Layout-Optimized Sorting of Goods with Decentralized Controlled Conveying Modules

Zäzilia Seibold, Thomas Stoll and Kai Furmans
Institute for Material Handling and Logistics (IFL)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{zaezilia.seibold, thomas.stoll, kai.furmans}@kit.edu

Abstract—To increase flexibility in intralogistics, decentralized controlled material handling systems have been developed. We investigate the suitability of one of these systems, the FlexConveyor, for sorting of goods. The FlexConveyor is a material handling system built out of multiple, identical modules, each equipped with a controller. By communicating with each other, the modules are able to cooperate and to transport goods from any source to its specific destination. For sorting of goods, densely connected layouts promise high throughput while requiring little space. To compare different layouts, to identify bottlenecks and to draw conclusion about algorithm optimizations, we do a layout analysis partially based on methods coming from the network analysis of national transportation networks. The results of this layout analysis are compared to experimental results of a discrete event simulation model.

Keywords—sorting goods, decentralized control, network analysis

I. INTRODUCTION

Today’s trends like e-commerce and product individualization cause more and more complex material handling systems in intralogistics while market prediction becomes less reliable. Thus, material handling systems should be flexible to adapt to changing needs without losing efficiency. Basic tasks in intralogistics comprise among others transporting, merging, sorting and buffering of goods [1]. Plug&Work material handling systems have decentralized control and enable a flexible adaption of the system to user’s requirements [2]. The system is specifically build for currently defined conditions and can be changed easily.

An example for Plug&Work material handling systems is the FlexConveyor, a decentralized controlled conveying system built out of multiple, identical modules that can be easily plugged and unplugged. It is optimized for the task of transporting goods in a user-defined layout. An item can be introduced in the system at any source and is transported to its specific destination. Complex transport tasks of multiple items with different sources and destinations can be performed thanks to the decentralized control and usage of alternative routes.

In this paper we concentrate on the specific task of sorting of goods. This means that items introduced at multiple entries in the system should be transported to their specific destination [3]. The objective is to produce high throughput while using the less space possible.

Today’s sorting systems are centralized controlled and not easily adaptable. Thus, they are usually oversized and high investment is necessary. The two common layouts of sorting systems are the ring structure and the line structure, both requiring a lot of space [4]. Also, the entry and exit points have to be at defined locations in order to optimize throughput.

With a dense network built out of FlexConveyors forming a closed surface (example see Fig. 1), a high-density and high-performance sorting system can be build which is easily adaptable to changes and allows flexible entry and exit points. The objective of this paper is to analyze how powerful the FlexConveyor is in sorting of goods and what kind of layouts are most suitable. For this reason, we do a layout analysis based on existing methods of network analysis and draw conclusions about algorithm optimizations.

II. RELATED RESEARCH

Several research institutes are working on flexible material handling systems with decentralized control. The approaches
and considered material handling tasks are different, but they all have the objective to develop flexible, self-adapting systems. One challenge of decentralized control is to design algorithms for decision-taking with locally restricted view, and at the same time, to achieve with these algorithms stable and efficient system behavior.

The Internet of Things describes a vision, where autonomous entities communicate and take routing decisions based on local information [5]. MultiShuttle Move is an autonomous Automated Guided Vehicle (AGV) which develops swarm intelligence by several vehicles working together. The allocation of jobs is done by negotiating and vehicles are able to drive on the floor and as shuttles in storage systems. Another autonomous AGV named KARIS [6] cannot only perform the transport of single items but is designed to form two different functional clusters, as shown in Fig. 2: As discontinuous cluster, four KARIS vehicles connect to each other in order to transport huge items, for example a pallet. As continuous cluster, several KARIS vehicles form a conveyor line to realize high throughput of goods.

![Figure 2. Single KARIS vehicle, discontinuous and continuous cluster](image)

To handle dynamic and structural complexity of logistics networks, autonomous capabilities for decentralized coordination of autonomous logistic objects in heterarchical structures are required [7]. New control strategies with autonomous logistic objects have been developed to ensure flexibility and reactivity to dynamically changing external influences while maintaining global goals.

In the research project CogniLog, several applications of cognitive logistics systems are developed [8]. One of these applications is a highly functional intralogistics node built out of small-scaled modules (see Fig. 3). Each module is smaller than the transported item and disposes of a controller. Together acting as functional node, they can perform tasks like transporting, collecting, separating, sorting, merging, sequencing and aligning of goods.

![Figure 3. Cognitive Conveyors as high-functional intralogistics nodes [8]](image)

Other control algorithms for material handling applications are developed that could be technically realized with a decentralized controlled system like the FlexConveyor. The three systems GridStore, GridPick and GridSequence are based on the idea of puzzle-movement. GridStore is a storage system combining high-density and high throughput [9]. GridPick is a picking system which requires little space, reduces the picking time and, thus, increases throughput. GridSequence puts goods that are incoming in arbitrary order in a defined order [10].

One of the main advantages of FlexConveyor is its technical simplicity. As it is only moved and installed by the user, the safety for people working in the same area can easily be guaranteed. The FlexConveyor logic can be adapted and installed in any conventional hardware which facilitates industrialization. In this paper we want to focus on the application of sorting with the FlexConveyor. For layout changes and the associated flexibility, we define the Plug&Play-capability as important characteristic of future material handling systems.

### III. THE FLEXCONVEYOR

The FlexConveyor can convey in the four cardinal directions. Being connected to neighboring modules (see Fig. 4), the modules can communicate by point-to-point message passing. The FlexConveyor has to fulfill several tasks in order to transport items to their destination without deadlocks. An underlying, permanent process is the recognition of the conveying network. Once a module identifies a new neighbor, this connection information is propagated through the network. Each module establishes an adjacency matrix of the complete network. With this matrix, the modules are able to compute necessary information about routes to possible destinations. With the Dijkstra algorithm [11], the path lengths to all destinations via the four cardinal directions are calculated.

![Figure 4. A conveying network of FlexConveyor modules](image)

Once an item is introduced in the network, the carrying conveyor starts the reservation process: A reservation message is sent into the direction of the shortest path to the destination. Each FlexConveyor receiving a reservation message decides on its own in which direction to forward the message. To prevent simple deadlocks, the receiving and sending conveying direction is reserved and, thus, blocked for items in opposite direction. If the shortest path is blocked because of another reservation, alternative routes are used. Once the reservation message has reached the destination, a confirmation message is sent back to the source and the transport is started. During transport, circular deadlocks have to be prevented which is done by message passing before conveying to the next FlexConveyor module [12].
IV. NETWORK ANALYSIS OF SORTING LAYOUTS

The objective of a sorting system is to produce high throughput while requiring low space and low resources. The decentralized algorithm of FlexConveyor is designed for user-defined layouts. For the application of sorting, densely connected networks are suitable because they require low space while offering numerous routing options. Layouts built out of FlexConveyor modules can be modeled as networks with nodes representing the FlexConveyor modules and edges representing conveying directions.

Usually, performance analysis of decentralized controlled systems is solely based on simulations as there is no analytical method for decentralized controlled systems. To analyze sorting layouts prior to simulations, we do a network analysis of the layouts to examine how suitable the layouts are for sorting of goods, independently of the applied decentralized algorithm.

Several measures exist to evaluate the connectivity and the accessibility of transportation networks, such as roads and airline networks. We use some of these measures for the sorting layout analysis and also define indices specifically for the application of sorting of goods. In the following, the definition of some measures and their meaning for our application are explained:

- **Required space**: This measure describes the rectangular surface which is used by the layout. It is given in number of module sites.

- **Required modules**: This measure is defined by the number of nodes, i.e. FlexConveyor modules.

- **Average shortest path**: This measure is the average of the shortest paths from every entry to all exits. It can also be described as an adapted Shimbel index [13], which is defined as sum of the shortest distances from one node to all other nodes.

- **Average number of direction changes**: A change of transportation direction requires additional time because the item has to be stopped before being transported in the new direction. The average number of direction changes is based on the shortest paths.

- **Alpha-Index (Cyclomatic Index)**: This measure describes the circularity of the network [14]. Applied to sorting of goods it can give an insight about the probability of congestion and the risk of deadlock.

- **Ratio of lowest to highest Shimbel-Index**: The Shimbel Index [13] measures the accessibility of a node. The lower the index, the more central the node is located. The lower the ratio is, the easier it is to distribute the throughput over the network.

Fig. 5 shows four different layouts which have been chosen for comparison. Every layout has 8 entries and 8 exits with defined incoming/outgoing directions (gray arrows). Every border modules has either an entry or exit. The square layout and the disk seem to be suitable for sorting because they represent shapes with high surface requiring low space. The ring structure is one of the commonly used sorting layouts.

![Figure 5. Sorting layouts with 16 entries/exits each](image)

Table 1 shows the results of the layout comparison. The ring layout, similar to existing sorting structures, needs fewest modules. But on the other side, it has the longest average path length and the highest number of direction changes. Risk of deadlock is low, also shown by a low Alpha-Index. All conveying modules have the same Shimbel-Index, their usage will be uniform.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Square</th>
<th>Ring</th>
<th>Circular disk</th>
<th>Trapezoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>16</td>
<td>16</td>
<td>49</td>
<td>36</td>
</tr>
<tr>
<td>Modules</td>
<td>25</td>
<td>16</td>
<td>37</td>
<td>24</td>
</tr>
<tr>
<td>Shortest paths</td>
<td>5.75</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Direction changes</td>
<td>1.5</td>
<td>2.75</td>
<td>1.5</td>
<td>1.63</td>
</tr>
<tr>
<td>Alpha-Index</td>
<td>0.36</td>
<td>0.04</td>
<td>0.35</td>
<td>0.28</td>
</tr>
<tr>
<td>Shimbel ratio</td>
<td>0.69</td>
<td>1.00</td>
<td>0.68</td>
<td>0.58</td>
</tr>
</tbody>
</table>

The square layout has three advantages: it requires little space, has short path lengths and only needs 1.5 directions changes in average. Deadlock risk is higher shown by a high Alpha-Index. A circular disk requires more space and more modules and, thus, increases the average path length. The trapezoid needs less conveyor modules than the square layout and has a lower risk of deadlocks. It has the lowest ratio of lowest/highest Shimbel-Index which is an indication for a bad distribution of usage.

Usually, a high number of modules increase the average path length, which can have to contrary effects on the throughput. On one hand, it leads to higher lead times and, therefore, to lower throughput. On the other hand, it decreases congestion and the risk of deadlock because there is more space available. The impact of these effects will be analyzed in section VI.
V. ENHANCEMENTS OF THE FLEXCONVEYOR ALGORITHM

A. Routing decision taking

In a densely connected network, there are many possible routes to transport the items to their destination. Routing of the items through the network has essential influence on system performance. Each FlexConveyor has to take the decision in which direction the reservation message, i.e. the item, should be forwarded. In the existing algorithm of the FlexConveyor, this decision is only taken based on path length and route availability. Directions with the same path length are chosen with the same probability. In the quadratic layout, for example, the border modules would try to change the conveying direction for every second incoming item. The traversing traffic would hinder incoming items to enter the system (see Fig. 6 left side). Throughput would decrease.

An additional example shows how an intuitive routing strategy does not perform well. In a quadratic layout with items entering in the North and leaving in the South, the FlexConveyors at the Eastern and Western border of the system are naturally less used. If the conveying modules would try to route all items diagonal through the layout, the center modules would be frequently used (see Fig. 6 right side). Again, this would create congestion and waiting times. In addition to that, a high number of direction changes would increase the lead time. A better designed algorithm would transport items at the border of the system as long as it does not conflict with the criteria of the shortest path.

In order to achieve high throughput the routing should fulfill the three following requirements: on one hand the modules should be used uniformly all-over the network. On the other hand the routes should be shortest paths and they should have the less direction changes as possible. To optimize routing decisions with network analysis, node-specific measures have to be used; the location of the FlexConveyor module and the destination will have an effect in its decisions. To use such criteria for routing decisions, node-specific or dynamic indices have to be defined. Based on those indices, potential bottle necks of the system can be identified and either the layout or the algorithm can be adapted.

Different possibilities exist to parameterize the routing decisions in order to optimize system behavior. Because the FlexConveyor modules have limited information about the topology and the system state, we decided to implement a simple parameterization using the distance to destination. It takes all three requirements into account:

- To achieve uniform distribution stochastic decisions are introduced that lead to direction changes at all modules. The routing decision is optimized by determining the probability with the distance to destination: The closer the destination, the higher the probability of a direction change.
- To minimize average path length, during the first attempts of routing reservation, the modules only accept shortest paths. If a certain number of attempts are unsuccessful, longer routes are accepted.
- To decrease the number of direction changes, modules prefer always to forward an item without direction change, but the preference decreases the closer to destination the module is.

B. Tandem vs. single item movement

In the existing FlexConveyor algorithm, a module always has to be completely empty before receiving another item. Especially in systems with high work in process, this leads to waiting times and start-stop operation. With tandem movement, items that move in the same direction are transported simultaneously. The number of required cycles for the same movement is reduced.

C. Deadlock Handling

Mayer et al. [12] describe a deadlock prevention process for the FlexConveyor. Before each movement of an item, the receiving conveyor checks the occurrence of deadlocks and only allows movement if there is no risk. Cross-deadlocks (see Fig. 7) cannot be prevented within this process; they are either excluded by limiting the topology or the reservation process.

In densely connected sorting layouts, the deadlock prevention is more complex: There are not only few possible circles defined by the topology as before. But there are numerous possible circles within the network overlapping each other in every way. With changing reservations, the circles relevant for deadlocks change dynamically. Limiting the reservation process would lead to unacceptable throughput decrease.

Also, the tandem movement demands fast deadlock check in order to accept an item while another items is leaving. The enhanced deadlock prevention algorithm developed for closely connected networks also deals with cross-deadlocks, as modules with the corresponding reservation status act in a special way. But for high throughput, if several circles overlap each other, deadlocks still occur. Further enhancements are required.
VI. SORTING PERFORMANCE ANALYSIS

To compare the system performance in the different sorting layouts, discrete event simulation with constant work in process (WIP) has been implemented in AnyLogic. We investigate the effects of the number of items in the system (WIP) and the influence of tandem/single item movement. The following conditions are used:

- When an item is put in the system, the destination is randomly assigned. All exit conveyors are assigned with the same probability as destination.
- Once an item leaves the system, a new item is put into the system at a free input conveyor which is randomly chosen. If no input conveyor is available, the item has to wait until one gets free.

The decentralized control rules were embedded into the corresponding number of (software) conveyor objects communicating with their neighbors and taking decisions. Each simulation run executed for 10,000 cycles plus a warm up period of 200 cycles. Results in the plots reflect 10 replications of each run.

In the following, we will first discuss the throughput results with and without tandem movement. In a second step, we will draw conclusions from the occupancy of the conveying modules. Fig. 8 shows system throughput per cycle for varying WIP and different layouts.

The square layout shows the highest throughput compared with the other layouts for same WIP. The reason therefore is the low average shortest path. The trapezoid shows similar behavior, but with lower throughput. The square and trapezoid cannot handle higher WIP than 11 because of congestion and deadlock.

As the circle consists of more conveying modules, the average path is longer and the throughput lies beneath the one of square layout. For high throughputs, more space means less congestion. Therefore, the circle achieves the highest absolute throughput. In average, nearly 5 of the 8 available exists are used per cycle.

In the ring layout, the throughput always lies beneath the other layouts, because of longer average path lengths. The ring shows a special behavior depending on WIP: With low throughput, the shortest paths can be used, bidirectional traffic occurs. With high throughput, the ring behaves like a traffic circle; one conveying direction dominates which results in longer paths. Fig. 9 shows the increase of the average path length of the used routes for increasing WIP. Also, average waiting time is defined by the difference between lead time and path length (see Fig. 9). High waiting times result from direction changes due to longer routes and incoming/outgoing items hindering traffic.

In the ring layout, the only possible deadlock circle is given by the layout (also shown by a low alpha index) which can be prevented by the applied deadlock handling. Maximal throughput is reached for around 9 items in the system. For higher WIP, throughput decreases, because congestion increases. Single item movement leads to high waiting times. In this case, items should wait on the input conveyor to achieve higher throughput.

Fig. 10 shows system throughput for varying WIP like Fig. 8, except that tandem movement is used instead of single item movement. Basically, the layouts show the same behavior. But all layouts reach higher throughput with tandem movement than with single item movement, because fewer cycles are required for the same movement.
Fig. 10 also shows that the throughput in the ring layout does not decrease for high WIP, but it seems to reach a boundary value. The tandem movement allows efficient movement in the ring, independently of the cell occupancy.

As explained in section V, a uniform usage of all conveying modules is a desirable characteristic to achieve high throughput. Fig. 11 shows the average occupancy of the four layouts for WIP equal 10 and with tandem movement to give an insight how well the algorithm uses the resources of the network. The gray values represent the percentage of time that the corresponding conveying module has been occupied.

Densely connected networks seem to be suitable for sorting, because they increase capacity per used space and they also increase flexibility concerning location of entries and exits. Networks with a uniform accessibility of all modules and a short average path length lead to a good system performance. To compete against current high capacity sorting systems, hardware characteristics of the decentralized controlled sorting systems have to be analyzed and optimized. Throughput is strongly influenced by cycle time, thus conveying velocity.

Parameterization of local decision making based on topological information is a promising method to optimize decentralized algorithms. The parameterization of the presented algorithm has to be further examined. The influence of parameters like costs of direction changes, timeouts and others is not fully explained. In future, other possibilities for parameterization with the actual usage of modules and their accessibility will be analyzed. The deadlock handling has to be enhanced and the deadlock prevention proven for all occurring cases.

Future research will deal with new approaches of designing the algorithm. A step-wise reservation could improve usage and facilitate deadlock handling. Also the development of omnidirectional conveying technology will require an adaption of the used algorithms.

REFERENCES