A generic approach for the graph-based integrated production and intermodal transport scheduling with capacity restrictions

Bernd Scholz-Reiter\textsuperscript{a}, Jens Hartmann\textsuperscript{b},*, Thomas Makuschewitz\textsuperscript{b}, Enzo Morosini Frazzon\textsuperscript{c}

\textsuperscript{a}University of Bremen, Bibliothekstraße 1, 28359, Bremen, Germany
\textsuperscript{b}BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Hochschulring 20, 28359 Bremen, Germany
\textsuperscript{c}Industrial and Systems Engineering Department, Federal University of Santa Catarina, Campus UPSC, 88040-970, Florianópolis, SC, Brazil

* Corresponding author. Tel.: +49-421-218-50112; fax: +49-421-218-50003. E-mail address: hmn@biba.uni-bremen.de.

Abstract

The performance of global manufacturing supply chains depends on the interaction of production and transport processes. Currently, the scheduling of these processes is done separately without considering mutual requirements, which leads to non-optimal solutions. An integrated scheduling of both processes enables the improvement of supply chain performance. The integrated production and transport scheduling problem (PTSP) is NP-hard, so that heuristic methods are necessary to efficiently solve large problem instances as in the case of global manufacturing supply chains. This paper presents a heuristic scheduling approach which handles the integration of flexible production processes with intermodal transport, incorporating flexible land transport and maritime transport running a given timetable. The method is based on a graph that allows a reformulation of the PTSP as a shortest path problem for each job, which can be solved in polynomial time. The proposed method is applied to a supply chain scenario with a manufacturing facility in Brazil and shipments to customers in Germany. The obtained results show that the approach is suitable for the scheduling of large-scale problems and can be flexibly adapted to different scenarios.

Keywords: Production and Transport Scheduling Problem; Graph based scheduling; Integrated scheduling

1. Introduction

The integrated scheduling of production and intermodal transport operations in global manufacturing supply chains challenges both practitioners and researchers. Models to better understand and evaluate this complex problem are being developed and studied, yet a sufficiently comprehensive and adaptable method is still missing. This paper contributes to this development by proposing and demonstrating the applicability of a graph based approach. The method turns the integrated production and intermodal transport scheduling problem into a set of shortest path problems. In order to improve decision making in dynamic and competitive global environments, the use of resources in logistics systems must be better considered in control systems [1-3]. In fact, due to growing globalization, planning and programming of intermodal transport systems are becoming increasingly relevant [4]. There has been a growing research interest in scheduling problems, especially for transport and production planning [5-6]. Different modeling paradigms can be successfully used to describe a supply chain, and better conclusions can be drawn from the comparison between these paradigms [7]. Currently, the scheduling of production and transport processes in manufacturing supply chains is done separately. However, these processes are in interdependency so that an integrated consideration of information and material flows can enable improvements in the overall supply chain performance [8-9]. Several approaches for the integrated scheduling of a supply chain, involving production and intermodal transport scheduling have been proposed [8, 10]. The integration and synchronization of production and distribution scheduling was addressed in literature for various supply chain scenarios [11-13]. Nevertheless, most of the proposed approaches focus on the tactical decision level of supply chains. However, an efficient operation of a supply chain also requires
planning methods for the operational level, which includes the assignment of jobs to specific resources such as machines or transport devices. For practical applications an intuitive approach is preferable, in order to generate comprehensible decisions and to achieve a high acceptance by the executing personnel. Therefore, the use of graph theory is a promising approach for dealing with integrated scheduling problems [14].

This paper presents a heuristic scheduling approach which can solve problems that combine flexible production with intermodal transport processes. The intermodal transport incorporates flexible land transport and maritime transport running a given timetable. The heuristic method is based on a graph of all possible production and transport operations. This allows for a reformulation of the PTSP as a shortest path problem for each job, which can be solved in polynomial time. The paper presents a generic construction scheme for the graph that models production and transport operations in the same way. The nodes of the graph are time dependent and represent the beginning and end of operations. The edges represent the operations by themselves. They are capacity and cost weighted. A schedule can be computed for a set of jobs, where each job has to be delivered to a certain destination before a specific due date. In addition, each job has specific capacity requirements and can only be processed by resources, i.e. edges, that offer enough capacity. The complete schedule is determined in an iterative approach. In each iteration step, the optimal path for one of the jobs is computed as follows. After the choice of a specific job, the graph is temporarily reduced by all the edges that cannot provide enough capacity. This way, the resulting subgraph contains only feasible edges for the job and the optimal path can be found by solving a shortest path problem. The proposed method is applied to a supply chain scenario with a manufacturing facility in Brazil and shipments to customers in Germany, including land and maritime transport.

The paper is organized as follows. Section 2 describes a method to represent manufacturing and intermodal transport operations of supply chains as a graph. Here, a heuristic algorithm is presented that uses the graph to transform the PTSP into a set of shortest path problems. It is also shown how exact solutions can be computed as a benchmark for small problem instances from a mixed-integer program formulation of the PTSP. Section 3 presents a test case comprising production processes and intermodal transport, incorporating flexible land transport and maritime transport running a given timetable. Both, heuristic and exact method, are applied to the test case with several scenarios of jobs and the results are compared. Finally, Section 4 draws conclusions of the presented approach and states aspects that future research should address.

2. Graph based heuristic

The scheduling of supply chain operations is a wide field and comprises many different settings. One building block of supply chains that operate on a global scale is the usage of intermodal transport operations between two manufacturing facilities. This includes road transport by trucks, usually as an initial and final stage, as well as maritime transport by ships in order to bridge intercontinental distances. A major difference between these two modes of transportation is the availability. The routing of a fleet of trucks can be done flexibly in regard to the demands of the production facilities. The capacities are relatively low but in case of a high demand, additional trucks can be provided. The maritime transport, in contrast, offers high capacities and is running a fixed timetable. In the following, a generic approach will be developed that represents both transport modes as identical structures within a single graph and performs an integrated scheduling of production and intermodal transport operations.

2.1. Building the graph

A graph $G=(V,E)$ consists of a set of nodes $V$ and a set $E$ of edges, that each represent a connection between a pair of nodes. In this approach, the nodes are located in the x/y-plane, where the x-axis represents time $t$ and the y-axis locations. A location $i$ can either be a physical place along the transport process, such as a port and the final customer, or an imaginary storage level between two consecutive manufacturing steps. Thus, a node $v$ of the graph can be expressed as a pair $(i, t_v)$ of a location $i$ and a point in time $t_v$. The time is discretized, e.g. into shifts of 8 hours. Figure 1 shows a basic example of a graph with one location. Following the path from the left to the right side represents storage at location $i$ for 3 shifts.

Each location along the supply chain is represented as a location on the vertical axis. The transport of cargo is the movement from one location at a certain point in time to the consecutive location at another point in time. Thus, the analogue representation in the graph is the connection of two nodes on different vertical levels. In order to distinguish several connections between the same nodes, e.g. parallel machines, the connection contains another node $r$ representing the processing resource as in Figure 2. All connections between a pair of nodes form the set of edges $E$. Each edge has an
assigned capacity and costs that are determined by the characteristics of the resource that an edge represents. Some logistic operations require a handling time beforehand or afterwards, e.g., the loading or unloading of a ship at the port or set-up times of machines. In the graph, these times are represented as a separate pre- and post-processing level for each location that can be used if necessary.

With the building blocks of the graph, described in the previous section, production and transport systems can be modeled. The integrated production and transport scheduling problem (PTSP) consists of finding an optimal schedule that assigns a set of jobs with specific release dates and due dates, capacity requirements and specific manufacturing and transport costs to the resources of the modeled system. The objective value of the optimization is the total cost that the schedule produces, including costs for storage between consecutive operations and penalty costs for early delivery. The production facility $i$, where a job $j$ is processed first and its supply date $t_s$ determine the source node $(t_s, i_s)$ for this job. The node $(t_d, i_d)$ with its destination $i_d$ and the due date $t_d$ form a sink where job $j$ leaves the system. Between those two nodes, a path within the graph has to be found, that represents the optimal way of job $j$ through all the necessary production and transportation steps and that respects the capacity restrictions of the edges. In other words, within the subgraph of all feasible edges, i.e., edges that offer sufficient capacity, the cheapest path for a job has to be found. With the costs interpreted as distances, this problem can be solved in polynomial time by the Dijkstra algorithm [15]. The complexity class depends on the data structure used for implementation. The implementation used for this paper results in $O(n^2 + m)$ where $n$ is the number of nodes and $m$ is the number of edges of the graph. A method to use the described setting for the integrated scheduling of production and transport systems with a given timetable for the maritime transport is shown in Algorithm 1. First, every location $i$ of the system is applied to the vertical axis and nodes for each of the discretized points in time are created, which are regarded as decision points. All nodes at the same location are connected via edges in chronological order (red edges in Figure 2). Then, all relevant vessels, i.e., those which start and arrive between the earliest job release date and the latest due date, are turned into resources consisting of two edges and one node, connecting the pre- and post-processing level of the ports of origin and destination, respectively. These nodes and edges are a consequence of the given vessel timetable and form the skeleton of the graph. Handling times are added, connecting the resource with the closest previous and subsequent decision point. If a vessel arrives at a port which makes land transport to the subsequent location possible, another resource is added after the vessel’s handling process, representing transport by truck. This way, the transport between port of origin and destination of a job can be modeled. Analogous to the transport, the manufacturing processes are added to the graph. Therefore, whenever a machine could be used during a shift, the storage level of this machine is connected to the consecutive one by a resource with the capacity and costs specifications of this machine.

**Algorithm 1: Graph based scheduling**

- **Initialize $Z = [\min[i_s], \max[i_d]]$** within all jobs as relevant time period for scheduling;
- **Initialize set $I$ of all locations $i$ within the system;**
- **Initialize empty sets $V$ and $E$ for nodes and edges;**

**Building graph $G$:**

```plaintext
for (all locations $i$) {
    add $(t_{\text{start}}, i)$ to $V$ with $t_{\text{start}} \in Z$ starting time of a shift;
    add edges $(t_{\text{start}}, i)$ to $(t_{\text{end}}, i)$ to $E$ with $t_{\text{end}} < t_{\text{start}} + Z$;
}

for (all vessels/machines $r$ with $t_{\text{start}} \in Z$ and $t_{\text{end}} \in Z$) {
    add $(t_{\text{start}}, r)$ and $(t_{\text{end}})$ to $V$;
    add resource to $V$ and connecting edges to $E$ (including cost/capacity information);
    $(t_{\text{start}} + \text{handling}, r)$ and $(t_{\text{end}} + \text{handling})$ to $V$ and connect to resource and storage level $i$;
    if $(\exists$ subsequent land transport $) \{
        add land transport from $i$ to $i_{\text{dest}}$;
    $\}$
}
```

**Scheduling:**

```plaintext
for (all jobs $j$ with $(t(j)) \leq (t(j)) \leq (t(j)) \leq \ldots$) {
    initialize a copy $G'$ of $G$;
    delete all edges $e$ from $G'$ with $c_j > c_e$;
    find shortest path $(t(j), i(j))$ to $(t(j), i_{\text{dest}})$ in $G'$;
    reduce capacity of the chosen path in $G$;
    if (no path from $(t(j), i(j))$ to $(t(j), i_{\text{dest}})$ in $G'$) $\{
        assign $j$ to direct link between both nodes;
    $\}
}
```

---

![Resource connecting two locations with handling times](image-url)
After the graph $G$ was built as described above it can be used for the scheduling, i.e., finding the shortest path for each job. The paths are computed sequentially, where the sequence of jobs is determined by a dispatching rule, e.g., the job with the earliest due date is scheduled first. The graph contains edges with different capacities so that some edges might not offer enough capacity to meet the requirements of a job. In order to find a feasible path for a job $j$ with a required capacity of $c_j$ the graph $G$ is reduced to a subgraph $G'$ containing only the edges $e$ with offered capacity $c_e > c_j$. This way, the subgraph $G'$ contains only feasible edges for job $j$ so that for no path in $G'$ the capacity restrictions would be violated and the shortest path is the optimal path for $j$. Once a job is assigned to its optimal path the capacity of the edges on this path has to be reduced for all subsequent jobs. This reduction is done in the overall graph $G$ and is passed to all subsequent subgraphs that are derived from $G$. If a subgraph $G'$ does not contain any feasible path for the corresponding job $j$, the origin and destination node are connected directly by an additional edge, which represents external processing.

2.3. Computing exact solutions

In contrast to the heuristic method solving the shortest path problems sequentially, an optimal solution might be derived by mathematical programming. Therefore, the problem is formulated as a mixed-integer program (MIP). The results in Section 3 will show that this method has major drawbacks for practical application. However, the optimal solutions can be used as a benchmark for the solution quality of the heuristic solutions computed by Algorithm 1. The MIP formulation uses the following nomenclature. The jobs are denoted as $j \in J$ and the nodes of the graph as $v,v',h \in V$. The set of nodes that share an edge with a node $v$ is denoted as $V(v)$. The type, origin and destination of a job $j$ are given by $k(j), v_o(j)$ and $v_d(j)$, respectively. The costs for processing a job of type $k$ are $c_{ek}(k,v,v')$ between two nodes $v$ and $v'$. In case of external processing it is determined by $c_{ext}(k)$.

The following equalities and inequalities are the main restrictions describing the presented shortest path problem as a MIP. Each job has to be assigned to one continuous path between its origin and destination. Thus, for each intermediate node the job has to be assigned to exactly one arriving edge and one edge leaving the node. The assignment of a job to an edge is represented by a binary variable $Z$. If a job $j$ is assigned to the edge between nodes $v$ and $h$, variable $Z(j,v,h)$ equals one and otherwise zero. The condition is assured by equation (1).

$$
\sum_{v,v' \in V(k)} Z(j,v,h) + \sum_{v,v' \in V(k)} Z(j,h,v') = 1
$$

The earliest point in time when a job can arrive at the first node $v'$ after its supply node $v_s$ is its supply date at $v_s$ plus the processing time between $v_s$ and $v'$ (2).

$$
t_s(j) + t_{proc}(v,v') - M \cdot (1 - Z(j,v,v')) \leq t(j,v')
$$

In addition, the arrival time at all nodes $v'$ which are not a direct neighbor of the origin cannot be earlier than the arrival time at the previous node $v$ plus the processing time between $v$ and $v'$ (3).

$$
t(j,v') + t_{proc}(v,v') - M \cdot (1 - Z(j,v,v')) \leq t(j,v')
$$

A late arrival of internally processed jobs at their destination is not allowed (4).

$$
t(j,v_d) \geq t_d(j)
$$

In case of an early arrival at the final customer, the time until the due date is counted as storage time and produces costs which must not be ignored. The additional inequality (5) makes sure that each path ends exactly at the due date.

$$
t(j,v_d) \geq t_d(j)
$$

The accumulated assigned capacity of an edge must not exceed the available capacity (6).

$$
\sum_{j \in J} Z(j,v,v') \cdot c_{kap}(k(j),v,v') \leq c_{cap}(v,v')
$$

Finally, the objective function of the MIP formulation is the following. It minimizes the sum of the total processing costs and the costs for external processing. The variable $E(j)$ is also binary and equals one, if job $j$ is processed externally.

$$
\min \sum_{j \in J} \sum_{v \in V(k)} \sum_{v' \in V(k)} Z(j,v,v') \cdot c_{kap}(k(j),v,v') + \sum_{j \in J} E(j) \cdot c_{ext}(k(j))
$$

3. Experiments

The graph-based scheduling heuristic of Algorithm 1 was implemented in MATLAB R2012a and applied to several instances of a test case. Here, the scheduling of a supply chain with a manufacturer in Brazil and final customers in Germany, including land and maritime transport is considered. The computation of exact solutions is only possible for problem instances of a limited size. However, the solutions for the smaller problem instances could be computed with GAMS 23.6 and serve as a benchmark for the heuristic solution quality.
3.1. Test case

The test case is the supply scenario presented in Figure 3. It consists of a manufacturing site located in Campinas (Brazil) with nine machines and customers in Bremen and Kassel (Germany).

Fig. 3. Supply scenario including land and maritime transport

The production takes place in three consecutive machine levels so that a job has to pass one machine of each level. Jobs are classified in types \( k \) with specific capacity requirements and processing times and costs on each machine. Each job has a supply date, i.e., the earliest date where it can be considered for production, a due date and a destination. After production the goods are transported to two Brazilian ports by truck. From these two locations they can be transported either to Rotterdam or Hamburg in Europe by ship, where land transport to both possible customer locations is available.

The trucks are assumed to be available when needed, whereas the capacities of manufacturing facilities and the pre-scheduled maritime vessels are limited. The weekly timetable for ship transport is given in Figure 4. In order to keep the system feasible, e.g. in case of jobs exceeding the provided capacity of the production or transport system, production as well as transport can be done externally at high costs.

![Timetable for one week of ship transport](image)

Fig. 4. Timetable for one week of ship transport

3.2. Results

The implemented algorithm was applied to several scenarios of the test case with varying numbers of jobs between 50 and 400. The computation times grow with the number of jobs as well as with the size of the graph. Since the Dijkstra algorithm has to be repeated for each job, the complexity class of Algorithm 1 is \( O(j(n^2 + m)) \), where \( j \) is the total number of jobs. The linear dependency of jobs and computation times can be recognized in the data of Table 1.

Table 1. Comparison of heuristic and exact method

<table>
<thead>
<tr>
<th># jobs</th>
<th>Gams</th>
<th>Matlab</th>
<th>dd ( \uparrow )</th>
<th>dd ( \downarrow )</th>
<th>sd ( \uparrow )</th>
<th>sd ( \downarrow )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>2.94</td>
<td>6.83</td>
<td>0.00</td>
<td>0.00</td>
<td>0.07</td>
<td>0.00</td>
</tr>
<tr>
<td>100</td>
<td>5.25</td>
<td>23.72</td>
<td>1.90</td>
<td>2.27</td>
<td>1.06</td>
<td>0.46</td>
</tr>
<tr>
<td>200</td>
<td>12.62</td>
<td>131.19</td>
<td>3.31</td>
<td>2.61</td>
<td>0.56</td>
<td>0.30</td>
</tr>
<tr>
<td>250</td>
<td>17.46</td>
<td>119.09</td>
<td>2.97</td>
<td>2.20</td>
<td>27.42</td>
<td>1.13</td>
</tr>
<tr>
<td>300</td>
<td>28.73</td>
<td>441.64</td>
<td>1.86</td>
<td>7.60</td>
<td>1.32</td>
<td>0.81</td>
</tr>
</tbody>
</table>

The first three scenarios schedule very similar time periods so that the resulting graphs are very similar in size. The higher growth of computation time for the last two instances is due to a slightly longer time period for scheduling and thus a higher number of nodes and edges. In comparison, the times for computing exact solutions with GAMS are higher and grow much faster. Problem instances with 400 jobs or more could not be solved with GAMS, whereas the heuristic algorithm solved the 400 job instance in 34.66 seconds. Regarding computation time and critical problem size the heuristic approach clearly outperforms the exact method.

The solution quality of the heuristic approach depends on the sequence in which the jobs are chosen for scheduling. Four different ways of sorting the jobs were applied: by ascending and descending due dates (dd) as well as by ascending and descending supply dates (sd). The results in Table 1 show that the heuristic solutions only have a small deviation from the exact solutions; most of the solutions deviate less than 5%. However, the solution quality depends on the specific scenario, as it is highlighted in the plotted data in Figure 5. Sorting the jobs by ascending supply dates gives good results except for one scenario with more than 25% deviation from the optimal solution. The reason for this extreme value is that this constellation leads to the need of external processing for some jobs at very high costs, which can be avoided by the other sequences. For all sequences a scenario with an analogue situation can be generated. In regard to this and to the low computation times it is advisable to investigate different sequences in parallel and choose the best resulting solution. For the test scenarios, this would lead to a deviation of less than
1.2% in all cases in a computation time of less than 2 minutes.

Fig. 5. Accuracy of heuristic solutions with different priority rules

4. Conclusions

The paper presented a graph based heuristic approach for solving the integrated production and transport scheduling problem (PTSP), which consists of finding an optimal schedule for the processing of a given set of jobs with due date within a manufacturing supply chain. The originally NP-hard problem was transformed to a set of shortest path problems, one for each job. The operations of the supply chain were represented as generic building blocks for a graph which allows a flexible modeling of production and transport systems. It was shown how the set of shortest path problems can be formulated as a mixed-integer program (MIP), which allows for the computation of optimal solutions. However, due to the high problem complexity, large problem instances can only be solved heuristically. Thus, a heuristic method was developed and implemented that solves the shortest path problems sequentially. The computational analysis showed that this heuristic is capable of solving large problem instances in low computation time and is thus suitable for the scheduling of complex production and manufacturing scenarios. The solution quality was compared to the optimal solution where this was available. The results show that the heuristic solutions depend on the sequence of solving the shortest path problems but only have a small deviation from the optimal schedules if a combination of different sequences is chosen. This paper focused on explaining the principle of operation and demonstrating it by means of a small computational analysis. Future research should address a more extensive analysis also applying the algorithm to real world data sets. Special attention should be paid to the choice of priority rules for the sequence in which the jobs are chosen for scheduling. The applied rules should complement each other, so that for each scenario at least one of the rules generates a good solution.

Acknowledgements

This research was supported by CAPES, CNPq, FINEP and DFG as part of the Brazilian-German Collaborative Research Initiative on Manufacturing Technology (BRAGECRIM).

References