Integration of Scheduling & Control for Sequential Batch Processes: An Iterative Approach

Shamik Misra*. Ravindra D. Gudi*!

*Department of Chemical Engineering, Indian Institute of Technology, Bombay

! Email: ravigudi@che.iitb.ac.in

Abstract: The need for integration of scheduling and control layers of an enterprise decision making framework is getting more relevant in recent times due to agile market conditions and increasing competition. To exercise automation in its entirety, information flow between various decisions making levels are necessary and integration of these levels not only ensures information exchange but also helps to reach an improved optimal solution for the whole system. In this paper, an iterative framework is outlined which is effective in facilitating integration and information exchange between the control and scheduling layer with reasonable computational efficiency. The framework is applied on a multi-unit, multi-task batch process case study taken from Chu & You (2014). The result demonstrates that the proposed iterative framework is computationally efficient and facilitates integration between the layers in an improved manner, which in turn shows better performance than the sequential approach.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Integration of scheduling & control, iterative approach, batch processes.

1. INTRODUCTION

Traditionally scheduling and control problems have been solved separately. Decision support tools for these tasks have been operated in an isolated way using local optimization models and the need of sharing information between these layers has been ignored (Shobrys & White 2000). The need for information sharing between these layers, which will lead eventually to the overall integration, can be understood by evaluating the following scenario. The solution of the scheduling problem consists of optimal production sequence, production time, resources allocation but the formulation does not consider the dynamic behaviour of the plant. Whereas, strict control policies to meet required product specifications may disrupt the whole schedule. So, it is evident that to execute automation in its entirety the integration between scheduling and control is important. Potential application in other lateral areas (other than the process engineering domain) include integration of flight scheduling with trajectory management to enhance the overall timeliness in aircraft operations.

Scheduling and plant wide control both seek to achieve the optimal operating conditions and it is reasonable to expect that an integration or merger of them will help in achieving more economically viable process. In the current economic environment, chemical industries are prone to many other external factors such as, variation in demand of the products, rapid change in the type of products. There also strict rules to abide by considering environmental and safety regulations. In this context making economic production while maintaining safety and environmental regulations needs information exchange between the control and scheduling layer (Baldea & Harjunkoski 2014). With integrated modelling, information can be shared between these two layers without delay, thus, more economical operation is guaranteed (Zhuge & Ierapetritou 2012). There are several variables which actually connect control and scheduling layer, such as, the state variables at the starting and ending of the transient periods, duration of the periods, amount of material processed in units etc. The integration of scheduling and control results in better modelling since transitions are accounted for; these are normally ignored in traditional approaches. However, this integration does pose several modeling, numerical and organizational challenges due to which scheduling and control has been developed independently over the years (Harjunkoski et al. 2009; Zhuge & Ierapetritou 2012; Shobrys & White 2000).

Though the first mention of integration of scheduling and control dates back to 1970-1980s (Mesarović et al. 1970; Jaikumar 1974), technical issues, such as, limited capability of optimization algorithms, limited computational power hindered the advancement in this topic (Shobrys & White 2000). However, the wave of new innovations in information technology sector in recent years has helped in overcoming these barriers and that is why this problem is now become more and more significant to researchers of academic as well as industrial communities.

There are several attempts that have been made in recent years to integrate the control layer with the production scheduling layer. The most common approach is to incorporate the dynamic model of the process into the scheduling structure, which in turn generates a mixed integer dynamic optimization (MIDO) problem (Flores-Tlacuahuac & Grossmann 2006; Zhuge & Ierapetritou 2012; Chu & You 2014). These MIDO models are then be converted to MINLP by various discretization methods such as, orthogonal collocation, Range-Kutta method etc. However, solving an MINLP for a large problem still poses serious challenges.
Alternatively, Nyström et al. (2005) proposed a decoupled modelling approach where, the scheduling problem acts as master problem and the DO part is considered as primal problem. The MILP (master problem) and DO(primal problem) are solved iteratively, following the same principle of solving MINLPs using outer approximation or benders decomposition method.

In this paper, a relatively simpler iterative approach is considered to attain an integration between the control & scheduling layer. The merits of the proposed approach are proper information exchange between the layers and also increased computational efficiency, which also results in improved overall profitability. The iterative approach is applied on a multi-unit, multi-task batch process to show the efficacy of integration and information exchange between the control & scheduling layer in achieving better and near optimal results.

The remainder of the paper is organized as follows. A brief overview about the case study is presented at section 2. Section 3 consists of the details regarding dynamic optimization and scheduling framework as well as the iterative framework, which is the main novelty of this paper. Then the aforementioned framework is tested against the sequential approach for the case study described in section 2.

2. BRIEF OVERVIEW OF THE BATCH PROCESS

The proposed iterative framework is designed for a multi-unit multi-stage batch process. The process is taken from the published work of Chu & You (2014). There are three operational stages in the process, two reaction stages and one filtration step. The above mentioned batch process is illustrated in Figure 1. There are 8 jobs that needs to be carried out using the process and each job follows the same production recipe. 4 reactor units and one filtration units are available to carry out the entire process. Reactors 1 & 2 can carry out the stage 1 reaction of each job whereas, reactor 3 or 4 can be chosen to perform stage 3. The filtration unit has a fixed processing cost and a fixed processing time. However, the processing costs and time of the reaction stages are calculated using the following dynamic model.

![Figure 1: Batch process recipe](image)

For a chemical reaction such as, \( A \rightarrow B \), the reaction kinetic model,

\[
\frac{dC_A}{dt} = -k_A e^{-\frac{E_A}{RT}(t)} C_A(t) \\
\frac{dI_F(t)}{dt} = T_F(t) - T_{\text{min}}
\]

The variable \( C_A(t) \) in Eq. 1 refers to the reactant concentration at any time period. The initial concentration for reactant A for stage 1 reaction is 1 kmol/m³, whereas, for stage 3 is 0.9 kmol/m³. The reactions shown above are assumed to follow first order reaction law and the rate coefficient is a function of the reactor temperature \( T_R(t) \). The processing cost \( (PC) \) is directly proportional to final integral temperature \( (PT) \) at processing time and can be calculated using the following equation,

\[
PC = C^R I_F(PT)
\]

The reaction times are bounded by the processing time. A desired conversion percentage is given and reactant concentration at the end of the reaction needs to reflect the prescribed conversion. The temperature of the reactor is bounded by the maximum and minimum reaction temperature, and the final reactor temperature should also be equal or less than the prescribed final reactor temperature. The details of the parameters are not given in this paper due to brevity and can be found in Chu & You (2014).

3. FORMULATION

3.1 Dynamic Optimization

Dynamic optimization as a part of the integrating framework essentially provides the reference trajectory for the controller. The continuous time dynamic model of the reaction tasks (Equation 1 – 2) will be discretized and converted to nonlinear algebraic equation by using orthogonal collocation methods, thus reducing the dynamic optimization problem into NLP. The entire reaction time needed to complete a reaction task (task indicates a certain stage of a certain job) is discretized into several smaller equidistant finite elements \( (r) \) and each of these finite elements contains \( q \) number of collocation points. The time at each of these collocation points can be calculated as,

\[
T_{i,j,k,r,q} = T_{i,j,k,r-1} + C_q L_{e,i,j,k} \\
\forall i, j, k(: k \neq 2 and JKdyn_{j,k} = 1), r, q
\]

Where, the parameter \( L_{e,i,j,k} \) denotes the length of the finite element, and can be calculated using Equation 5.

\[
T_{i,j,k,r} = T_{i,j,k,r-1} = L_{e,i,j,k} \\
\forall i, j, k(: k \neq 2 and JKdyn_{j,k} = 1), r
\]

The ODE (Equation 1) to calculate the reactant concentration trajectory are then discretized into algebraic equations using the abovementioned collocation points, and can be expressed as shown in Equation 6-7.
\[
C_{A_{i,j,k},q} = C_{A_{i,j,k},r} + L e_{i,j,k} \sum_{q=1}^{Q} A_{i,j,k} * \left(-k_A e^{-E_A / T_{k,j,k,q} C_{A_{i,j,k},q}}\right)
\]
(6)
\[\forall i, j, k (k \neq 2 \text{and } J K D y n_{j,k} = 1), r, q \]

\[
C_{A_{i,j,k},q} = C_{A_{i,j,k-1}} + L e_{i,j,k} \sum_{q=1}^{Q} b_q \left(-k_A e^{-E_A / T_{k,j,k-1,q} C_{A_{i,j,k-1},q}}\right)
\]
(7)
\[\forall i, j, k (k \neq 2 \text{and } J K D y n_{j,k} = 1), r, q \]

The collocation matrices are adopted from Hedengren et al. (2014).

Equation 2 can similarly be discretized into nonlinear algebraic equations using the abovementioned orthogonal collocation methods. The time needed to perform each reaction stage in a specific reactor can be calculated as,

\[
PT_{i,j,k} = L e_{i,j,k} * (N f e - 1)
\]
(8)
\[\forall i, j, k (k \neq 2 \text{and } J K D y n_{j,k} = 1) \]

Where, \(N f e\) indicates the number of finite elements.

The processing cost calculation expressed in Equation 3 can also be rewritten as,

\[
PC_{i,j,k} = C^R * I_{i,j,k,n} \]
(9)
\[\forall i, j, k (k \neq 2 \text{and } J K D y n_{j,k} = 1) \]

The dynamic model also includes the initial & final specifications for the state variables such as, reactant concentration, integral temperature. The upper and lower limit specifications & the final temperature bounds for the reactor temperature are also needs to be included in the formulation.

The objective function for the dynamic optimization is to minimize the total reaction time of all the reactions, and can be expressed as,

\[
\min TT^R = \sum_{i=1}^{n} \sum_{j=1}^{J} \sum_{k \in K_{i,j,n}} PT_{i,j,k}
\]
(10)

3.2 Scheduling Model

A mixed integer linear programming framework is adopted to address the constraints and limitations of the scheduling problem for the abovementioned batch process. The processing time and costs, calculated using the dynamic optimization, are used in this layer as parameters, and based on that a proper schedule for all the tasks are calculated respecting the process recipe and other constraints. As, all the processing time and costs for all the tasks (reaction task and filtration) are considered as parameters in this layer, all the tasks can be considered as fixed recipe task and the ending time for each of these task can be calculated as,

\[
TE_{i,j,k} = TS_{i,j,k} + \sum_{i \in (J K D y n_{j,k} = 1)} e_{i,j,k} * PT_{i,j,k}
\]
(11)
\[\forall j, k \]

Where, \(e_{i,j,k}\) is a binary variable which indicates the assignment of task \((j,k)\) in unit \(i \in I\). All the tasks has to happen once in the entire horizon and can be processed in only one unit, so the assignment variable is constrained as,

\[
\sum_{i \in (J K D y n_{j,k} = 1)} e_{i,j,k} = 1 \quad \forall j, k
\]
(12)

The batch process has a fixed recipe and the stages of each job needs to be performed sequentially. So, the starting time of say, stage \(k\) should be greater or equal to the ending time of the previous stage.

\[
TS_{j,k} \geq TE_{j,k-1} \quad \forall j, k (k \geq 2)
\]
(13)

A binary precedence variable \(\beta_{i,j,k,j',k'}\) is introduced which essentially dictates the sequence of the tasks that can be performed in a unit. A task changeover times between the tasks to be performed in the same unit are also accounted in the formulation.

\[
TS_{j,k} + PT_{i,j,k} + T_{i}^{ch} - b_u (1 - \beta_{i,j,k,j',k'}) \leq TS_{j,k'}
\]
(14)
\[\forall j, k, j', k' (j,k) \neq (j',k'), i (: IK_{i,k} = 1 \text{and } IK_{i,k'} = 1) \]

The relation between the task assignment variables and the precedence variables are captured in the following equation.

\[
e_{i,j,k} + e_{i,j',k'} - \beta_{i,j,k,j',k'} - \beta_{i,j',k',k} \leq 1
\]
(15)
\[\forall j, k, j', k' (j,k) \neq (j',k'), i (: IK_{i,k} = 1 \text{and } IK_{i,k'} = 1) \]

The abovementioned equation ensures that if task \((j,k)\) and \((j',k')\) are assigned to the same unit then one of the precedence variable, either \(\beta_{i,j,k,j',k'}\) or \(\beta_{i,j',k',k}\) will assume the value 1.

The objective of the scheduling framework is to maximize profit, by increasing sales and decreasing processing costs.

\[
\max \text{Profit} = \sum_j PR_j + \sum_j e_{i,j,k} * PC_{i,j,k}
\]
(16)
Where, $PR_j$ indicates the revenue earned from each job. Processing cost can be further segregated into two parts, processing cost for fixed tasks (filtration) and processing cost for dynamic tasks (reaction). However, in standalone scheduling model, the processing cost for all the tasks will be handled as parameter and will not have any direct effect on the maximization objective. The revenue from sales is also a function of tardiness of the jobs, and depreciates with the delay in completion times. To account the depreciation in the revenue after the due time following set of equations are incorporated.

$$DT_j = DT_j^l + DT_j^u \quad \forall j$$  \hspace{1cm} (17)
$$\gamma_j d_j \leq DT_j^l \leq d_j \quad \forall j$$  \hspace{1cm} (18)
$$0 \leq DT_j^u \leq \gamma_j (C_j^{\text{max}} - d_j) \quad \forall j$$  \hspace{1cm} (19)
$$DT_j = TE_{j(k=\vert K \vert)} \quad \forall j$$  \hspace{1cm} (20)
$$DT_j \leq C_j^{\text{max}}$$  \hspace{1cm} (21)

So, the corrected revenue for each job accounting the effect of the depreciation can be calculated as,

$$PR_j = pr_j \left(1 - \frac{1}{C_j^{\text{max}} - d_j} DT_j^u\right) \quad \forall j$$  \hspace{1cm} (22)

Where, $pr_j$ indicates the actual price of the job. The abovementioned models are adopted from Chu & You (2014) and details of the parameters used in the model can be found in that paper, so those are omitted here to maintain brevity.

### 3.3 Iterative Integration Framework

In the sequential approach, the above mentioned dynamic optimization model (NLP) will be solved first to get the processing cost and processing times for the reaction tasks. These processing costs & times are then used as parameter in the MILP model to get an optimum schedule for the tasks for the given processing time and costs. However, the objective function of the dynamic model is minimization of the processing time which essentially increase the processing costs. Furthermore, if the processing costs are minimized then, reaction times increases which in turn makes the jobs tardy and decreases the revenue that can be earned from sales. The sequential approach does not account for this trade-offs in the framework and leads to suboptimal solutions.

To obtain globally optimal solutions, information exchange between these layers are needed and that can be achieved by integrating & optimizing these two layers simultaneously. However, simultaneous optimization attempts leads to a MIMO framework which after discretization converts to a MINLP framework, which is also very computationally expensive and also global optimum solutions are not guaranteed. Chu & You (2014) have attempted simultaneous optimization framework in their paper, for the same case study described above; however they reported 17.6% optimality gap in the solution of MINLP and also a high computational intensity of 50000s (13.hhr.).

To make the integration more computationally efficient an iterative framework is presented in this paper. The framework is illustrated in Figure 2.

![Figure 2: Iterative framework for integration](image)

The central idea is to design an iterative framework in which the dynamic optimization layer dictates the minimum processing times for the dynamic stages (reactions) for a given processing costs. These processing times are then treated as parameters in the scheduling layer and an optimum schedule of the tasks is produced which ensures maximum profit for those processing times and costs. The processing cost for each of the reactors will then be perturbed and the aforementioned steps are repeated until there is no improvement on the overall profit. The following constraints need to be added with the dynamic model to facilitate the above idea.

$$PC_{i,j,k} = \kappa_{i,j,k}$$  \hspace{1cm} (23)

Where, $\kappa_{i,j,k}$ is a tentative processing cost for the reaction task $(j,k) \in JKD y_{i,j,k}$ in unit $i \in I^D$. Initially a relatively small value of $\kappa_{i,j,k}$ is propagated through the iterative framework, which will result in relatively larger reaction times which in turn makes the jobs tardy and reduces the profit. Then, the values of $\kappa_{i,j,k}$ will be increased by small amount until the profit starts decreasing. It must be pointed out that this search procedure appears to be ad-hoc and direct search like in the quest for the correct set of $\kappa_{i,j,k}$. However a more systematic iterative procedure to improve and accelerate the search procedure is currently under investigation and will be a subject matter of a future publication (Misra & Gudi (2017)).
4. RESULTS & DISCUSSIONS

The abovementioned iterative framework is applied to the case study described in Section 2. The model is solved using GAMS© Optimization suite version 23.3.0 on an Intel® Core™ i7 (3.6 GHz) machine with 16GB RAM. The NLP & MIP models are iteratively solved to the termination criteria of 0.1% improvement between the subsequent iterations, in 1957s (32.62 min). The NLP model has 16993 constraints, 14113 variables and is solved using CONOPT solver. The MIP model is solved using CPLEX 12.4 and consists 4060 constraints, 2732 variables, and 2648 binary variables. The optimal schedule found by iterative method is given in Figure 3.

Figure 3: Gantt chart depicting the optimal schedule found by the proposed iterative method

Chu & You (2014) reported that the profit for the abovementioned case study using sequential approach is 326.3 monetary units. The iterative method introduced in this paper, calculates the maximum profit as, 391.522 monetary units, which amounts to almost 20% improvement when compared with the result obtained by sequential approach. Integration between the control & scheduling layer facilitates in exploiting the benefits that the trade-off offers. The sequential method focuses on minimizing the reaction times as it will increase the sales. But, smaller reaction time effectively increase the processing costs. As, the processing times and costs are sent to the scheduling layer as parameters, sequential methods do not have any tool to exploit this trade-off. The schedule produced using sequential method, presented in Chu & You (2014) also shows more idle times for the reactors as, other stages (filtration) are not completed for the following tasks. Iterative approach exploits this idle times and extends the reaction times which in turn reduces the reaction costs. The exploitation of trade-offs in the proposed iterative and its differences with sequential approach can be discerned more vividly in Figure 4.

Figure 4: Comparison of the objective function components

Table 1: Comparison of computational aspects of integrated and iterative approaches

<table>
<thead>
<tr>
<th>Problem</th>
<th>Nature</th>
<th>Optimality Gap (%)</th>
<th>Time (seconds)</th>
<th>Profit (m.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated (Chu &amp; You (2014))</td>
<td>MINLP</td>
<td>17.6</td>
<td>50000 (13.9 h)</td>
<td>429.4</td>
</tr>
<tr>
<td>Iterative (Proposed)</td>
<td>NLP</td>
<td>-</td>
<td>1957 (0.544 h)</td>
<td>391.522</td>
</tr>
</tbody>
</table>

The following observations may be made. The integrated approach of C & Y is an MINLP which when solved leads to perhaps a local solution with an optimality gap of 17.6 %. The profit reported is higher as discussed earlier. However, it must be noted that the iterative approach proposed here is able to achieve a relatively closer solution (391 monetary units versus 429.4 reported in Chu & You (2014)) at a much improved computational efficiency (32.62 minutes versus 13.9 hours reported in Chu & You (2014)). Also the optimality gap in the proposed iterative approach on the MIP side is 0.5 %. It is expected that with an improved approach to the search of the proposed costs $K_{i,j,k}$, the quality of the objective function in the iterative approach could be made to approach the integrated solution. These aspects are currently under investigation.
5. CONCLUSIONS

In this paper, an alternative and iterative approach to integrate the scheduling and dynamic optimization layers in enterprise wide decision making is introduced. The proposed iterative framework is applied to multi-unit, multi-task batch process case study taken from Chu & You (2014). The results shows that iterative framework effectively exploits the inherent trade-offs of the process and garner significantly more profit than the traditional sequential approach. The proposed iterative approach also approaches the solution obtained through the integrated approach relatively closely. The proposed iterative framework is computationally efficient and produces optimum schedule in significantly less time when compared to the simultaneous optimization approach. Other computational aspects to improve the efficacy of the proposed iterative approach are currently under investigation.

REFERENCES


