Integrating routing and scheduling for pipeless plants in different layouts

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: HUANG and CHUNG, 2005. Integrating routing and scheduling for pipeless plants in different layouts. Computers and chemical engineering, 29 (5), pp. 1069-1081

Additional Information:

- This article was published in the journal, Computers & chemical engineering [© Elsevier] and is also available at: http://www.sciencedirect.com/science/journal/00981354

Metadata Record: https://dspace.lboro.ac.uk/2134/2352

Publisher: © Elsevier

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Integrating Routing and Scheduling for Pipeless Plants in Different Layouts

W. Huang \(^a\), P.W.H. Chung \(^b,\)*

\(^a\) Department of Computing and Information Systems, University of Luton, Park Square, Luton, Bedfordshire, LU1 3JU, UK
\(^b\) Department of Computer Science, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK

Abstract

Pipeless batch plants are built to provide the ability to adapt to fast market changes. The scheduling of pipeless plants adds new challenges to the scheduling of conventional batch plants. This is because of the requirement to consider the allocation of resources and also the plant layout so that vessels can move from one processing station to another without conflict. This research investigates the integrated problem of scheduling and routing in pipeless plants and develops a constraint-based methodology suitable for different layouts. During the scheduling process, besides time and resource allocation, a planner is also called to find possible routes to move vessels from one processing station to the next proposed station. Each of the dynamically generated routes is immediately check for feasible track allocation so that any infeasible route is discarded as soon as possible to save computational time. A typical scheduling problem and results are described in detail to show how this integrated approach works.

Keywords: Scheduling, Routing, Planning, Integration, Constraint, Pipeless Plant

* Corresponding author. Email address: p.w.h.chung@lboro.ac.uk

Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081
1. Introduction

In the process industries chemical batch plants are attracting attention because of their suitability for the manufacturing of small-volume, high-value added products. Batch processing often involves multiple production activities such as mixing, blending and separating etc. Intermediate materials are transferred through different stages via a piping network in a traditional batch plant. In the past two decades, pipeless batch plants have been developed and built to increase plant flexibility to adapt to fast market changes (Zanetti, 1992; Niwa, 1993).

Pipeless plants are a kind of batch plants in which materials are transported from one processing stage to another in moveable vessels and processing takes place at a number of fixed processing stations. The same vessel is normally used to transfer and hold the material being processed at each station (Zanetti, 1992, Realff et al, 1996). Niwa (1993) draws an analogy between a pipeless batch plant and a chemical laboratory to make the concept easier to understand. In the laboratory, a beaker or flask is a “moveable vessel”, and the laboratory’s stationary equipment consists of a number of “processing stations”, such as weighing balances, mixers and Bunsen burners. To synthesize a product, the chemist generally uses a single flask, moving it to the appropriate processing station to carry out a specific operation.
Pipeless batch plants are already in use to manufacture products such as lubricants, oils, inks, and paints. The key advantage of pipeless batch plants is the flexibility with respect to moving materials using moveable vessels instead of a piping network. Without the maze of the piping network, pipeless batch plants permit a wide range of products to be handled with frequent changeover to meet market demands and opportunities. The flexibility has the potential of improving plant productivity, but presents a challenge in scheduling.

The scheduling of pipeless batch plants is a neglected area in the scheduling research for batch plants. Compared with the scheduling of traditional batch plants, the scheduling of pipeless plants is more challenging since it needs to consider the layout of a plant. The routes that are chosen for the moveable vessels influence significantly the transfer time of materials because they may result in waiting when more than one vessel is requiring the same track at the same time. In such cases alternative routes or alternative processing stations will need to be considered. Therefore route planning and production scheduling need to be taken into account simultaneously. One of the obvious dangers associated with moving vessels is the possibility of vessel collision. It is, therefore, essential to find conflict-free routes. However, current scheduling research associated with pipeless plants has not considered the issue of integrating routing and scheduling (Pantelides et al., 1995; Realff et al., 1996; Gonzalez and Realff, 1998; Lee et al., 2001).

This paper describes a constraint-based approach for scheduling pipeless plants that considers plant layout and route planning. In this study, a constraint model is proposed and an application system is developed to apply the model. This paper has
five sections. Section 2 gives a brief description of the scheduling and routing problems arising from pipeless plants and also differentiate them from other similar problems. Section 3 presents the developed model and system. Section 4 illustrates how a pipeless plant can be scheduled by using the proposed approach and discusses the results. Conclusions are drawn in Section 5.

2. Problem Definition

The production scheduling and route planning are two aspects that need to be addressed simultaneously within the scheduling of pipeless plants. The description of each aspect is given below.

2.1 Production Scheduling

The aim of production scheduling is to harmonize the entire plant operation to achieve the production goals and satisfy a variety of constraints such as deadlines, priority etc. These goals are normally related to producing a number of batches of products. Each batch production involves a series of activities such charging, reacting, blending, discharging, and cleaning, etc. in order to convert raw materials into final products. Production scheduling usually needs to allocate time slots for processing, waiting and transfer, and allocate resources such as materials, processing stations and moveable vessels.
2.2 Route Planning

A moveable vessel (MV) is used to transfer a batch of material to visit a number of stations to process the material into a batch of products. A route is made up of the consecutive tracks that a vessel takes to move from one station to another. Usually a number of moveable vessels are operated simultaneously and the routes moveable vessels will take need to be planned in relation to the plant layout to avoid vessel collision. The purpose of route planning is to find conflict-free routes for all moveable vessels to complete the schedule.

2.3 Relations between scheduling and routing

Producing a batch of product involves a number of activities and the scheduling process needs to allocate a suitable station to each activity. During the allocation process, the possible routes, by which a moveable vessel transfers material between two stations, need to be created as well. The feasibility of these possible routes must be checked to avoid vessels collision and other constraint violations.

The routing and scheduling problems arising in pipeless plants can be viewed as the problem of integrating planning and scheduling from the AI (Artificial Intelligence) point of view. For many years, there has been a gap between planning and scheduling research in AI, as planning and scheduling are treated as two separate processes to simplify the problem (Garrido and Barber, 2001). A simple approach is for the planer to determine the activity sequence and then the plan is used by the scheduler for time and resource allocation. If constraint violations are detected, the whole plan is discarded and the above processes are repeated until a feasible solution is obtained.

Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081
This approach is infeasible for complex problems such as the scheduling and route planning problems in pipeless plants. These problems involve a large number of resources to be allocated and a large number of possible routes to be checked. It is too time consuming to repeat the entire planning and scheduling processes again and again to reach a solution. The two processes needed to be integrated so as to detect conflicts as soon as possible to save computational time. Research in integrating planning and scheduling (IPS) is attracting increasing interest because of its immediate application to real problems (Garrido and Barber, 2001).

2.4 Similar Problems involving AGV

Intuitively, the scheduling and routing problems arising in pipeless batch plants are thought to be similar to those of AGV (automated guided vehicles) systems that are used to deliver loads to different machines in manufacturing plants, which are usually called AGV-served manufacturing plants. The scheduling and routing problems in these plants have been investigated for decades and many papers have been reported (Ganesharajah et al, 1998; Qiu et al, 2002; Lee and Chen, 2001; Hall et al, 2001). The operation of AGV systems also needs to take layouts into account and the adopted layouts include linear topology, loop topology and complex network topology (such as mesh topology). In order to make AGV scheduling and routing simple, the loop topology including single-loops and multi-loops, which only employs a few vehicles that move in the same direction within the loop, is often used. However, the system throughput may not be very high. Another way to avoid vehicle collision is to design special configurations such as segmented floor topology (SFT) and tandem configuration in which a single vehicle serves a non-overlapping segment or loop while buffers are placed between different segments or loops for load transfer.
(Ganesharajah et al, 1998; Qiu et al, 2002). Many researchers have reported their results on scheduling and routing of AGV-served manufacturing plants. However, due to the complexity of these problems, the proposed approaches can only solve either the scheduling or the routing problem and no published papers has been found to solve the integrated problem successfully (Qiu et al, 2002). Although there are similarities between AGV-served manufacturing plants and pipeless plants, there are differences between them. In manufacturing plants, AGVs are mostly used to deliver loads (e.g. workpiece) to different machines. After an AGV is unloaded, and while the workpiece is being machined, the AGV becomes idle and can be sent to pick up another load. When many machines compete for one AGV or a machine needs to select an AGV from many, the scheduling of AGVs becomes important (Ganesharajah, 1998). A simple rule that is often used is First-Come-First-Serve. In pipeless batch plant, moveable vessels (MVs) are usually used to transfer a batch of material from one station to another and processing takes places in these vessels at each station. Once a MV is allocated to a batch production process, it must visit a series of stations determined by the production process until the final product is produced. Therefore, in pipeless plants, production scheduling is closely tied in with MV allocation and routing. In AGV-served manufacturing, plants production scheduling, and AGV allocation and routing are often dealt with separately.

3. Constraint-Based Methodology

Constraint satisfaction techniques (CST) have been adopted in this research. By applying CST a scheduling problem is treated as a constraint satisfaction problem (CSP). An advantage of CST is the ease of problem formulation.
A CSP typically consists of a number of variables and a number of constraints on and among those variables. The solution of a CSP is defined as the assignment of values to the variables so that all imposed constraints are satisfied simultaneously. The set of all possible assignments of values to the variables is known as the search space. Another advantage of CST is that rather than searching blindly the entire space for a solution, they exploit the stated constraints to reduce the search space. Constraints are used in constructive ways to deduce other constraints and to detect inconsistencies among possible solutions. These techniques are called constraint propagation and consistency checking. Unsuitable values due to inconsistency in the domain of variables are removed.

A simple example is used here to illustrate how CST works. Suppose there are two variables “n” and “m”, and their domain of possible values are [0,1,2,…10] and [5,6,7] respectively. Two constraints “n<10” and “n+m=15” are imposed. When the constraint “n<10” is set, the value 10 is removed from the domain of “n” and it becomes [0,1,2…9]. When the constraint “n+m=15” is set, unsuitable values are removed from domains of “n” and “m” and they become [8,9] and [6,7] respectively. Therefore the search space has been greatly reduced before searching for solutions begin. To search for a solution, a choice point is set by assigning the value 6 or 7 to “m”. If the value 6 is assigned to “m” then constraint propagation will remove the value 8 from the domain “n” and bind the value 9 to “n”, a solution is found. The system can also backtrack to find the alternative solution “m”=7 and “n”=8. If the optimal criterion is to minimize “m” then the latter solution is discarded since “m” is larger in this case. From this simple example, it can be seen that finding a solution using CST includes two stages: 1) in the preprocessing stage constraint satisfaction...
techniques, such as consistency checking, are applied to reduce the search space; 2) in the search stage the search space is explored using constraint propagation and backtracking to find different solutions. The example also shows that the order in which the constraints are imposed is not very important from the correctness point of view as the obtained final solution is the same. However, the number of steps required to reach the same solution may differ.

Huang and Chung (1999) proposed a constraint model for production scheduling of traditional chemical batch plants, and it was later extended to cover pipeless plants without considering route planning (Huang and Chung, 2000). In this paper the constraint model is further extended. It takes into account of production scheduling, route planning, the integration of scheduling and routing. The proposed model can handle different plant layouts and it brings together many constraints and categorizes them according to their functions.

Producing a batch of products involves a number of related activities. The relations among different activities and resources impose many constraints on the production process. The proposed constraint model is defined to represent these relations. Variables representing production activity, batch activity, processing station, moveable vessel and material are defined as follows:

- **Production Activity** \( (J_i) \)

  A batch production may consist of one or more production activities. Each production activity transforms materials from their input states into their output states.

- **Batch Activity** \( (B_r) \)
A batch activity is defined to represent those related production activities within a batch production. A moveable vessel is selected by a batch activity to visit a series of processing stations to produce the product. The duration of a batch activity covers the processing time of all the related activities and the time a vessel needs to move from one station to another. For example, a batch activity may represent a number of production activities, Charge A, Charge B, Blend AB, Discharge P, and Clean the soiled vessel, which convert a batch of raw material into a batch of final products. If several batch productions are required to meet the quantity demand for a specific product, several batch activities are needed to represent them. The concept of a batch activity is necessary to prevent a vessel from being used by more than one batch simultaneously, i.e. once a vessel is selected to produce a batch of product then it cannot be used by another batch activity until the previous one finishes.

- **Moveable Vessel** ($V_x$)
  
  $V_x$ refers to a moveable vessel. A vessel is selected by a batch activity to visit a series of processing stations to produce a batch of products.

- **Processing Station** ($S_p$)
  
  A processing station is a working unit where the material in a vessel is converted from one state into another state.

- **Material** ($M_u$)
  
  $M_u$ refers to a kind of materials proceeded in a station.

Resource allocation is done by considering the constraints described in the next section. The constraints will be applied to decide on the specific start and end time of every activity and the occupied time of resources by these activities. Although the
constraints are presented in a particular order under different categories, during run time the constraints do not have to be imposed in any particular order. As long as all the constraints are satisfied then a feasible solution is found. The appropriate timing and ordering of activities are determined dynamically based on the imposed constraints during run time. The variable to be optimized in a schedule is the makespan, which represents the elapse time between the beginning of the first activity and the end of the last activity of the whole production process.

### 3.1 Production Scheduling

Let a schedule have $n$ activities, $J_1$ to $J_n$. $J_i$ represents a production activity and $J_j$ represents another one, where $i \in [1, n]$ and $j \in [1, n]$.

**Time-Bound Constraints**

There are several time-bound constraints in the model. The start time of the first production activity of the whole process is the time origin and the end time of any production activity cannot exceed the time horizon:

$$ST(J_i) = TO \quad \text{and} \quad ET(J_i) \leq TH$$

$ET$ and $ST$ represent the end time and the start time of an activity respectively. $TO$ and $TH$ represent the time origin and time horizon of the schedule respectively. $TO$ is a constant specified by the user and is constrained to be the start time of the schedule i.e. the start time of the first production activity, $J_1$. Which activity is scheduled to be the first one is decided dynamically during run time based on all the constraints. $TH$ is required to be set for every schedule. It can be a constant specified by the user, which represents the upper time bound. If a schedule is not required to be limited by a
specific upper time bound then it can be set to be the sum of the durations of all activities.

Sometimes, a product may be required to be delivered before a time point $T_d$. Any production activity, which is represented by $J_d$, within a batch production that produces the product is constrained to end before $T_d$ in order to meet this requirement.

\[ ET(J_d) \leq T_d \] (2)

**Precedence Constraints**

Under certain requirements, a production activity must precede another one, then

\[ ET(J_i) \leq ST(J_j) \] (3)

As mentioned, more than one batch production may be required to meet the amount demand of a product. No matter which batch production they belong to, the activities for producing the same product are exactly the same, so a precedence constraint is set to let a production activity within a batch production start after the start of the corresponding production activity within the previous batch production.

\[ ST(J_a) \leq ST(J_b) \] (4)

$J_a$ is an activity in a batch production and $J_b$ is the corresponding activity in the next batch production. This constraint makes $J_b$ start after $J_a$, but they do not have to be consecutive. The exact time allocation is decided during system run time. This constraint narrows the search space by reducing the needless possibilities (choice points) for a solution.

*Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081*
The start time of a batch activity must be equal to the start time of the first production activity within the batch production.

\[ ST(B_r) = ST(J_{r1}) \]  \hspace{1cm} (5)

**Constraints on Processing Station Allocation**

There are different types of processing stations available and different types of stations are suitable for different activities. A suitable station should be selected by an activity.

\[ J_i \leftarrow S_p \]  \hspace{1cm} (6)

Because a processing station also is a unary resource, any two activities requiring the same processing station cannot overlap, i.e. the end time of an activity must precede the start time of the other:

If \( S_p(J_i) = S_p(J_j) \) then \( ET(J_i) \leq ST(J_j) \) or \( ET(J_j) \leq ST(J_i) \) \hspace{1cm} (7)

Under some requirements such as maintenance, it is possible that a station is not available within a time window from \( T_x \) to \( T_y \). If the station is required by an activity, it must end before \( T_x \), or start after \( T_y \):

If \( J_i \leftarrow S_p \) and \( T_x < T_y \) then \( ET(J_i) \leq T_x \) or \( ST(J_i) \geq T_y \) \hspace{1cm} (8)

**Constraints on Moveable Vessel Allocation**

To produce a batch of product, a batch activity needs to require a suitable vessel for this batch production.

\[ B_r \leftarrow V_x \]  \hspace{1cm} (9)
Because a moveable vessel is a unary resource, any two batch activities requiring the same moveable vessel cannot overlap, i.e. the end time of a batch activity must precede the start time of another (\(B_s\) and \(B_r\) represents two different batch activities):

\[
\text{If } V_x(B_s) = V_x(B_r) \text{ then } ET(B_s) \leq ST(B_s) \text{ or } ET(B_r) \leq ST(B_r)
\]  

(10)

After a vessel is allocated to a batch activity, the vessel will be kept for that batch from the start time of the first production activity to the end time of the last production activity plus the time for the vessel to move from the last station to the start point of another batch production. The period is the duration of that batch activity.

\[
ET(B_r) - ST(B_r) = ET(J_m) - ST(J_{r1}) + T_r
\]  

(11)

\(J_m\) and \(J_{r1}\) represent the last and the first production activity respectively within a batch production. \(T_r\) represents the moving time that a vessel needs from the last station to the start point of another batch production.

**Constraints on Material Allocation**

Besides the processing stations, a production activity also requires material and converts it from one state to another state. The type and amounts of required material depend on the specific production demand.

\[
J_i \leftarrow M_u
\]  

(12)

Materials are discrete resources with certain quantity. If the available quantity is limited, two and more production activities requiring the same materials at time point
$T_u$ may overlap in time as long as their total requirement for the materials does not exceed the limited quantity:

If $ST(J_i) \leq T_u \leq ET(J_i)$ then \[ \sum_i Q(J_i) \leq Q(M_a) \] (13)

Where $Q(J_i)$ represents the quantity of material required by the $i$th activity at time point $T_u$ and \[ \sum_i Q(J_i) \] represents the total required quantity of material at time point $T_u$. $Q(M_a)$ represents the total available quantity of this material at time point $T_u$.

**Optimal Criterion**

The optimal criterion is set to minimize makespan.

$ET(B_t) \leq M_s$ and $Min(M_s)$ (14)

Where $B_t$ represents any batch activity in a schedule. $Min$ represents “Minimize” and $M_s$ represents makespan.

### 3.2 Plant Layouts

One of the novel aspects of this work is the consideration of the plant layout when scheduling a pipeless plant. The common layouts for pipeless batch plants are “herringbone”, “circular” and “linear”, which are illustrated in Figures 1 and 2. Usually the detailed geometry of the plant layout is determined by the shape of the available space and other considerations such as design and construction requirements that are outside the scope of scheduling.

In the proposed model, a plant layout is represented by a set of connected nodes, which are labeled A, B, C, D, E, F, G, H, I, J K, L, M, N, O and P in the above two figures. The nodes shown as rectangles are places where processing stations are
located. All nodes are connected by tracks and the length of every track is provided by the user. For example, the herringbone layout in Figure 1 has a total of 11 tracks, which are labeled $H_0$, $H_1$, $H_2$, $V_0$, $V_1$, $V_2$, $V_3$, $V_4$, $V_5$, $V_6$, and $V_7$. Besides the nodes representing the station locations, the other nodes represent the track connection points where buffer spaces, such as $B_1$, $B_2$, $B_3$ and $B_4$ in Figure 1, can be used as waiting spaces for MVs. The representation allows the detailed movements of vessels to be studied. Traveling vessels are assumed to move at a constant speed on all tracks so the length of a track can be represented in terms of the time that a vessel needs to pass through that track. The representation is sufficiently general to represent any plant layout and the number of nodes is not limited.
Figure 1: Herringbone layout of pipeless batch plants

Figure 2: Other layout examples of pipeless plants

Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081
3.3 Integrating Route Planning and Scheduling

A complete route is the whole journey, from the start station via the intermediate stations to the end station, which a moveable vessel needs to pass through so that a batch of raw materials can be converted into a batch of products. The journey between two stations, for two consecutive activities, as part of the complete route is called “a partial route” or simply “a route”. There maybe more than one possible partial route between the two stations. A complete route consists of a number of partial routes. As mentioned previously it is not sufficient to plan the route prior to scheduling but it needs to be planned dynamically along with the scheduling process during which stations are allocated to production activities. For example for two consecutive activities, such as blending and reacting, a particular blender and a particular reactor needs to be selected and scheduled, the route planner is then called to plan the partial route between the two stations, and then the scheduler will check its feasibility by allocating the tracks along the partial route. The route is feasible only if it does not cause any conflicts with the movement of other vessels. Once the scheduler has dynamically allocated suitable stations to every production activity within a batch production, and allocated tracks for conflict-free routes, which are dynamically planned, then a feasible solution is produced.

The route planner is developed based on representing a plant layout as a connectivity table. The approach can be applied to any plant layout. For example, the connectivity table for the herringbone layout shown in Figure 1 is given in Table 1.
Table 1: Connectivity table for a herringbone layout with 12 nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In a connectivity table, if two nodes are connected to each other, i.e. there is a track between them, then the value is one, otherwise it is zero. Given a layout represented by connected nodes and every station locating on a node, the possible routes between any two stations can be found using the connectivity table. Once a route is found, where the nodes do not appear more than once, the tracks in the route can be determined based on the nodes that make up the route. For example, in Figure 1, the route from A to F includes the nodes in the order of A, I, J, K and F. Therefore, the corresponding tracks are in the order of V₀, H₀, H₁ and V₅. A herringbone layout only has one possible route between two locations, but there is more than one possible route in other types of layouts such as a circular layout.

The pseudocode for the route planner based on depth-first search (Drozdek and Simon, 1995) is shown in Figure 3. When the route planner is called, the start node becomes the current node at the initial time and the intermediate nodes will be determined one by one towards the end node. In order to save computational time, the route planner does not search for all possible routes between two stations in one go, which may consume a large amount of time. Instead, it generates one route at a time.
and save the related data so that it can continue to search for another route if necessary. The feasibility of each generated route is checked by the scheduler by imposing the related constraints such as those on track allocation. If the route is found to be feasible then it will be accepted and other possible routes are not explored. However, if a conflict is detected then the route is discarded and the system backtracks to trigger the route planner to search for another route. In the worst case, every possible route may have to be checked to find a feasible one, or to confirm that no feasible route is available. In the latter case the system will backtrack and find alternative stations for the production activities.
While there is a current node \( (N_i) \) in the array //an outer loop
For every node within the plant layout //an inner loop
If there is a connected node \( (N_{i+1}) \) to the current node, then
  If the node was already used in the array, then
    The node is not suitable. //continue the inner loop
  Else if it was tried before, then
    The node is not suitable. //continue the inner loop
  Else
    Mark it as a tried node.
    Put it into the array and set it as the current node.
    Break the inner loop.
End if
End if
End loop

If a suitable node \( (N_{i+1}) \) is found and set as the current node \( (N_i) \), then
If it is the end node, then //a route is found
  Find the tracks and buffers included within the route.
  The scheduler is called to impose related constraints such as track allocation to check the route’s feasibility. If a conflict is detected, the end node is removed and the route planner is called again to search for another route.
Else //a suitable node found but a route not found yet
  Continue the outer loop.
End if
Else //no suitable node found
  Remove the current node \( (N_i) \) from the array.
  If the array is empty, then //no route found any more
    The route planner is terminated.
  Else
    Set the last node \( (N_{i-1}) \) as the current node \( (N_i) \) in the array.
End if
End loop

Figure 3: The pseudocode for route planner

Several kinds of scheduling constraints are proposed to check the feasibility of a route generated by the route planner. These constraints are route dependent and are created and added to the system dynamically within the scheduling process. This dynamic feature is significant since only dynamic constraint additions can achieve the integration of scheduling and route planning in pipeless batch plants. The integrated problem cannot be solved by defining a complete specification of the whole problem...
before searching for a solution because a possible route by which a moveable vessel can transfer material between two consecutive production activities is generated on the fly. Since a production activity may have more than one suitable station to select from and the selection of stations for the two consecutive production activities is carried out during search time, the route planner is called to establish routes between the two selected stations during the search. After a possible route is found, scheduling constraints are imposed dynamically to check its feasibility. As soon as a route is found infeasible, the system backtracks and all the constraints added dynamically based on that route selection are removed. The route planner will then be called again to generate another route. The system repeatedly interleaves the execution of the scheduler and the route planner until a feasible route is found. If no feasible route is found then the scheduler will backtrack and redo the station selection process for the two consecutive production activities. A constraint-based system is developed to allow dynamic addition and deletion of constraints during run time making it possible to solve the integrated problem of scheduling and route planning in pipeless batch plants.

One type of constraint that is added dynamically is the transfer time constraint that determines the temporal relationship of two consecutive production activities that require the two stations. Let the two consecutive activities be \( J_i \) and \( J_{i+1} \). The temporal constraint on them, taking the generated route into account, can be obtained by the following steps. A suitable station is selected for each activity by the scheduler, say, \( J_i \leftarrow S_p \) and \( J_{i+1} \leftarrow S_q \), and then the locations that the two stations occupy, e.g. location \( a \) and \( b \), are detected. The route planner is now called to find a route between them. The transfer time, which a moveable vessel needs to pass through the route between \( a \) and
\[ b, \text{ i.e. } R_{ab}, \text{ can be calculated based on the length of all tracks along the route.} \]

Considering there are buffers at the track connection points where moveable vessels may stay temporarily, the temporal constraint can be described by the formula:

\[ ET(J_i) + R_{ab} \leq ST(J_{i+1}) \]  \hspace{1cm} (15)

As mentioned above, usually more than one vessel is in operation in a pipeless batch plant and different moveable vessels may have different routes. It is possible that more than one moveable vessel will require the same track at the same time and they move in opposite directions. In order to ensure that a generated route is free of conflict, constraints are necessary to avoid having more than one vessel going through a track simultaneously. Therefore, a track is treated as a unary resource that is allocated to different vessels over time. To achieve the goal, a new type of activity called moving activity is proposed. Since a track is a unary resource, after it is allocated to a moving activity i.e. used by a vessel, this track cannot be used by another moveable vessel during the same period. A route normally includes more than one track; therefore moving activities are required for all tracks along the route. The track allocation constraints are presented as follows:

\[ MA_a \leftarrow T_b \hspace{1cm} T_b \in K \]  \hspace{1cm} (16)

\( MA_a \) and \( T_b \) represent a moving activity and a track respectively. The above formula means that when a moving activity is created it requires a track within a route. \( K \) represents a domain that includes all the tracks in the route. \( “T_b \in K” \) means \( T_b \) is a track in the route.

\[ ET(J_i) \leq ST(MA_1) \]
\[ ET(MA_a) \leq ST(MA_{a+1}) \quad a=1.2\ldots k-1 \] (17)

After a production activity \( J_i \) finishes at a station, the moveable vessel will move to the downstream station. The above two formulae mean that moving activities are created for the tracks that make up the route that the vessel will take to the next station. \( MA_1 \) represents the first created moving activity requiring the first track and \( MA_k \) represents the last one. A moving activity except the first one was constrained to start after the end time of its previous one. The total number of tracks in the route is \( k \).

If \( J_i \) is not the last production activity within a batch production, a constraint is set as

If \( J_i \neq J_m \) then \( ET(MA_k) \leq ST(J_{i+1}) \) (18)

If \( J_i \) is the last production activity within a batch production, a constraint is set as

If \( J_i = J_m \) then \( ET(MA_k) = ET(B_r) \) (19)

The above formulae mean that if the upstream production activity is not the last activity within a batch production, the next production activity \( J_{i+1} \), which takes place in the downstream station, will start after the end of the last moving activity in the route. If \( J_i \) is the last activity within a batch production, the moveable vessel needs to move back to its start point, so the end time of the last moving activity must be equal to the end time of the current batch activity.

Besides the track allocation, the buffer allocation constraints are also imposed. Usually buffers, such as \( B_1, B_2, B_3, \) and \( B_4 \) in Figure 1, are set at the track connection points. Since tracks are treated as unary resources allocated over time, it is possible
that a moveable vessel might have to stay in a buffer space temporarily until the track it is going to pass is available for conflict-free traveling. How many vessels can stay in a buffer area simultaneously depends on the size of the vessels and the buffer areas. Therefore, two constraints are introduced:

\[ MA_a \leftarrow Buf_a \quad \text{and} \quad \sum_a Buf_a \leq Buf_{\text{max}} \]  

(20)

Where a buffer is treated as a resource and is allocated to a moving activity. \( Buf_{\text{max}} \) is the capacity of a buffer, which represents the total number of vessels allowed to stay in the buffer simultaneously. The user can specify the buffer capacity.

The proposed constraint-based model for scheduling is implemented in C++ using ILOG (ILOG, 1999), which provides a library of object classes and methods that support CSTs such as constraint propagation, consistency checking and backtracking (Ginsberg, 1993; Tsang, 1995). The route planner is also developed in C++ as part of the system so that the scheduler and route planner work in an integrated fashion as described previously.

4. Example and Discussion

The application system can be applied to solve scheduling problems for pipeless plants in different layouts. As an example, a typical scheduling problem for pipeless plants (Huang and Chung, 2000) is used and extended here to consider route planning. The herringbone, linear and circular layouts are investigated to demonstrate the feasibility of the developed methodology.
4.1 Problem Description

The process recipe for manufacturing a variety of products in a pipeless plant is given in the form of a State-Task Network (STN) (Kondili et al., 1993). For pipeless batch plants, different states represent different types of materials and/or moveable vessels in different conditions (e.g. “Blended AB in vessel”, “Empty dirty vessel”, and “Empty clean vessel”). Production activity (i.e. task) represents a transformation from one set of states into another. The production process of a scheduling problem is shown in Figure 4. Circles and rectangles denote states and production activities respectively.

![Figure 4: Production processes of a pipeless batch plant](image-url)

The process starts with clean vessels being charged with the appropriate amount of A and B. The charged vessels are then taken to a blender, where the content is homogenized in a blending operation to form material AB. Following this, AB reacts...
with a third raw material C to form another intermediate material (Int.). Three final products P1, P2 and P3 are produced by blending intermediate materials with three different additives A1, A2 and A3 respectively. The corresponding products P1, P2 and P3 are discharged through a discharge station and the empty vessels are cleaned before they can be used again. Finally, the clean vessels are moved back to the start point, i.e. in front of the station where A is to be charged. It can be seen from the production process that each batch production involves seven production activities: Charge A, Charge B, Blend A with B, Reaction, Blend Intermediate Material (Int.) with an Additive, Discharge the Product and Clean vessels. The duration and the suitable stations for each production activity are shown in Table 2. In this example eight processing stations of six distinct types are provided.

<table>
<thead>
<tr>
<th>Production Activity</th>
<th>Duration (Hour)</th>
<th>Suitable Station</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge A</td>
<td>0.5</td>
<td>Charger 1</td>
</tr>
<tr>
<td>Charge B</td>
<td>0.5</td>
<td>Charger 2</td>
</tr>
<tr>
<td>Blend A+B</td>
<td>0.8</td>
<td>Blender 1 Blender 2</td>
</tr>
<tr>
<td>Reaction</td>
<td>0.8</td>
<td>Reactor 1 Reactor 2</td>
</tr>
<tr>
<td>Blend Additives + Int.</td>
<td>0.5</td>
<td>Blender 1 Blender 2</td>
</tr>
<tr>
<td>Discharge</td>
<td>0.5</td>
<td>Discharger</td>
</tr>
<tr>
<td>Cleaning</td>
<td>0.5</td>
<td>Cleaner</td>
</tr>
</tbody>
</table>

Moveable vessels are also needed in pipeless batch plants. In this example, there are three vessels available: Vessel 1, Vessel 2 and Vessel 3, each with 10m$^3$ capacity. In this example Vessel 3 is only used to produce product P2, but Vessel 1 and Vessel 2 can be used to produce either P1 or P3. This limitation is introduced into the problem to test the circumstance that a special vessel is needed for a particular product or activity. For example, if the final product P2 is corrosive, a special glass-lined vessel

*Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081*
such as Vessel 3 might be necessary for holding materials undergoing production activities Blend A2 + Int. and Discharge P2, while cheaper moveable vessels such as Vessel 1 and Vessel 2 might be adequate for materials undergoing Blend A1 + Int. and Blend A3 + Int. etc. Since there is normally no provision for transferring material directly between moveable vessels, glass-lined vessels would also have to be used for the upstream activities of Blend A2 + Int. even if it was not strictly necessary. It is assumed that all vessels are initially empty and clean, and lined up in front of the first charging station. The production demand is 20m$^3$ of P1, 20m$^3$ of P2, and 10m$^3$ of P3. Since the vessel capacity is 10m$^3$, five batches are required. The time horizon of the schedule is 48 hours and the optimal criterion is to minimize the makespan.

**4.2 Herringbone Layout**

In pipeless batch plants, plant layout is a very important factor that affects the scheduling results. In this section, the herringbone layout shown in Figure 1 is used for the example. The connectivity relations between the nodes representing the adopted herringbone layout have been shown in Table 1. Circular and linear layouts will be considered in the next sub-section. It is understood that for any layout the available stations can be placed in different locations. Where these stations should be located is determined by the shape of the available space and/or other geometry considerations, e.g. design and construction factors, that are outside the scope of scheduling, so the station locations are assumed to be known in advance. Figure 5 shows a herringbone layout with the stations placed in a symmetry style, where the different stations are distributed evenly over the layout structure. Linking Figure 1 with Figure 5, the relations between a station and the node it occupies are:
The moveable vessels are assumed to move on “tracks”, whether these are rails or wires buried under the floor, and thus cannot pass each other except at designated buffer areas. It is also assumed that moveable vessels carrying material to be processed at a processing station may wait in a space just before that station. Similarly, a vessel loaded with material that has just undergone processing may wait in a space immediately after that station.

Figure 5: Scheduling of a pipeless batch plant in herringbone layout

In this example, buffers such as B₁, B₂, B₃ and B₄ are placed at junctions where tracks intersect. Vessels are only allowed to pass each other at these buffers. Vessels can also stay at these buffers to wait for a track to become available. The data on tracks and buffers are provided in Table 3 and Table 4 respectively. The track length is given.
in terms of the time that a vessel needs to travel the whole track. In this example one unit of time is set as 0.1 hour. Buffer capacity indicates the total number of vessels that can stay at the buffer simultaneously.

**Table 3: Track data for a herringbone layout with 12 nodes**

<table>
<thead>
<tr>
<th>Track Name</th>
<th>First Node</th>
<th>Second Node</th>
<th>Track Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0</td>
<td>A</td>
<td>I</td>
<td>1 unit</td>
</tr>
<tr>
<td>V1</td>
<td>I</td>
<td>B</td>
<td>1 unit</td>
</tr>
<tr>
<td>V2</td>
<td>C</td>
<td>J</td>
<td>1 unit</td>
</tr>
<tr>
<td>V3</td>
<td>J</td>
<td>D</td>
<td>1 unit</td>
</tr>
<tr>
<td>V4</td>
<td>E</td>
<td>K</td>
<td>1 unit</td>
</tr>
<tr>
<td>V5</td>
<td>K</td>
<td>F</td>
<td>1 unit</td>
</tr>
<tr>
<td>V6</td>
<td>G</td>
<td>L</td>
<td>1 unit</td>
</tr>
<tr>
<td>V7</td>
<td>L</td>
<td>H</td>
<td>1 unit</td>
</tr>
<tr>
<td>H0</td>
<td>I</td>
<td>J</td>
<td>1 unit</td>
</tr>
<tr>
<td>H1</td>
<td>J</td>
<td>K</td>
<td>1 unit</td>
</tr>
<tr>
<td>H2</td>
<td>K</td>
<td>L</td>
<td>1 unit</td>
</tr>
</tbody>
</table>

**Table 4: Buffer data for a herringbone layout with 12 nodes**

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Nodal Label</th>
<th>Buffer Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>B2</td>
<td>J</td>
<td>1</td>
</tr>
<tr>
<td>B3</td>
<td>K</td>
<td>1</td>
</tr>
<tr>
<td>B4</td>
<td>L</td>
<td>1</td>
</tr>
</tbody>
</table>

The application system was run on a PC machine with a P3 500 MHZ CPU to schedule the problem. The statistics of the generated solutions are presented in Table 5 and the optimal solution is illustrated in figure 6.

**Table 5: Solution statistics for scheduling a pipeless plant in herringbone layout**

<table>
<thead>
<tr>
<th>Solution Statistics</th>
<th>First Solution</th>
<th>Last Solution</th>
<th>System Termination (Confirm Optimal Solution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Activities</td>
<td>130</td>
<td>130</td>
<td>130</td>
</tr>
<tr>
<td>Number of Choice Points</td>
<td>53</td>
<td>137383</td>
<td>241586</td>
</tr>
<tr>
<td>Computer running time (s)</td>
<td>0.1</td>
<td>722.599</td>
<td>945.609</td>
</tr>
<tr>
<td>Makespan (unit of 0.1 hour)</td>
<td>177</td>
<td>123</td>
<td>123</td>
</tr>
</tbody>
</table>

The statistics show that the first solution was found very quickly. However, much more time is needed to find the optimal solution. This example involves a total of 130
activities. This number is made up of forty production and batch activities plus ninety intermediate moving activities that were created dynamically during the search process. The statistics also indicate that 17.7 hours and 12.3 hours are needed to achieve the production demands in the first and optimal solutions respectively. The makespan ratio of the optimal solution to the first solution is about 70%.

Figure 6 shows that both time and resources are allocated properly. The movement of the third vessel is traced to show clearly how the vessel moves from one processing stage to another. The utilization of every station as well as that of the vessels is also illustrated. For a vessel, the gap between two processing activities represents the time that the vessel is either moving or waiting.

In the examined layout the processing stations are evenly distributed over the layout structure. The salient feature is the locations of the blenders and reactors. Blender1 and Reactor1 are located opposite to each other on one “side” of the layout structure, and Blender2 and Reactor2 are both located opposite to each other on the other “side”. By tracing the movements of the vessels, it can be seen that vessels cycled either on the “left” side or on the “right” side. This type of behavior resulted because all stations needed to complete a product recipe exist on either the “left” side or the “right” side and it saves moving time.
Figure 6: The optimal solution for scheduling a pipeless plant in herringbone layout

The first solution is also illustrated in Figure 7 in order to compare it with the optimal solution (Figure 6). The first solution is feasible but has a longer makespan. On closer inspection, the reason is obvious as Vessel 2, Blender 2 and Reactor 2 are not used in
this solution. The optimal solution is better than all other solutions, but the trade-off is that it took longer to find.
Figure 7: The first solution for scheduling a pipeless plant in herringbone layout

Note:
Production Activities include (i.e. the number of Y refers to):
1: Charge A;
2: Charge B;
3: Blend A + B;
4: Reaction;
5: Blend Additives + Int.;
6: Discharge Product;
7: Cleaning
4.3 Other Layouts

The proposed approach can solve scheduling problems of pipeless batch plants in different layouts. The circular and linear layouts shown in Figure 2 are adopted here for the above scheduling example. Figure 8 and Figure 9 show where the stations are located and what tracks and buffers are included for the adopted circular and linear layouts. The application system can be easily applied to these layouts. The user only needs to provide the layout-related data in the form of connectivity tables and the information on tracks and buffers. The resulting solution statistics for the two layouts are presented in Table 6 and 7.

![Figure 8: Scheduling of a pipeless plant in circular layout](image-url)
By scheduling the same problem using different layouts, differences in performance can be identified by comparing Tables 5, 6 and 7. It is clear that the layouts affect the scheduling process and results. Optimal solutions can be found for the herringbone and linear layouts, but optimal solution cannot be found for the circular layout within the set run-time limit of 3 hours. In this case the last solution reached is the best solution so far. Since a batch production includes a number of activities and a station has to be selected for each activity, there can be many possible routes between the first and the last stations for any batch production. Compared with the herringbone and linear layouts, the circular layout has more possible routes and this is reflected in the time required to search for the optimal solution. It is also found from the results that the makespan for the circular layout is the largest in both the first and the last

Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081
4.4 Further Discussion

The above example is presented to show how the proposed approach can solve the integrated problem of scheduling and routing in pipeless batch plants. A number of examples with higher production demands have also been tested and feasible solutions were obtained reasonably quickly. However, the approach may take a long time to find an optimal solution. It is impossible to present every example here due to limited space. In general, a constraint-based approach can be applied to solve a complex problem with many constraints with the aim of finding a feasible solution quickly. However, it is not good to find an optimal solution although it can do so.

A direction comparison of a MILP approach with a constraint-based approach on the scheduling of pipeless plants is not appropriate as no MILP approach is used for route planning so far. There is no published result on integrating scheduling and route planning using a MILP approach.

5. Conclusions

The novel contribution of this work is the development of reports a general constraint-based approach to solve the integrated problem of scheduling and routing in pipeless batch plants. A constraint model is proposed and an application system has been developed. The integrated problem is solved due to the system’s capability to add and delete constraints dynamically during run time when a route is proposed or discarded.
respectively. A typical scheduling problem is illustrated in detail to show how a pipeless plant can be scheduled by considering different layouts and the results demonstrate the feasibility of the proposed methodology. An interesting trend is identified when applying the system to further problems. The constraint-based approach is able to solve a complex problem satisfactorily and quickly. However, for a complex problem it is not able to find the optimal solution within a reasonable time.

References


*Published in Computers and Chemical Engineering, Vol 29(5), 2005, pp1069-1081*


