallets. In Section 4 we outline how we find promising branches such that we can enumerate the solution set only partially.

In Section 5 we describe the criteria that influence the decisions in detail. The methods for evaluating alternate possibilities range from simple ad-hoc decisions to network flow models.

In the final section we report on some experiences with the implementation.

2 Description of the lay-out

The storing process takes place in three steps: In the *arrival area* the pallets are brought into the warehouse and intermediately stored at one of three gates. In the *central storage area* three stacker cranes take them up and bring them to their destination in the warehouse. After storing a pallet a stacker crane typically takes out another pallet to the *dispatching (delivery) area.* There, two small vehicles deliver the pallets to the trucks.

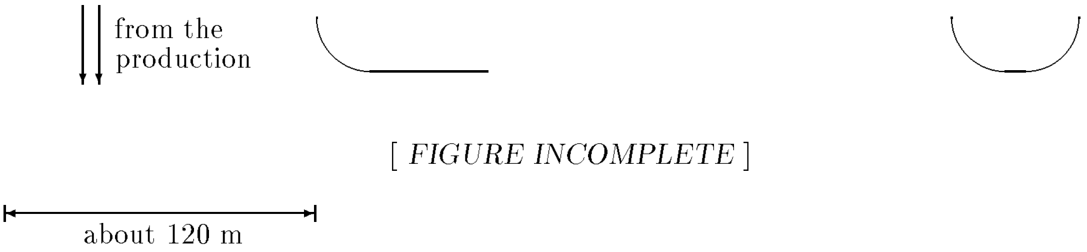


Figure 1: Lay-out of the arrival area.

The arrival area consists of a long gangway (see Figure 1). At its beginning the pallets come out of the production area and are loaded on the so-called "turbo"-vehicle which has space for four pallets and brings them to the three *entrance gates* which connect the arrival area with the central storage area. Two pallets can be delivered at each of the three gates.

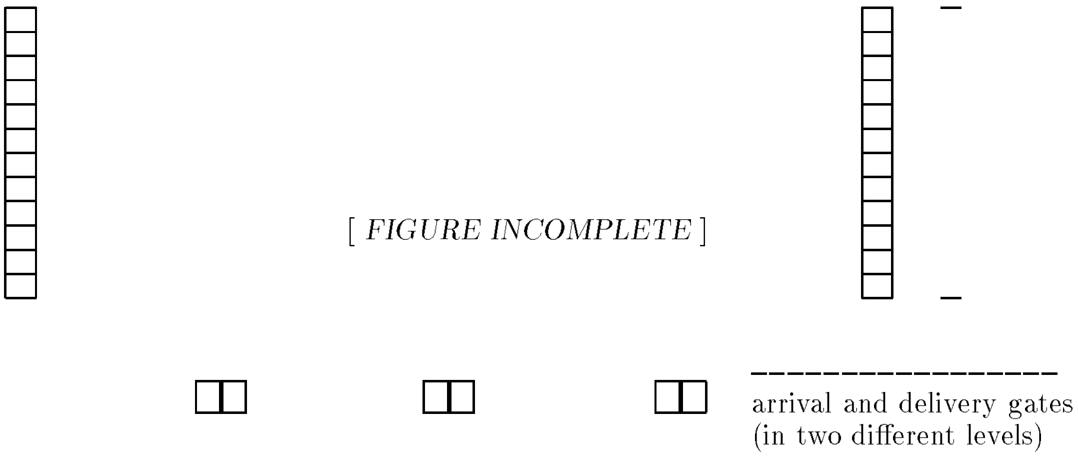


Figure 2: Lay-out of the central storage area

The central storage area (see Figure 2) consists of 24 meter high rack stackers along seven dead-end gangways with a length of 60 meters each. The warehouse has a storage capacity of about 8700 pallets, and it is 95 % full on average.

The seven gangways are connected by the so-called switching gangway. Along the switching gangway there are the three entrance gates and three *delivery gates* which connect the central storage area with the arrival and dispatching area. The entrance gates are located on the top level of the warehouse, the delivery gates are located on the ground level underneath the entrance gates. Three stacker cranes move on rails between the gates and the gangways. Every stacker crane can carry only one pallet at a time. From the switching zone the stacker cranes move via switches into the gangways. Since the control of the stacker cranes is implemented by light signals, only one stacker crane can enter a gangway at a time; however, no gangway is permanently assigned to a stacker crane, i. e., every stacker crane is allowed to use any gangway.

One of the main problems is that the stacker cranes hinder each other in the gate- and switching zone. The stacker cranes must obey a certain safety distance which very often prohibits more than two stacker cranes from using the switching zone at the same time. The stacker cranes spend quite a lot of time in the switching zone to transfer pallets.

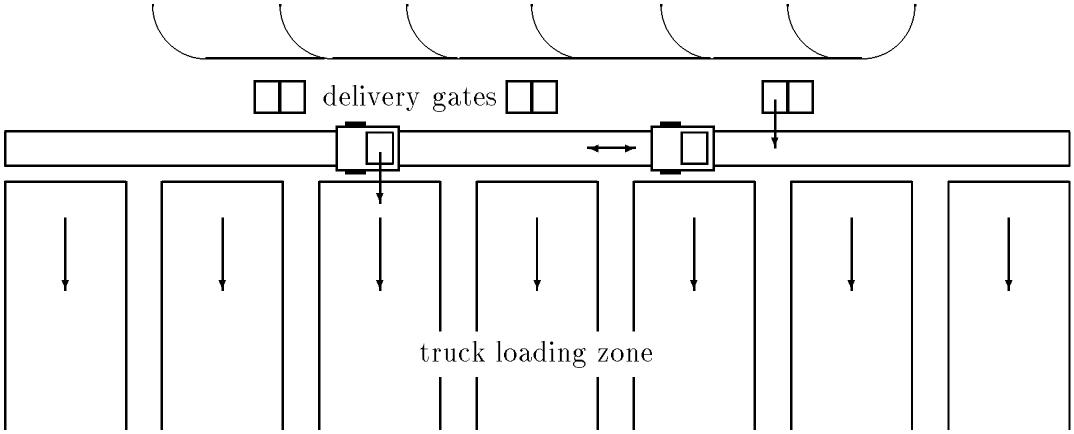


Figure 3: Lay-out of the delivery area

The central storage area is connected via the three delivery gates with the delivery area (see Figure 3). There two small delivery vehicles can fetch one pallet each from a gate and transport it to one of the seven delivery points where the pallet is loaded to a truck. Both delivery vehicles use the same rail, and they cannot pass each other.

Some articles are stored on *unnormed* pallets, which are handled like the usual normed ones except that they can only be unloaded at the central delivery gate.

The control and engineering aspects of this system are described in more detail in Bauer-Kieslinger [B90].

3 Analysis of the system

We list the basic questions which determine the operation of the warehouse, for the control of each of the three vehicle types.

Control of the turbo vehicle (arrival area). The pallets come in a given sequence from the production. The following decisions have to be made:

- In which sequence and
- at which entrance gates should the pallets be unloaded from the turbo?

Control of the stacker cranes (central storage area).

- Which stacker crane takes
- which pallet
- and brings it where in the rack stackers?
- Which (nearby stored) pallet is fetched for delivery and
- is brought to which delivery gate?
- Which stacker crane waits (or moves to the side to make way) in case of a conflict (i. e., if two cranes would get too close in the switching zone)?

Control of the delivery vehicles (dispatching area).

- Which delivery vehicle fetches
- which pallet?
- Which vehicle waits (or moves to the side) in the case of a conflict?

4 Decision points and the exploration of possible solutions

4.1 The solution space

When operating the warehouse one has to continuously make decisions for the questions described in the previous section. For example: When a stacker crane has just unloaded a pallet to a delivery gate, which pallet should be picked up next? A possible solution is characterized by the sequence of decisions made. The set of all solutions can be viewed as a tree; the branching points correspond to the questions that arise during the operation. Each outgoing arc at a branching point represents one possible alternative for the corresponding decision.

This solution tree is infinite, but we look only at a specific time span of about two minutes. This time horizon was determined by the available data about incoming pallets and by practical considerations. So we cut all decisions that occur only later than after two minutes off the tree. A solution, for our purposes, is now a path from the root to a leaf in the solution tree.

The structure of the tree depends on the sequence in which the decisions are made. The next branching point after some initial sequence of decisions can only be determined by simulating this partial solution and looking which decision has to be made next, as in a discrete-event simulation. A reasonable sequence of decisions must (at least roughly) coincide with the temporal order in which the questions pose themselves, because an earlier decision may affect the type of branching point that occurs next. On the other hand, we want to delay each decision to the latest possible time in order to have as much information as possible. (In section 5.2 we will give an example of a decision which is delayed past the point where it would have to be made when actually controlling the vehicles.) We have identified seven critical decision points (events):

1. The turbo vehicle enters the switching zone.
2. A stacker crane takes a pallet from the entrance gate.
3. A stacker crane finishes storing a pallet.
4. A stacker crane enters the switching zone.
5. A stacker crane dispatches a pallet at a delivery gate.
6. A delivery vehicle takes a pallet from the delivery gate.
7. A delivery vehicle finishes dispatching.

The complete solution tree is of course much too complex to be explored completely, even if the time horizon is only two minutes. Therefore, we have to do some partial enumeration of the possible solutions. In order to reduce the size of the tree somewhat, we classify the decisions into two kinds:

- *fixed* decisions which are made for once and never changed again
- *soft* decisions for which alternatives might be tried later in order to get a better solution.

For fixed decisions we do not consider alternative branches, and accordingly we generate no branching points for the solution tree. Every soft decision, on the other hand, corresponds to a branching point which is explicitly stored in the solution tree data structure.

4.2 The partial enumeration procedure

In a first step we make all decisions sequentially, in a more or less heuristic way, based on "local" optimization criteria. (These criteria are discussed in greater detail in the following sections.) In other words, we select one solution path from the root to a leaf.

As we construct the initial solution, we note those places where alternate solution paths branch off. Then we examine all those branching points and we select one which looks most promising to possibly yield a better solution. We examine this branch by starting at this node of the old solution, making the alternate decision, and following the path straight to a leaf. Our tree (as far as we have explored it) has now two leaves, and several places where alternate paths could branch off. In general we have a set of possible solution paths and some possible alternate decisions that would branch off one of these solutions (see Figure 4). We select one of these alternate branches and follow the path from there to a leaf.

Note that this method is different from the graph search procedure usually employed in heuristic search algorithms (cf. Nilsson [N80]), because when we explore a new possibility we always follow some path to a leaf before exploring other alternate branches of the solution. This ensures that we get many *complete* solutions fast, instead of having a lot of good but only partial solutions.

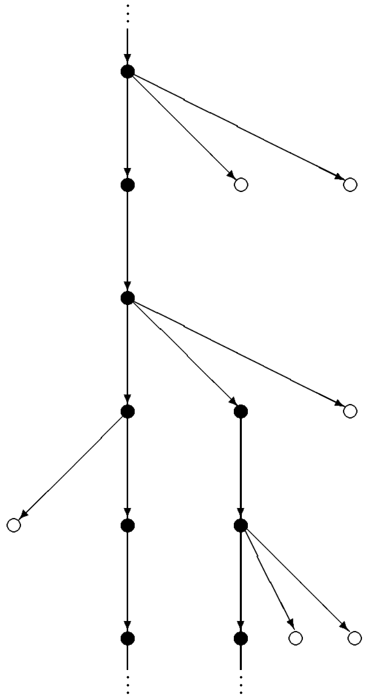


Figure 4: A section of the solution tree. Empty nodes denote alternatives which have not been explored yet.

4.3 The embedding of the optimization algorithm in the operation of the warehouse

After running the above algorithm for some time, the best path in the tree is accepted as actual control for the next time period. (In our implementation, this time bound was set to 10 seconds on a HP 9000/815 workstation.) At this point, the solution consists of a list of high-level "commands", like: Stacker crane 3 waits at the exit of gangway 5 until stacker crane 2 has passed. This program is now translated into a sequence of low-level instructions that is used for actually controlling the vehicles. Since the actual duration of actions of the vehicles cannot be computed precisely, this plan has to be event-driven. This means that commands like "start moving" or "unload" are not issued at predetermined times, but they are triggered by the conditions on which they depend. In this way the system will operate smoothly even if the actual execution does not exactly correspond to the planned and simulated solution.

After some time (about a minute), a new optimization run is made. A new optimization run is also made whenever new information becomes available, e. g., when a new set of pallets that is to be brought to the trucks becomes known, or when some part of the system breaks down. The optimization run starts from the current (updated) status of the system, but it ignores any decisions of the previous plan which have not yet taken effect. In any case, the newly optimized plan is used immediately in place of the remaining part of the previous plan.

4.4 The objective function

The decisions concerning turbo, the delivery vehicles and the stacker cranes mutually influence each other. Both simulations and the first experiences indicated that the real bottleneck is the storage/delivery operation of the cranes and not the operations in the arrival or delivery area. It is most important to minimize the idle movements and the waiting time of the stacker cranes. Thus we can measure the quality of a solution by the total number w of *idle stacker crane seconds*, which is defined for each stacker crane as the elapsed time minus the theoretical minimum possible time corresponding to the work which the stacker crane has done, ignoring any conflicts with the other vehicles. To compare solutions of different length we use the normalized quantity w/t, where t is the total elapsed time of a solution.

We have to specify how we select the next alternative solution branch that we explore in the partial enumeration procedure. For each alternative we try to roughly estimate the expected savings Δw in idle stacker crane seconds. For example, if we find out that a certain decision has led to a conflict, we may expect an improvement from an alternative decision, and we will have $\Delta w < 0$ for that alternative; if we test the second- and third-best alternatives of a decision, they will have $\Delta w > 0$. We rank all alternatives of all branching points according to the following somewhat ad-hoc formula:

$$\frac{w}{t} + 2 \cdot \frac{\Delta w}{t + t_0}$$

where t is the time when the alternate decision comes into effect and w is the idle stacker crane time until t. This expression is a weighted sum of two terms: The first term makes it more likely to explore variations of good solutions than variations of bad solutions. The second term accounts for the expected savings. It is also normalized in order to encourage branches that occur early in the solution tree. These branches will more likely lead to a completely different solution. A fixed amount of $t_0 = 60$ seconds is added to the normalizing denominator so that its influence is weakened for alternatives that branch off very early. Since idle times occur irregularly, the comparison according to the first term w/t might sometimes not be significant; therefore we have given more weight to the second term by multiplying it with the factor 2. (Of course this factor has to be adjusted by experiments.)

5 Criteria for the specific decisions

The decisions are influenced in quite different ways. Among the general objectives is the minimization of idle stacker crane seconds and the avoidance of foreseeable conflicts. In the following we outline the main ideas for each particular situation. In some cases we will be rather explicit about the method by which the alternatives are compared. Most often we will only state the criteria which govern the decisions. The alternatives are ranked according to a weighted sum of numbers representing those criteria, but we will not go into details. Clearly this involves some parameters which can be tuned experimentally for optimal results.

We will deal in succession with the seven decision points mentioned at the beginning of Section 4.

5.1 The assignment of pallets to gangways

When the turbo vehicle enters the switching zone we must choose an entrance gate for each of the pallets. To facilitate the movements of the stacker cranes the gate should be close to the gangway where a pallet will be stored. Therefore we will determine a gangway for each of the new pallets also at this point. To ease the delivery of pallets in the future we want to distribute the pallets of each specific article uniformly over all gangways. (However, we should also take into account that unnormed pallets can

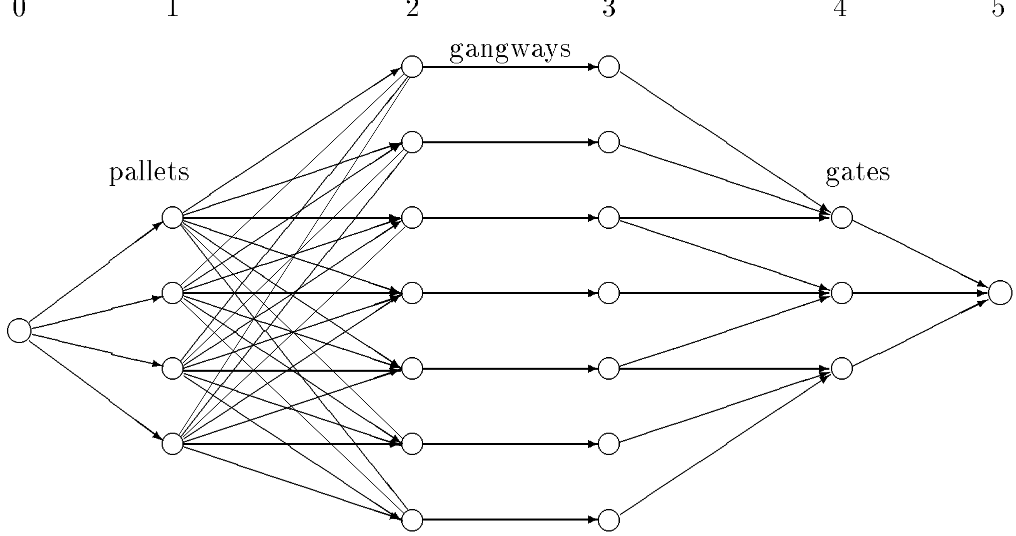


Figure 5: The network flow problem for assigning gates and gangways to pallets.

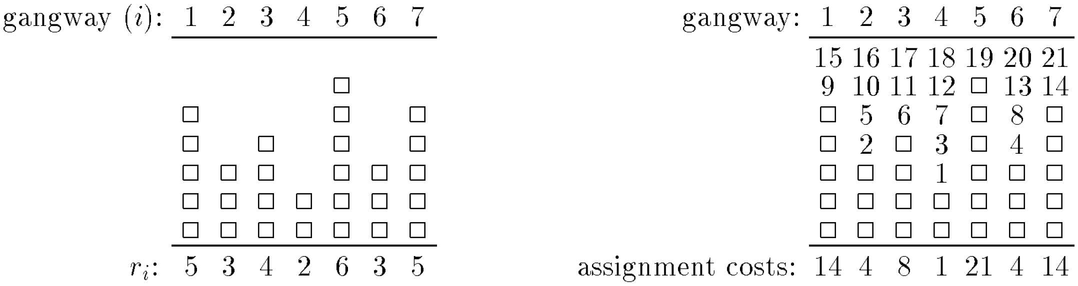
only be unloaded from the main storage area through the central delivery gate, and therefore they should be stored only in the middle gangways).

The assignment of pallets to entrance gates and further to gangways can be modeled as a minimum cost network flow problem in a layered network (Figure 5): In the first layer we have (at most) four nodes representing the pallets carried on the turbo. In the second layer we have one node for every gangway. In layer 3 we repeat these nodes. Layer 4 contains 3 nodes representing the three entrance gates, and layer 5 is just a sink node. Arcs connecting the nodes of layer 2 with the corresponding nodes of layer 3 have cost 0 and capacity 1, which means that we allow at most one pallet per gangway. The capacity of the arcs connecting nodes of layer 4 with the sink equals the number of free intermediate storage places at the entrance gates. All other arcs have infinite capacity. The costs of the arcs leading from a node in layer 1 to a node in layer 2 represent the desirability of storing the pallet in the respective gangway which depends on the number of already stored pallets of this article type in this gangway. Let $r_1, r_2, \ldots, r_7$ denote the number of pallets per gangway for a specific article. Then the cost of assigning a pallet to gangway i is defined as

$$\sum_{j=1}^{7} \max(r_i + 1 - r_j, 0).$$

This is the number of pallets that are necessary to fill up all other gangways to a level of at least $r_i + 1$ pallets of this article. For example, suppose that the pallets are distributed over the gangways as shown in the left part of Figure 6. If we fill the gangways with additional pallets, always putting a pallet into the most empty gangway, the pallets are inserted in the order shown in the right half of Figure 6. The cost of assigning a pallet to a particular gangway i (shown in the last row of Figure 6) is the largest number that appears in the row above the topmost pallet of that gangway.

The arcs between layers 3 and 4 bear costs representing the proximity of the gate to the corresponding gangway. The costs of the arcs between layers 4 and 5 represent the desirability of using certain gates. We want to have at least one pallet waiting for a stacker crane at each of the three entrance gates, because this will give the stacker



gangway (i):	1	2	3	4	5	6	7
					□		
	□				□		□
	□		□		□		□
	□	□	□		□	□	□
	□	□	□	□	□	□	□
	□	□	□	□	□	□	□
r_i:	5	3	4	2	6	3	5

gangway:	1	2	3	4	5	6	7
	15	16	17	18	19	20	21
	9	10	11	12	□	13	14
	□	5	6	7	□	8	□
	□	2	□	3	□	4	□
	□	□	□	1	□	□	□
	□	□	□	□	□	□	□
	□	□	□	□	□	□	□
assignment costs:	14	4	8	1	21	4	14

Figure 6: The computation of assignment costs of pallets to gangways

cranes the most flexible choice of a gate for fetching a pallet. Thus, if a gate has two free places, we put two unit-capacity arcs from its node in layer 4 to the sink, and we give one of these arcs a very negative cost. The other arc has zero cost. We also give zero cost to the single leaving arc for a gate with one free place.

If gangways with many pallets to be removed should be preferred (because a stacker crane ideally stores a pallet in a gangway and the fetches a pallet from the same gangway), this can be modeled by costs on the arcs between layers 2 and 3.

Note that the order of the layers is not the order in which the pallets "flow" through the system (from the gates to the gangways), but this order allows to model the constraints and objectives conveniently.

An optimal solution of this minimum cost flow problem yields for every involved pallet the entrance gate where it should be brought to, and the gangway where it will be stored.

If for some technical problems a certain gate or a certain gangway signals that it is out of order, the capacity of the corresponding arc is set to zero. The algorithm will balance the load among the remaining parts of the warehouse.

In the actual system, some pallets must be brought from production directly to the trucks, without being stored in the warehouse. The stacker cranes must move such a pallet from an entrance gate to a delivery gate. (This is mainly a vertical movement.) These pallets can also be incorporated into our network model by additional nodes and arcs with appropriate capacities that allow to bypass the gangway arcs. We omit the details, because they do not add anything interesting.

5.2 Optimization of stacker crane movements

We now investigate the four decision points concerning the stacker cranes.

A stacker crane takes a pallet from an entrance gate: The gangway has already been fixed at this point, so we need only choose a definite storage place in the rack stackers. For each product, the long-term quantity and frequency of production is known, and hence the expected storage time in the warehouse can be computed. Clearly, products with an expected short storage time (a higher turnover) are to be stored at easier accessible parts in the warehouse (e.g. in the parts of the gangways closer to the gates) than goods with an expected long storage time (so-called ABC-analysis).

Ideally, we would want to consider the distribution function $F(t)$ of articles whose storage time is at most t:

$F(t) =$ The proportion of articles among the articles *in the warehouse* whose expected storage time is at most t.

If we know the (empirical) distribution function

$G(t) =$ The proportion of articles among the articles *produced* whose expected storage time is at most t,

where the different articles are weighted according to the quantity (of pallets) in which they are *produced* per year, then we can compute $F(t)$ as follows:

$$F(t_0) = \frac{\int_0^{t_0} t\, dG(t)}{\int_0^{\infty} t\, dG(t)} = \frac{\int_0^{t_0} (G(t_0) - G(t))\, dt}{\int_0^{\infty} (1 - G(t))\, dt}$$

An article with storage time t should be stored at a distance $F(t) \cdot l$ away from the switching zone, where l is the length of the gangway.

Of course all these data about articles are just averages which are subject to seasonal changes as well as long-term trends. In practice, the turnover rate t according to which the articles are sorted was set to their average storage time during the last two months, and $F(t)$ was computed from the articles currently in the warehouse.

Thus, the type of product specifies a desired storage place, at least in the horizontal direction. A final definite storage place is selected from the available places in the vicinity of the desired place. The proximity to a pallet which is going to be removed plays also a role.

Some pallets have to be passed from the arrival area directly to the dispatching area via a stacker crane. The choice of the delivery gate for such a pallet are governed by the same considerations as for pallets that come from the rack stackers (see below).

A stacker crane finishes storing a pallet: At any time there is a given set of pallets from which the next pallet to be removed can be selected. This set is determined in the following way: Customer orders ask for certain articles that have to be loaded on trucks and delivered. However, not all articles of a truck load are available for removal at once, since a truck may serve several customers, and the different products have to be loaded onto the trucks in a specified order. In addition, the warehouse uses the first-in-first-out rule: Of each product that is to be delivered, the oldest pallets in the warehouse must be selected. On average, about 40 pallets are available for removal at a given time.

The above considerations are beyond the control of the optimization strategy. The first-in-first-out rule, however, is the reason why it is important to achieve a balanced distribution of pallets of a particular product over the gangways when the pallets are stored: In this way it is guaranteed that the pallets that are to be removed will also be distributed evenly, and there will be no congestion in single gangways.

To summarize, from the viewpoint of the optimization algorithm there is a current set of pallets which are available for removal, and we have to choose the pallet which is removed next. Our algorithm makes this choice depending on the proximity to a just stored pallet and on time priorities: A pallet which has waited for a long time for delivery must eventually be removed. In addition, special user-specified priorities can be taken into account here.

In order to have more information about possible conflicts which might be affected by this decision, we use the trick of *delayed decision points*: If a stacker crane has just stored a pallet, we compute the earliest time at which it may possibly return to the switching zone. This is the earliest time at which some interaction between other vehicles takes place. It is clear that there is no need for the other vehicles to know about the actions of this stacker crane (other than that it occupies a certain gangway). Therefore we delay the decision about the actions of this stacker crane until this delayed time which is computed as above. More precisely, we use this delayed time point to synchronize the decisions of the various vehicles. In this way we can to some extent look into the future without any effort: when we make our decision what the stacker crane should do in its gangway, we already know a lot more about the other vehicles' plans. For example, we may find that if our crane takes a certain pallet from the rack stackers, the exit of the gangway will just be blocked by another crane when the first crane is about to leave the gangway. Thus, since we would have to lose time anyway, we might consider fetching a pallet which is further down in the gangway. By this simple trick, many conflicts between stacker cranes that would otherwise arise can be avoided.

A stacker crane enters the switching zone: The choice of the delivery gate depends on the proximity between the accessible gates and the gangway where it comes from. Unnormed pallets can only be dispatched at the middle gate. In the beginning we also considered the proximity of the delivery gate to the actual track leading to the waiting truck. This criterion was omitted since it turned out that the delivery vehicles have usually no problem to fulfill their tasks. When the system runs they serve the stacker cranes rather than limit them.

A stacker crane dispatches a pallet at a delivery gate: For choosing the next pallet to be taken by a stacker crane we consider the following: A pallet at the "home gate" (i. e., the entrance gate directly above the gate where the stacker crane has just unloaded a pallet) should be preferred for the next storage operation. Also entrance gates where both places are occupied by waiting pallets should be preferred. Moreover, pallets which have to be stored in a gangway with many storage and delivery operations should be preferred, since these might later produce a bottleneck.

5.3 Handling of conflicts for stacker cranes

Conflicts occur in the switching and gate zone which is used by all three stacker cranes in common. Some conflicts can already be avoided by the kind of limited look-ahead discussed above. In order to resolve conflicts we must select one of the two cranes and let it wait. We can base this choice on simulating (for the range of this conflict only) the loss of effectively used stacker crane seconds. If the two choices do not differ much, we can add the second choice as an alternative to our tree. In addition, we might consider alternate decisions for the involved stacker cranes which would eliminate the conflict altogether. More specifically, we look back at the list of decisions for the two vehicles, and we consider the alternatives of the latest decision. These alternate actions might avoid the conflict, and we add them to our solution tree. (These are the kind of branches that will have $\Delta w < 0$ in the evaluation function.) However, as described above, we do not backtrack immediately, but we follow the selected path (including the conflict) and extend it until we have a solution. Only later will we consider these alternatives again (among all other alternatives that branch off the tree).

5.4 Optimization for the delivery vehicles

The decisions for the delivery vehicles are similar as for the stacker cranes, although the choices are much more limited. Thus we state the decision criteria only briefly: Delivery vehicles should remove pallets from gates with two waiting pallets. Moreover, time priorities again play a role as well as the actual moving pattern of the two delivery vehicles which forbids crossings (see the following paragraph).

5.5 Handling of conflicts for the delivery vehicles

Since the delivery vehicles cannot cross, conflicts can occur only in a limited number of ways, and they can be handled quite easily. In contrast to the conflicts of stacker cranes, where the alternate possibilities of handling a conflict are tried out as part of the implicit enumeration procedure of the solution tree, we can here check all possible solutions explicitly. In principle we may have the following operations performed by the two vehicles A and B:

a_1: A fetches a pallet from a delivery gate.

a_2: A brings its pallet to a certain dispatching track.

The operations b_1 and b_2 are defined in an analogous way. In case of a conflict there are now six possibilities to bring these four single operations a_1, a_2, b_1, and b_2 into a linear order, namely:

$a_1 — a_2 — b_1 — b_2$: B waits until A has delivered.
$a_1 — b_1 — a_2 — b_2$: B waits with fetching, then A waits with dispatching, and finally B dispatches.
$a_1 — b_1 — b_2 — a_2$: etc.
$b_1 — b_2 — a_1 — a_2$
$b_1 — a_1 — b_2 — a_2$
$b_1 — a_1 — a_2 — b_2$.

In this context, "waiting" includes the possibility of moving to the side to make way for the other vehicle. The waiting can also be vacuous, in case there is no conflict; then some or all of the six possibilities may coincide. Since there are only at most six possibilities, we can afford to look at them all. A simple evaluation and comparison of these cases immediately leads to an optimal resolution of the conflict.

6 Running the system; concluding remarks

The above ideas were implemented by Salomon Automationstechnik, a software house specialized in control systems for automated warehouses. Since the summer of 1990, the system is actually in use with success at Bauernfeind Co., one of the main producers of packing material in Central Europe. Salomon mainly wanted to get the system running quickly, and they did not carry out extensive experiments or detailed sensitivity analyses which we could report. Therefore we give only some global characteristics of the system.

The program was written in the C language, and it runs on a Hewlett-Packard HP9000/815 workstation with 8 Megabytes of main memory. This workstation is dedicated to operating the warehouse. Besides the real-time control of the vehicles there are only small administrative tasks like database management and the user interface. The optimization is thus the only computationally intensive process on the computer. As mentioned above, the time limit for one optimization run was 10 seconds.

The maximum allowed degree of branching in the decision tree was set to 3. The average depth of the decision tree was about 15. In view of a time horizon of 2 minutes, this means that a decision has to be made approximately every 8 seconds.

We now mention some operating characteristics of the warehouse, and in particular, the stacker cranes. Under reasonable and typical assumptions, a single stacker crane is capable of approximately 25 double moves per hour, ignoring the possibility of hindrance. (This performance figure is computed according to a German DIN norm, which defines a double move as the time for taking a pallet, bringing it to a place one third of the length down the gangway, picking up another pallet two thirds of the length down the gangway in the same lane, and putting it down at the gate.) With the methods described, after some tuning of the parameters, 68 double moves per hour could be achieved with three stacker cranes. If we want to compare this with $3\times 25 = 75$, we must take into account that the "average pallet" which is delivered into or out of the warehouse is more likely to be close to the switching zone than far away (see section 5.2). Still, 68 double moves can be regarded as quite an accomplishment, considering that all three stacker cranes must use the switching zone and they spend a lot of time there to transfer pallets. In addition, they must obey a safety distance which implies that in most situations only two stacker cranes can use the switching zone at the same time.

It soon turned out that the unnormed pallets caused problems. On the average they occur much less frequently than the normed ones. But during certain periods they were used extensively, which led to a congestion at the middle delivery gate. This bottleneck has been removed by providing a second gate with the capability of taking unnormed pallets from the main storage area to the dispatching area. Also, a fourth stacker crane was installed to improve the performance of the warehouse.

An analysis of the stacker crane movements revealed that arrival and delivery gates at very different levels are a drawback, because they incur a time loss of the stacker cranes for the vertical movement between putting down a pallet on a gate and picking up the next one. This experience has been used in the design of future warehouses with similar characteristics.

References

[AG85] J. Ashayeri, L. F. Gelders, Warehouse design optimization, *European J. Operational Research* **21** (1985), 285–294.

[B90] Franz Bauer-Kieslinger, Optimale Bewegungsplanung, *Materialfluss*, September 1990, 53–57.

[G76] T. Gudehus, Staueffekte vor Transportknoten, *Zeitschrift für Operations Research* **20** (1976), B207–B252.

[G93] T. Gudehus, Analytische Verfahren zur Dimensionierung von Fahrzeugsystemen, *OR Spektrum* **15** (1993), 147–166.

[N80] Nils J. Nilsson, *Principles of artificial intelligence*, Tioga, Palo Alto 1980.