

Planning and Control of Automated Material Handling Systems: The Merge Module

Sameh Haneyah, Johann Hurink, Marco Schutten, Henk Zijm, Peter Schuur

Abstract We address the field of internal logistics, embodied in Automated Material Handling Systems (AMHSs), which are complex installations employed in sectors such as Baggage Handling, Physical Distribution, and Parcel & Postal. We work on designing an integral planning and real-time control architecture, and a set of generic algorithms for AMHSs. Planning and control of these systems need to be robust, and to yield close-to-optimal system performance. Currently, planning and control of AMHSs is highly customized and project specific. This has important drawbacks for at least two reasons. From a customer point of view, the environment and user requirements of systems may vary over time, yielding the need for adaptation of the planning and control procedures. From a systems' deliverer point of view, an overall planning and control architecture that optimally exploits synergies between the different market sectors, and at the same time is flexible with respect to changing business parameters and objectives is highly valuable. An integral planning and control architecture should clearly describe the hierarchical framework of decisions to be taken at various levels, as well as the required information for decisions at each level, e.g., from overall workload planning to local traffic control. In this research, we identify synergies among the different sectors, and exploit these synergies to decompose AMHSs into functional modules that represent generic building blocks. Thereafter, we develop generic algorithms that achieve (near) optimal performance of the modules. As an example, we present a functional module from the Parcel & Postal sector. In this module, we study merge configurations of conveyor systems, and develop a generic priority-based real-time scheduling algorithm.

Sameh Haneyah
University of Twente, 7500 AE Enschede P.O. Box 217, The Netherlands, e-mail:
s.w.a.haneyah@utwente.nl

1 Introduction

In our effort to identify synergies among the AMHSs used in different business sectors, we decompose AMHSs into functional modules that represent generic building blocks. Thereafter, we develop generic algorithms that achieve (near) optimal performance of the modules. Few authors address generic planning and control of AMHSs, Boyd et al. [2] review general control forms. Chen et al. [1] present a control framework for AMHSs using the holonic approach, however it does not solve detailed operational material handling problems. In this paper, we focus on the merge module, which appears in different types of AMHSs using conveyor belts (see Figure 1). Merge configurations consist basically of several infeed conveyors (or *infeeds*) that transport items to a *merge conveyor* on which the items are gathered. We develop a generic real-time scheduling algorithm that allocates space on the merge conveyor to items transported by the infeeds. In literature, studies related to merge configurations are more analogous to traffic systems than conveyor belts (see Shladover [7]). Studies relevant to conveyor systems are mostly simulation-based (El-Nashar and Nazzal [6]), where scheduling and control are not evident beyond simple routing rules. Many studies deal with AMHSs as a supporting resource within a manufacturing environment (see Bozer and Hsieh [3]).

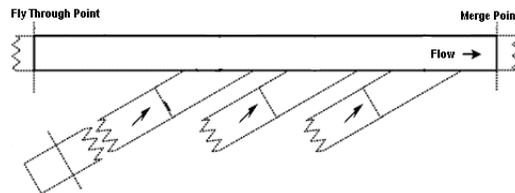


Fig. 1 Merge configuration.

The objective of the algorithm is two-fold: throughput maximization, and workload balancing among infeeds. Briefly stated, throughput refers to the output of the merge area measured in the average number of items passing per hour at the *merge point* (see Figure 1). This is equivalent to space utilization on the merge conveyor. Workload balancing refers to an even distribution of workload among all infeeds, this is reflected by synchronizing waiting times on the infeeds. As items on the infeeds look for merge spaces on the merge conveyor, space should be allocated in a way that results in an even distribution of waiting time elapsing before the items are physically merged. Current practices show that it is possible to optimize for one of these objectives, but at the cost of deteriorating the other objective. Therefore, it is important to maintain the right balance between the two conflicting objectives. In general, throughput is the main objective, and it is mostly unacceptable to optimize for workload balancing at the cost of low throughput. The main research question for this sub-problem is: How can the space on the merge conveyor be allocated to the parcels on the infeeds to achieve high throughput and a workload balance among

infeeds, and how to keep the right balance between these conflicting objectives? A relevant example from practice for this problem comes from the parcel & postal business sector. Figure 2 shows the generic layout of an express parcel sorting system. At the unload area, operators place parcels onto the infeeds. These infeeds transport parcels to the merge conveyor represented by the big loop in Figure 2. The merge occurs when the parcels transported on the infeeds reach the merge conveyor. After parcels are transferred to the merge conveyor, they circulate on the merge conveyor to reach the load area. There, parcels are automatically sorted to their destinations. Parcels are released into sorting *chutes* (see Figure 2). These chutes end at big containers where parcels with the same destination are collected in preparation for further transport. *Flying through* parcels are those parcels that flow back into the unload area, because their destination chutes were full or had some disruptions. McWilliams [4, 5] works on the parcel hub scheduling problem, but focuses on loading and unloading schedules, where the AMHS is dealt with as a black box .

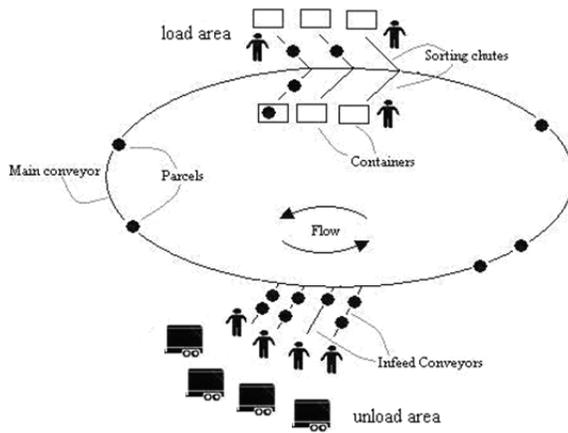


Fig. 2 Generic layout of a closed-loop conveyor sorting system.

We focus on this particular example. Section 2 illustrates the challenge of workload balancing that arises due to the early reservation phenomenon, and presents the general idea of the algorithm we develop to control the merge operation. Section 3 reports on some experimental results. Finally, Section 4 concludes this paper.

2 Early reservation phenomenon and algorithmic design

Figure 3 sketches a merge configuration with two infeeds of the same length; each infeed has a parcel. As soon as a parcel is loaded on an infeed, it is announced in the system and requests a merge space. The parcel from infeed 1 can arrive at point A, and therefore can reserve it. The parcel from infeed 2 can arrive at point

B. However, if all infeeds are busy with parcels, then already at an earlier decision moment, point B could have been allocated to some previous parcel from infeed 1. This phenomenon induces the dedication of most of the space as required by parcels from infeed 1, while forcing parcels from infeed 2 to wait for a space at a later point than requested. The main point is that parcels from infeed 1 can reserve spaces on the merge conveyor earlier than parcels from infeed 2, due to the restricted look ahead horizon. Therefore, as the system operates for long time, the total waiting time for parcels of infeed 2 accumulates. Moreover, when there is a larger system with more infeeds, this phenomenon propagates, and may result in high imbalance measures. The phenomenon occurs mainly when infeeds farther downstream are not long enough to see all incoming parcels (that compete for the same merge space) at the time of making allocation decisions. Then, parcels from those infeeds are forced to wait before being merged, more than parcels from upstream infeeds.

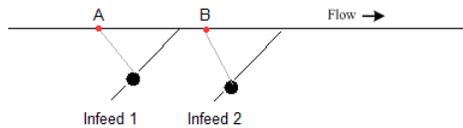


Fig. 3 Free space search.

We take a dynamic approach to the space allocation problem in merge configuration. We develop a generic real-time priority-based algorithm (PBA). The generic PBA contains three main processes:

- **Process A: Merge space search process:** This process is a simple search for an empty space on the merge conveyor. An announced parcel (on an infeed) may activate this process to search for an available merge space.
- **Process B: The pending requests queue process:** this process allocates empty space, appearing at the merge conveyor, to parcels waiting to be allocated. It uses priorities in allocating available space to waiting parcels. Priorities are calculated according to a formula that gives weights to space utilization and workload balancing. This process is activated by empty space appearing in the merge area.
- **Process C: The reallocation process:** this process aims at balancing the workload among infeeds by changing allocation decisions after more information becomes available in the system. This is an improvement procedure that is activated by (*priority*) parcels and not by empty space. Some parcels may overrule the space allocated to other parcels, and claim the space as long as it is possible to change allocation decisions. Overruling is governed by certain priority calculations.

3 Experimental results

We build a simulation model to analyze the performance of the algorithm given different operating conditions and different infeed lengths. We model a merge configuration with four infeeds. Our main input variables are: parcels' lengths, parcels' inter-arrival times, and the density of flying through parcels. The latter is described in terms of the space occupancy on the merge conveyor, when appearing at the *fly through point*, which is the upstream boundary of the merge configuration (see Figure 1). Normally, parcels are loaded successively when they belong to the same batch, but it takes some time until a new batch starts loading. In the simulation experiments the distribution of inter-arrival times within the same batch is an input variable, using three different uniform distributions. However, batch arrivals are always embedded in the experiments with a certain probability in the same manner, and so provide an input that is not varied. We first perform simulation experiments to tune some parametric values used in the algorithm. Later, we report on our main output measures, which reflect the aforementioned objectives: First, space utilization on the merge conveyor. Second, relative difference in waiting times, calculated as the difference between the maximum and minimum values of total waiting times on the modeled infeeds, divided by the maximum value of total waiting time among infeeds. Moreover, we report on the frequency of reallocations.

In standard operating conditions, the algorithm achieves good results (utilization exceeds 90%, and relative difference in waiting times is 1-2%). Main remarks are: First, as inter-arrival times increase, space utilization on the merge conveyor drops. The explanation of this trend is intuitive. When inter-arrival times are long, much space on the merge conveyor is missed as the conveyor moves. Second, given a fixed range of inter-arrival times, increasing the density of flying through parcels increases utilization of the merge conveyor in most of the cases. As the density of flying through parcels increases, parcels on the infeeds are more likely to wait for merge spaces, and so more time is available to make more informed allocation decisions. Allocation decisions are more likely to be made by Process B (see Section 2), which considers space utilization in calculating priorities. Third, for short and medium ranges of inter-arrival times, varying the density of flying through parcels has a negligible effect (1%) on the relative difference in waiting times. Fourth, an operating condition where long inter-arrival times are combined with no flying through parcels, results in relatively high imbalance measures (20%). In this case, the effect of process B is limited, because the possibility that parcels from different infeeds are simultaneously waiting for space allocations is small. Moreover, the effect of process A is limited, because changing space allocation decisions becomes less feasible. The reason is late awareness of new incoming parcels, due to long inter-arrival times, and less waiting times of available parcels. Less waiting time is caused by no or low density of flying through parcels. We discuss that this condition may not create a problem in practice. We also argue that better workload balancing may not be achievable even with other algorithms, as long as the layout of the merge configuration is not altered. However, we show how small variations in the lengths of the infeeds helps overcome this issue. Furthermore, we experiment with different

infeed lengths, the impact of the reallocation process, and the relation between the number of reallocations executed and the length of the infeeds. We also report on the distribution of waiting time, and of reallocations, among infeeds.

4 Conclusion

The generic PBA works on maximizing throughput as the main objective, and then executes a reallocation procedure to balance the workload as the second objective while not deteriorating throughput. Simulation shows that the algorithm gives good results for different layouts of the merge configuration, and under varying operating conditions. An important lesson learned is that a layout design with certain control rules can have dramatic results. However, an integrated solution of layout design and control rules may result in significant improvements. This study was conducted in cooperation with an international supplier of express parcel sorting systems, where the algorithm is currently being implemented (Company name is concealed for confidentiality reasons). Going back to the wide picture of our research, which is planning and control of AMHSs, we see this study as an important component of the overall project. We provide a generic algorithm that can be adapted to different operational conditions while maintaining the same generic structure. For example, the calculations of priorities within the algorithm may include different criteria. Although the merge operation occurs in AMHSs of different sectors, it is mostly critical for express parcel sorting systems. This is due to the emphasis on providing sorting systems with high throughput, and so it is crucial to use the space within the system as efficiently as possible. Since the generic PBA can be applied to different merge configurations, it can replace the various customized control algorithms that are currently used for different systems and different sectors.

References

1. Chen F.F. Sturges R.H. Babiceanu, R.F. Framework for the control of automated material-handling systems using the holonic manufacturing approach. *International Journal of Production Research*, 42(17):3551–3564, 2004.
2. Boyd N.P. Whorms H.H. Dilts, D.M. The evolution of control architectures for automated manufacturing systems. *Journal of Manufacturing Systems*, 10(1):79–93, 1991.
3. Bozer Y.A. Hsieh, Y.-J. Analytical modeling of closed-loop conveyors with load recirculation. volume 3483, pages 437–447, 2005.
4. Douglas L. McWilliams. A dynamic load-balancing scheme for the parcel hub-scheduling problem. *Computers & Industrial Engineering*, 57(3):958 – 962, 2009.
5. Douglas L. McWilliams. Iterative improvement to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, 59(1):136 – 144, 2010.
6. El-Nashar A. Nazzal, D. Survey of research in modeling conveyor-based automated material handling systems in wafer fabs. 2007.
7. S.E. Shladover. Operation of merge junctions in a dynamically entrained automated guideway transit system. *Transportation Research Part A: General*, 14(2):85–112, 1980.