Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARTIN GRÖTSCHEL DIETRICH HAUPTMEIER
SVEN O. KRUMKE JÖRG RAMBAU

Simulation Studies for the Online Dial-a-Ride Problem

Preprint SC 99-09 (March 1999)

SIMULATION STUDIES FOR THE ONLINE DIAL-A-RIDE PROBLEM

MARTIN GRÖTSCHEL, DIETRICH HAUPTMEIER, SVEN O. KRUMKE, AND JÖRG RAMBAU

ABSTRACT. In a large distribution center of Herlitz AG, Berlin, we investigated the elevator subsystem of the fully automated pallet transportation system. Each elevator may carry one pallet and has to serve eight levels. The goal is to minimize the average resp. the maximum flow time. The variants of this elevator control problem have been subject of recent theoretical research and are known as online-dial-a-ride problems. In this paper we investigate several online algorithms for several versions of online-dial-a-ride problems by means of a simulation program, developed on the basis of the simulation library AMSEL. We draw statistics from samples of randomly generated data providing for different load situations. Moreover, we provide preliminary studies with real production data for a system of five elevators connected by a conveyor circuit, as can be found at the Herlitz plant. We show which algorithms are best under certain load situations and which lead to break downs under particular circumstances.

1. INTRODUCTION

The pallet transportation system at the Herlitz PBS AG in Falkensee near Berlin (the major European office supply provider) consists of truck loading stations, pallet sorters, conveyor belts (horizontal transportation), elevators (vertical transportation), registration and inspection stations, and an automatic storage system. The flow of pallets from the automatic storage system to the truck loading stations is controlled by a central computer unit. Online decisions with respect to control have to be made, e.g., which way a pallet should travel to its destination, or when a pallet is to be stored in the automatic storage system. There are constraints restricting the organization of the system, for instance the FIFO-principle which requires that pallets must be delivered in the order of their manufacturing date.

The overall goal is to control the system in such a way that there is a quick and congestion free flow of pallets which satisfies all restrictions. Violating constraints inhibits the throughput. For instance, if the capacity of a route is exceeded this might lead to blocking of the automatic storage system. In the mathematical literature transport problems of this type are called "Dial-a-ride problems" (DARP).

The pallets that have to be transported during one day of production are not known in advance. Thus, decisions have to be made *online* without any knowledge of the future. Upon the arrival of a new transportation request a (temporarily) locally optimal control might have bad consequences which even reoptimization can not cure. This situation is mathematically modeled by the online version of DARP called ONLINEDARP.

Decisions do not only have to be made online, they also must be made within severe time restrictions which brings up the *realtime aspect* of the problem.

Figure 1 displays a layout diagram of the pallet transportation system at Herlitz. In this paper we focus on the control of the elevators within this system. The pallet transportation system employs several vertical transportation systems (elevators) in order to move pallets between the eight floors of the building. The elevators are connected by conveyor belts, each elevator can carry one pallet at a time. Figure 2 shows a top view of one of the eight floors. The arrows indicate the direction of the conveyor belts on that floor which

---

Key words and phrases. Simulation, elevators, online-opimizaton, online-dial-a-ride problem.
Research supported by the German Science Foundation (DFG), grant 883/5-2.

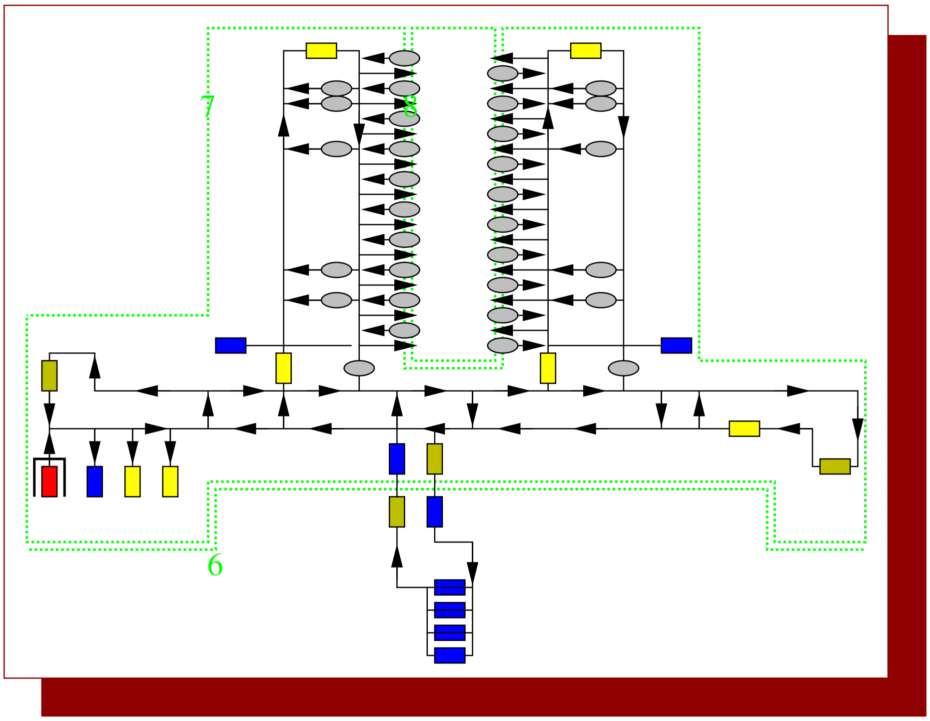

FIGURE 1. Layout of the pallet transportation system.

move pallets between the five elevators. A schematic horizontal view of a single elevator is shown in Figure 3.

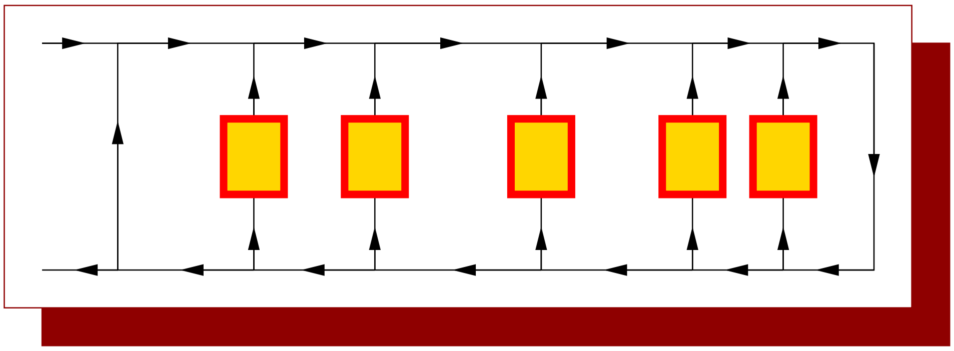

FIGURE 2. Top view of one of the eight floors.

A major problem in the mathematical investigation of the system is the lack of an obvious objective function. The intuitive goal of "maximal throughput" is best reflected by the following two objectives:

- minimize the average flow time, where the flow time of a transportation request equals the sum of its waiting time and the time needed to serve this request;
- minimize the maximum flow time.

Preliminary investigations revealed that the two objectives conflict. Moreover, the corresponding offline problems are NP-hard to solve and not approximable within a constant factor (unless P = NP).

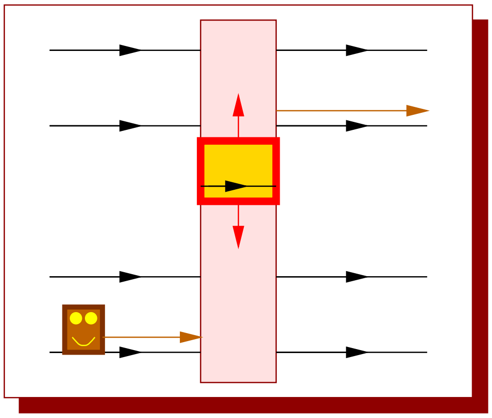

FIGURE 3. Horizontal view of an elevator.

2. RELATED RESEARCH

The model of online computation using time stamps or release dates for the requests was introduced in [6, 5] in the context of the *Online Traveling Salesman Problem*. The problem ONLINEDARP has been investigated with respect to competitive analysis in [2, 7].

DARP is also called the Stacker-Crane-Problem. In [12] it is shown that DARP is NP-hard to solve even on caterpillars, which are trees of maximum degree three. On paths DARP can be solved in polynomial time[4]. Approximation algorithms for DARP are given in [10, 12, 9]. An important extension of DARP with respect to practical applications was investigated in [12]. The authors consider the case when "First-In-First-Out" precedence constraints for the requests are given. It is shown that the problem can be solved in polynomial time on paths but is NP-hard on caterpillars. Approximation algorithms for general graphs and also improved results for trees are presented. Preemptive versions of DARP, where objects are allowed to be dropped at intermediate vertices, are studied in [11, 8].

3. THE SIMULATION MODEL

Our simulation programs are built on top of AMSEL [1], a callable C-library to design event-based simulation programs. The input data consists of a set of *event points*, a set of *modules*, and a collection of *requests*. Every request becomes an *object* which flows through the system via the event points. For every event point a method is specified to derive a successor event from the current state of the system. If an object is on an event point then the event is stored in the global *event list* together with a time stamp, and the object stores it as its current event. The basic flow of objects is modeled as follows: the currently next event in time is read from the event list. Then the successor event is derived together with the point in time when this event should be processed. Now, the object updates its current event point to the successor event, and the successor event is entered into the event list; the old event point is deleted from the list. Modules are closed regions in the system where the number of objects inside is constrained by a capacity value. Modules are entered through entry events and left through exit events. For more details on AMSEL see [3].

All our studies were produced on a Sun UltraSparc 10 workstation. The running time was approx. 2% of the simulated time. This shows that our simulation environment is fast and that the algorithms under consideration are all real-time compliant.

4. THE ALGORITHMS

We tested the following algorithms in our simulation environment:

FIRSTFIT: The server always serves a "nearest request". A request is "nearest", if corresponding empty move, i.e., the move of the empty server to the source of the request, is shortest.

FIFO: Requests are served in the order of their occurrence.

REPLAN: As soon as a new request arrives, the server completes the current carrying move (if it is performing one), then the server stops and does a replan: it computes a new shortest schedule which starts at the current position of the server and takes care of all yet unserved requests.

IGNORE: The server remains idle until the first request becomes known. It then serves the first request immediately. All requests that arrive during the service of the first request are temporarily ignored. After the first request has been served, the server computes a shortest schedule for all unserved requests and follows this schedule. Again, all new requests that arrive during the time that the server is following the schedule are temporarily ignored. A schedule for the ignored requests is computed as soon as the server has completed its current schedule. The algorithm keeps on following schedules and temporarily ignoring requests this way.

IG GREEDY: The algorithm works basically like IGNORE. However, if a new request becomes known and this request can be inserted into the current schedule without increasing the cost, then the new request is added to the schedule.

FF MAXAGE: The algorithm works like FIRSTFIT, but additionally for each unserved request records its waiting time. If the waiting time for a request exceeds the grace period parameter then this request is served next.

FF DYNAGE: This is a modification of FF MAXAGE which does not have a fixed grace period parameter, but which adjusts the parameter dynamically during the run to a fraction of the maximal waiting time of all served request so far. The value of this fraction has to be given to the algorithm as a parameter.

Our partners in industry told us that in their elevator systems are exclusively controlled by FIFO- and FIRSTFIT-type algorithms.

5. Simulation Studies of a Single Elevator with Random Data

The basic layout investigated in this section consists of one elevator and a fixed number of levels. Waiting requests line up in front of the elevator; the requests on one level have to be served in a First-In-First-Out fashion (FIFO). Requests are generated randomly as follows: at every point, the time until the next requests occur is taken from a uniformly at random on the interval $(0, t_\varepsilon]$, where t_ε is an adjustable parameter. The number of requests issued at that time is taken uniformly at random from the integers in the interval $[1, n_{\max}]$, where n_{max} is to be chosen by the user in advance. With these two parameters it is possible to control the load of the system: a smaller t_ε yields a higher load; a larger $n_{\max}$ leads to a stronger peakedness of the input data.

Under low load almost all algorithms yield the same performance concerning the minimization of the average flow time. For the minimization of the maximal flow time we see that FIRSTFIT and REPLAN perform considerably worse than the rest.

If the system works under medium load then we observe that algorithms performing especially well for the average flow time (FIRSTFIT, REPLAN) yield approx. 30% worse results for the maximal flow time than the other algorithms. Note that FIFO is still a feasible strategy with comparably poor performance for the average flow time and good performance for the maximal flow time. The algorithms IGNORE, IG GREEDY, FF MAXAGE, and FF DYNAGE all show a balanced behavior in the sense that average and maximal flow times are not too much apart.

Under high load the FIFO strategy breaks down because it runs in such an inefficient way that requests are issued faster than FIFO can serve them. Algorithm FF MAXAGE almost breaks down because of the following: if many requests are in the system then at some point there will be many requests whose time has run out. All these requests are

Algorithm	$\emptyset$ flow/s			max. flow/s			completion/s		
	$\emptyset$	max	min	$\emptyset$	max	min	$\emptyset$	max	min
FIRSTFIT	**45**	**49**	**41**	276	409	192	**57648**	**57722**	**57616**
FIFO	*48*	*54*	*43*	227	420	**163**	*57649*	**57722**	**57616**
REPLAN	**45**	50	*43*	*278*	*490*	*197*	**57648**	**57722**	**57616**
IGNORE	46	51	42	221	383	164	**57648**	**57722**	**57616**
IG GREEDY	46	50	42	224	381	164	**57648**	**57722**	**57616**
FF MAXAGE	**45**	**49**	**41**	**220**	370	175	*57649*	**57722**	**57616**
FF DYNAGE	**45**	**49**	**41**	226	**342**	189	**57648**	**57722**	**57616**

System Parameter: .... 7 systems, 1 elevator each, 9 FIFO levels.
Request Parameter: .... $t_\varepsilon = 165s$, $n_{\max} = 3$.
Statistics Parameter: .... simulated time: $8h$, sample size: 20.

TABLE 1. Simulation studies for low load.

Algorithm	$\emptyset$ flow/s			max. flow/s			completion/s		
	$\emptyset$	max	min	$\emptyset$	max	min	$\emptyset$	max	min
FIRSTFIT	**54**	**60**	**48**	*410*	*715*	*263*	57667	57737	**57619**
FIFO	*68*	*87*	*55*	287	477	**206**	*57675*	*57751*	**57619**
REPLAN	55	64	49	371	539	253	**57664**	**57716**	**57619**
IGNORE	59	69	51	279	381	**206**	57669	*57751*	**57619**
IG GREEDY	58	67	50	279	412	210	57667	57719	**57619**
FF MAXAGE	57	66	50	**256**	375	209	57667	57737	**57619**
FF DYNAGE	55	61	49	270	**365**	229	57667	57737	**57619**

System Parameter: .... 7 systems, 1 elevator each, 9 FIFO levels.
Request Parameter: .... $t_\varepsilon = 120s$, $n_{\max} = 3$.
Statistics Parameter: .... simulated time: $8h$, sample size: 20.

TABLE 2. Simulation studies for medium load.

then scheduled first-in-first-out, which is inefficient. Thus, the number of requests whose age is too large will increase, resulting in an even worse situation. We conclude that FF MAXAGE is unstable under heavy load and under large peakedness. While the average performance on 20 instances of FF DYNAGE is acceptable, we see that there is a large deviation. Hence, we consider FF DYNAGE not very robust. For both FF MAXAGE and FF DYNAGE we have to cope with the problem that parameters have to be adjusted to the actual system; this makes both algorithms difficult to use in praxis. Algorithm FIRSTFIT works very efficiently under high load (small average flow time). This is plausible because if many requests are in the system then the probability that FIRSTFIT can proceed without non-carrying moves is large. A similar argument explains why REPLAN also yields a small average flow time: the potential for the optimization of a schedule is the larger the more requests can be planned. Both algorithms, however, trade the flow times of individual requests for the global efficiency. In this respect, IGNORE and IG GREEDY show the most balanced behavior, where the average flow time results of IG GREEDY show that the insertion of additional requests at no extra costs pays off.

Algorithm	$\emptyset$ flow/s			max. flow/s			completion/s		
	$\emptyset$	max	min	$\emptyset$	max	min	$\emptyset$	max	min
FIRSTFIT	**125**	**175**	**98**	1119	1380	791	57751	**58064**	57653
FIFO	*3017*	*4676*	*1765*	*5887*	*8103*	*3840*	*63357*	*65706*	*60884*
REPLAN	135	194	110	1074	1576	781	**57734**	58072	**57644**
IGNORE	215	383	155	864	1290	651	57816	58331	57653
IG GREEDY	185	292	135	**832**	**1234**	584	57799	58200	57653
FF MAXAGE	782	1889	276	1628	2858	875	58680	60325	57662
FF DYNAGE	234	1094	109	869	2171	**484**	57796	58416	57653

System Parameter: .... 7 systems, 1 elevator each, 9 FIFO levels.
Request Parameter: .... $t_\varepsilon = 75s$, $n_{\max} = 3$.
Statistics Parameter: .... simulated time: $8h$, sample size: 20.

TABLE 3. Simulation studies for high load.

We conclude that online control of elevator systems shows a trade-off between "global efficiency" (small average flow time) and "balanced behavior" (small maximal flow time).

Algorithm	average flow	maximal flow	completion
FIRSTFIT	**108**	**312**	**86363**
conveyor	**78**	**303**	
elevator	**29**	**104**	
FIFO	**108**	**312**	**86363**
conveyor	*79*	**303**	
elevator	**29**	**104**	
REPLAN	**108**	**312**	**86363**
conveyor	**78**	**303**	
elevator	**29**	*170*	
IGNORE	**108**	**312**	**86363**
conveyor	*79*	**303**	
elevator	**29**	**104**	
IG GREEDY	**108**	**312**	**86363**
conveyor	*79*	**303**	
elevator	**29**	**104**	
FF MAXAGE	**108**	**312**	**86363**
conveyor	**78**	**303**	
elevator	**29**	**104**	
FF DYNAGE	**108**	**312**	**86363**
conveyor	*79*	**303**	
elevator	**29**	**104**	

System Parameter: .... 7 systems, 5 elevators each, 8 levels.
Requests: .... Real request data, simulated data: 24*h*.

TABLE 4. Simulation study using real data.

The load of the system has a substantial influence on the performance of the algorithms under consideration. Our simulation renders FIFO unusable under high load, while FIRSTFIT and REPLAN show an inacceptable imbalance between average and maximal flow time. The performance of the algorithms FF MAXAGE and FF DYNAGE heavily depends on their parameter settings. We were not able to find parameters for them that worked equally well in all load situations.

By ignoring upcoming requests IGNORE and IG GREEDY do not ignore waiting requests for an arbitrarily long period of time, thereby showing a stable behavior with respect to the maximal flow time; the local improvement procedure of IG GREEDY enhances it with a better performance for the average flow time than IGNORE. We propose IG GREEDY for robust control of an elevator system under varying load situations.

6. Preliminary Studies of an Integrated Elevator System

The layout investigated in this section integrates five elevator systems into a conveyor system. This is the layout of the elevator system in the distribution center of Herlitz. On each level, pallets are transported on circuits which have connections to single-capacitated waiting slots in front of the elevators. See Figure 2 in the introduction for a sketch. For the following simulation studies, the simulation parameters, such as capacities and times that pallets take for traveling between components, were adjusted to reflect the real situation at the Herlitz plant.

We first observe that in this particular system vertical transportation is faster than horizontal transportation. A pallet takes about 4s to move from one floor to another (not counting another 5s for boarding and also for alighting from the elevator). However traveling from the system entrance to the first elevator takes 15s and completing a whole journey around the conveyor belt takes more than 2 minutes.

This observation together with the fact that there is at most one pallet waiting for an elevator on each floor could lead to the conclusion that the scheduling of elevators has a rather small influence on the performance of the system.

In fact, looking at a simulation using real data as shown in Table 4, we see that the systems behave very similar, when different algorithms are used for controlling the elevators.

	average flow			maximal flow			completion		
Algorithm	mean	high	low	mean	high	low	mean	high	low
FIRSTFIT	**120**	**122**	*118*	340	438	239	*28924*	**28968**	**28879**
conveyor	**87**	**89**	**86**	**281**	**333**	**155**			
elevator	**32**	**34**	**31**	167	270	105			
FIFO	*121*	*125*	*118*	341	**427**	*264*	**28923**	*28973*	**28879**
conveyor	**87**	**89**	**86**	*289*	**333**	**155**			
elevator	*33*	*37*	**31**	140	175	106			
REPLAN	**120**	123	*118*	*354*	*554*	**213**	*28924*	**28968**	**28879**
conveyor	**87**	**89**	**86**	283	**333**	**155**			
elevator	**32**	35	**31**	*210*	*435*	*140*			
IGNORE	*121*	123	*118*	337	449	255	**28923**	**28968**	**28879**
conveyor	**87**	**89**	**86**	287	**333**	**155**			
elevator	*33*	36	**31**	**132**	**168**	**96**			
IG GREEDY	**120**	123	**117**	**336**	**427**	257	**28923**	**28968**	**28879**
conveyor	**87**	**89**	**86**	287	**333**	**155**			
elevator	*33*	36	**31**	136	174	**96**			
FF MAXAGE	**120**	**122**	*118*	340	438	235	*28924*	**28968**	**28879**
conveyor	**87**	**89**	**86**	**281**	**333**	**155**			
elevator	**32**	**34**	**31**	153	194	105			
FF DYNAGE	**120**	**122**	*118*	340	438	235	*28924*	**28968**	**28879**
conveyor	**87**	**89**	**86**	**281**	**333**	**155**			
elevator	**32**	**34**	**31**	153	182	105			

System Parameter: .... 7 systems, 5 elevators each, 8 levels.
Request Parameter: .... $t_\varepsilon = 90s$, $n_{\max} = 3$.
Statistics Parameter: .... simulated time: $8h$, sample size: 20.

TABLE 5. Simulation study with low random load.

	average flow			maximal flow			completion		
Algorithm	mean	high	low	mean	high	low	mean	high	low
FIRSTFIT	**159**	**163**	**155**	644	815	562	29023	29121	28946
conveyor	**103**	**105**	**100**	519	735	369			
elevator	**56**	**60**	**54**	456	572	398			
FIFO	*208*	*220*	*199*	*814*	*1088*	*675*	*29041*	29095	*28953*
conveyor	*119*	*125*	*113*	*688*	927	*565*			
elevator	*88*	*97*	*84*	308	347	261			
REPLAN	162	166	157	634	902	484	**29005**	**29073**	**28939**
conveyor	**103**	**105**	**100**	**499**	734	**352**			
elevator	58	61	56	*504*	*831*	*402*			
IGNORE	170	176	166	623	828	524	29012	29090	28943
conveyor	106	108	103	561	801	499			
elevator	63	67	61	273	306	241			
IG GREEDY	169	173	163	**611**	828	**452**	29027	29113	*28953*
conveyor	106	108	103	526	801	369			
elevator	62	66	60	281	336	259			
FF MAXAGE	167	171	161	638	**755**	506	29018	29102	28941
conveyor	105	108	102	535	**713**	372			
elevator	61	65	58	**256**	**295**	**223**			
FF DYNAGE	161	165	157	662	798	528	29007	*29123*	28947
conveyor	**103**	**105**	101	553	*967*	375			
elevator	57	62	56	274	299	256			

System Parameter: .... 7 systems, 5 elevators each, 8 levels.
Request Parameter: .... $t_\varepsilon = 30s$, $n_{\max} = 3$.
Statistics Parameter: .... simulated time: $8h$, sample size: 20.

TABLE 6. Simulation study with medium random load.

A similar pattern emerges, when running the system on randomly generated data. The load used for generating Table 5 is chosen similar to the maximum load observed in the real data.

However, when we increase the load of the system, some trends appear which are somewhat similar to the observations for elevators without a connecting conveyor system. The following observations draw on Tables 6 and 7.

The algorithms FIRSTFIT and REPLAN once again show the best average flow behavior. We also notice again that these algorithms are performing badly when considering maximal flow times through the elevator subsystem, ie., the maximal time it takes a pallet

Algorithm	average flow			maximal flow			completion		
	mean	high	low	mean	high	low	mean	high	low
FIRSTFIT	**175**	**180**	**171**	778	**866**	682	**29058**	29169	28972
conveyor	**111**	**114**	**109**	634	797	**517**			
elevator	**63**	**67**	**61**	575	*795*	*451*			
FIFO	*293*	*350*	*256*	*1752*	*4019*	*1141*	*29181*	*29350*	*29058*
conveyor	*171*	*211*	*146*	*1602*	*3811*	*1048*			
elevator	*121*	*138*	*110*	337	390	313			
REPLAN	180	186	176	776	1062	684	29063	**29149**	28965
conveyor	113	116	111	**615**	**789**	520			
elevator	66	70	64	*589*	744	413			
IGNORE	191	201	186	795	1012	647	29069	29155	28961
conveyor	117	122	115	714	979	579			
elevator	73	79	70	296	356	257			
IG GREEDY	190	200	184	771	992	**637**	29068	29152	28965
conveyor	117	121	114	676	842	565			
elevator	72	78	69	303	397	260			
FF MAXAGE	192	203	186	835	1163	669	29088	29191	**28941**
conveyor	118	123	115	716	993	565			
elevator	74	80	70	**291**	331	**242**			
FF DYNAGE	178	184	175	**755**	1013	**637**	29074	29222	28945
conveyor	112	116	110	636	984	528			
elevator	65	69	64	295	**326**	265			

System Parameter: .... 7 systems, 5 elevators each, 8 levels.
Request Parameter: .... $t_\varepsilon = 25s$, $n_{\max} = 3$.
Statistics Parameter: .... simulated time: $8h$, sample size: 20.

TABLE 7. Simulation study with high random load.

Elevator	Passengers
1st elevator	683
2nd elevator	188
3rd elevator	45
4th elevator	8
5th elevator	5

TABLE 8. Results of elevator simulation with 24h traffic data from Herlitz's Falkensee plant

from entering the elevator waiting slot until emerging from the elevator on its target floor. However, in the special settings of the Herlitz system the conveyor belt levels off most of these effects, so that the overall maximal flow times of FIRSTFIT and REPLAN are not much worse than those achieved by other algorithms.

Once again FIFO performs initially quite well, however with rising load both its average and maximum flow times deteriorate compared to other algorithms.

The performance of the modified first fit algorithms FF MAXAGE and FF DYNAGE is for all samples similar to that of IG GREEDY. Preliminary studies with both higher load and systems with more floors suggest that this remains true. This would suggest that the instability of FF MAXAGE and FF DYNAGE which was observed for elevators with waiting queues is not apparent, when there is at most one passenger waiting at each floor. However this needs to be verified in further studies of the algorithms.

Additionally to controlling the elevators, the system has to decide on which elevator a pallet is to be transported. Currently this is done using a "first fit" strategy: pallets enter the first free waiting slot which they encounter. This obviously leads to the first elevators being used much more extensively than the further elevators as shown in Table 8.

In future research we will also look at other strategies for assigning pallets to elevators. However, the current "first fit" algorithm could prove a very good strategy for the system under consideration, since in this particular case vertical transportation is much faster than horizontal transportation and therefore choosing the first possible elevator could well be efficient.

7. Conclusion

We have simulated various algorithms for the basic elevator control problem in varying load situations. In particular, we focused on the performance with respect to minimizing the average and the maximal flow times, resp. It turned out that algorithms seeking for highest possible "global efficiency" (FIRSTFIT, REPLAN) may leave single requests unserved for an unacceptable long period of time. Algorithms like FF MAXAGE or FF DYNAGE seem to require a new parameter setting for extreme load situations—otherwise the system may break down completely. Parameters suitable for normal load have both algorithms imitate the inefficient FIFO strategy under high load. The algorithms IGNORE and IG GREEDY achieve a good balance between the two objectives in every load situtation.

Our preliminary simulation studies on the integrated elevator system at Herlitz show the same effects if only the elevator performance is considered, even though there waits at most one request per level and elevator. However, the complete system suffers from the poor performance of the conveyor system so that the optimization effects on the single elevators alone do not lead to a substantially higher overall performance. This suggests to further integrate the optimization efforts on both the elevator subsystem and the conveyor control.

References

1. *Amsel—a modelling and simulation environment library*, Developed at Konrad-Zuse-Zentrum für Informationstechnik, Berlin. Online-Documentation at `http://www.zib.de/ascheuer/AMSEL.html`.
2. N. Ascheuer, S. O. Krumke, and J. Rambau, *Competitive scheduling of elevators*, Preprint SC 98-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin, November 1998.
3. Norbert Ascheuer, *Hamiltonian path problems in the on-line optimization of flexible manufacturing systems*, Dissertation, Technische Universität Berlin, 1995.
4. M. J. Atallah and S. R. Kosaraju, *Efficient solutions to some transportation problems with applications to minimizing robot arm travel*, SIAM Journal on Computing **17** (1988), no. 5, 849–869.
5. G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo, *Serving request with on-line routing*, Proceedings of the 4th Scandinavian Workshop on Algorithm Theory (SWAT'94),, Lecture Notes in Computer Science, vol. 824, July 1994, pp. 37–48.
6. ______, *Competitive algorithms for the traveling salesman*, Proceedings of the 4th Workshop on Algorithms and Data Structures (WADS'95), Lecture Notes in Computer Science, vol. 955, August 1995, pp. 206–217.
7. E. Feuerstein and L. Stougie, *On-line single server dial-a-ride problems*, Manuscript, submitted for publication, 1998.
8. G. N. Frederickson and D. J. Guan, *Preemptive ensemble motion planning on a tree*, SIAM Journal on Computing **21** (1992), no. 6, 1130–1152.
9. ______, *Nonpreemptive ensemble motion planning on a tree*, Journal of Algorithsm **15** (1993), no. 1, 29–60.
10. G. N. Frederickson, M. S. Hecht, and C. E. Kim, *Approximation algorithms for some routing problems*, SIAM Journal on Computing **7** (1978), no. 2, 178–193.
11. D. J. Guan, *Routing a vehicle of capacity greater than one*, Discrete Applied Mathematics **81** (1998), no. 1, 41–57.
12. D. Hauptmeier, S. O. Krumke, J. Rambau, and H.-C. Wirth, *Euler is standing in line*, Preprint SC 99-06, Konrad-Zuse-Zentrum für Informationstechnik Berlin, March 1999.

Konrad-Zuse-Zentrum für Informationstechnik, Takustr. 7, 14195 Berlin, GERMANY
E-mail address: {groetschel,hauptmeier,krumke,rambau}@zib.de