A novel hybrid framework integrating artificial intelligence and mathematical programming approaches for chemical batch scheduling

Dan Li, a Taicheng Zheng, a Jie Li a,\*

aCentre for Process Integration, the Department of Chemical Engineering, The University of Manchester, Manchester, M13 9PL, United Kingdom

jie.li-2@manchester.ac.uk

Abstract

In this work, a novel hybrid algorithm that combines a gene expression programming and a sequence-based mixed-integer linear programming model is introduced for the short-term scheduling of multipurpose batch plants. It operates in three stages: the gene expression programming is implemented to generate its best possible solutions; an extracting algorithm is then executed to extract batching, sequencing and allocation information from solutions generated in the first stage; finally, a sequence-based model, with most binary variables being pre-fixed, is solved to optimize task timings and batch sizes. Computational results demonstrate that the proposed hybrid algorithm is capable of handling variable processing times and limited storage capacities. It effectively improves the solution quality by reducing the makespan, with a maximum reduction of up to 21.2%, in comparison to existing hybrid algorithms.

**Keywords**: Multipurpose batch facilities, scheduling, hybrid framework, gene expression programming, mathematical programming

* 1. Introduction

The multipurpose batch process is a collective definition of plants where products have specific processing steps sharing limited resources, such as machines, material, and utilities. To realize reasonable utilization of resources and achieve economic benefits, effective decision tools on scheduling are desirable but challenging to develop due to its NP-hardness. Mathematical formulations have been considerably developed to address the scheduling problems of multipurpose batch plants. These approaches demonstrated that mathematical programming approaches dominate on high solution qualities even the guarantee of global optima. However, excessive computational resources are essential to solve these formulations especially for complex systems with large time horizons or enormous customer demands. Artificial intelligence algorithms, including genetic algorithms, abbreviated as GA (He and Hui, 2010), and evolution algorithms (Han and Gu, 2021), have been developed to tackle the large-scale scheduling challenges, demonstrating commendable computational efficiency. However, these algorithms have been observed to yield solutions of inferior quality compared to those generated by the exact approaches. In our previous study (Li et al., 2022), we endeavored to address this limitation by introducing a hybrid algorithm that combines the strengths of a GA approach in rapid search capability with a mixed-integer linear programming (MILP) model for the superior solution quality. This approach exhibited good performance on reduced computational burdens and the generations of high-quality solutions. However, the genetic algorithm lacks transparency in problem knowledge. Specifically, we are hard to discern why certain scheduling decisions result in a high quality of their corresponding scheduling solution, and we lack insight into which parameters or variables exert the most significant influence on the solution quality. The problem knowledge, partially manifested in dispatching rules (DRs), is crucial to understand the essence of the scheduling problems and make determinations to rapidly respond to the occurrence of uncertainties.

Gene expression programming (GEP) is capable of acquiring problem-specific knowledge through the extraction of DRs. These rules play a crucial role in calculating priorities of tasks or units and serve as the basis for decision-making. They are constructed using attributes related to problem features and objective functions, offering advantages in terms of easy implementation and remarkably rapid computation. In this work, a novel hybrid algorithm (denoted as HA), integrating an enhanced GEP algorithm and a MILP formulation, is proposed to address the multipurpose batch scheduling problems. The objective is to minimize makespan (i.e. time horizon required to complete the production) under a given production requirement. In GEP, DRs are generated to determine the machine assignments and operation sequences individually. The efficacy of these DRs is enhanced through genetic operators. Subsequently, the acquired solution from GEP is further refined using a simplified MILP formulation to orientate the most-reasonable operation timings. Computational studies show that HA is capable to locate the optima in a short time frame for most considered examples and exhibits a significant advantage in computational efficiency relative to the reference mathematical programming approach.

* 1. Problem statement

In the scheduling problem of a multipurpose batch facility, several tasks $i$ from the set $I$ can be executed in their available processing units $j\in J$. Units $j$ able to perform a task $i$ constitute a set $J\_{i}$. Processing times of tasks on the assigned units are variable with their batch sizes. And the variable and fixed processing time of a task $i$ on unit $j$ are denoted by $a\_{ij}$ and $β\_{ij}$, respectively. Batch size of a task $i$ performed on one unit $j$ is bounded by its maximum $B\_{ij}^{max}$ and minimum $B\_{ij}^{min}$ unit capacity. There are $S$ states involved in the process, including raw material $S^{R}$, intermediate material $S^{IN}$ and products $S^{P}$. One state $s$ is consumed or produced by a task $i$ with the proportion $ρ\_{is}$. The production process route to produce a product $s\in S^{P}$ is expressed using $p\in P$, which includes all involved tasks $i\in I\_{p}$ and states $s\in S\_{p}$. Four types of storage policies, including zero-wait (ZW), unlimited intermediate storage (UIS), finite intermediate storage (FIS), and no intermediate storage (NIS) are discussed. The objective considered in this work is to minimize the makespan.

* 1. Methodology
		1. Hybrid framework

In the proposed hybrid algorithm, there are three stages as illustrated in Figure 1. The improved GEP in the first stage generates good-quality solutions where processing times of tasks are fixed as the maximum and independent on the batch size. Then in the second stage, scheduling information concerning batch numbers of tasks, unit allocations and performing sequence for batches of tasks are extracted and expressed using algorithmic parameters. Finally, the great majority of binary variables within a sequence-based MILP model are prefixed using the above extracted parameters, and an improved solution would be yield by solving the MILP formulation to refine the batching and timing determinations.

A pre-processing step is required in the first-stage algorithm of HA to fix the processing times $τ\_{ij}$ of tasks $i$ on its available unit $j$ as the maximum value that can be calculated using the maximum batch size (i.e. $τ\_{ij}=a\_{ij}+β\_{ij}⋅B\_{ij}^{max}$), as shown in Figure 1. This step is necessary because heuristics on batch size in GEP solutions indicate that solutions featuring tasks with maximum batch sizes and minimum batch numbers are likely to exhibit a higher level of quality in minimizing makespan. The third stage in HA employs a sequence-based MILP formulation from Li et al. (2022).



Figure 1. Framework of the proposed hybrid algorithm

* + 1. Gene expression programming
			1. Transformation of GEP individuals

In a GEP algorithm, each individual in a population can be transformed into a unique set of algebraic expressions, specifically DRs. These rules play a crucial role in determining the priority values of candidates, including products to be produced, units for a required task, and tasks producing a necessary state. The production sequence, unit allocation, and task selection are thus established. Subsequently, all batches are assumed to be processed with maximum batch sizes and processing times. To minimize makespan, batches of tasks start at the earliest possible time, allowing for the insertion of a batch before a previously performed one if all storage constraints and material balance are satisfied (Li et al., 2022). Consequently, a feasible scheduling solution can be generated as all variables for required tasks to meet production demands are determined. The makespan for this solution is then to evaluate the fitness of the individual.

In a GEP chromosome, the terminal sets employed to derive DRs for determining production sequence, unit allocation, and task selection encompass various attributes. For instance, when selecting a final product for the next production step, eleven attributes, denoted as $a\_{1}^{P}\~a\_{11}^{P}$, are considered to calculate the priority values of candidate products, as outlined in Table 1. The product with the minimum priority result is then chosen for execution.

Table 1. Terminal sets for constructing DRs to decide the production sequence

|  |  |
| --- | --- |
| Attribute description | Notation |
| Remaining production amount | $$a\_{1}^{P}$$ |
| Storage inventory ratio of states $s\in S\_{p}$ in $p$ | $$a\_{2}^{P}$$ |
| Remaining reaction amount for tasks $i\in I\_{p}$ in $p$ | $$a\_{3}^{P}$$ |
| Minimum production amount for one batch of $p$ | $$a\_{4}^{P}$$ |
| Maximum production amount for one batch of $p$ | $$a\_{5}^{P}$$ |
| Number of states involved in $p$ | $$a\_{6}^{P}$$ |
| Number of tasks involved in $p$ | $$a\_{7}^{P}$$ |
| Number of recycle stream included in $p$ | $$a\_{8}^{P}$$ |
| Accumulate processing times of tasks in $p$ | $$a\_{9}^{P}$$ |
| States $s\in S\_{p}$ are subject to ZW or NIS | $$a\_{10}^{P}\in \left\{0,1\right\}$$ |
| $p$ is included in production process route for other final products | $$a\_{11}^{P}\in \left\{0,1\right\}$$ |

* + - 1. Chromosome representations

A multigene structure is employed to construct the chromosome, as exemplified by Figure 2, which depicts the segment of a chromosome constructing the DR for the production sequence. Each gene, consisting of a head and tail with a fixed length of symbols, can be encoded as an expression tree using the breadth-first searching method. Different genes are connected using linking functions, represented by a linking gene (see the last gene in Figure 2). The terminal set in the linking gene differs from those in other genes, where terminals denote the ordinal numbers of genes (i.e. G1, G2 denoting the Gene 1 and 2, respectively) in the chromosome. The function sets in all genes comprise of five operators ($+, -, ×, ÷, √$).



Figure 2. The part of chromosome constructing DRs for the production sequence

* + - 1. Evolution of population in GEP

The evolutionary process of the population in GEP involves fitness evaluation, selection, crossover, and mutation. These operators are crucial for identifying individuals with exceptional adaptability, inheriting valuable metrics, and preserving genetic diversity. The tournament selection strategy is employed, where two candidate individuals are compared, and the more fit individual survives. To address the possibility of premature convergence, particularly in the mid to late stages of evolution, adaptive crossover and mutation approaches are implemented. The adaptive crossover ($C\_{g}^{r}$) and mutation ($M\_{g}^{r}$) rates at current generation $g$ can be calculated using equations (1-2), respectively. In these equations, the variables $MS\_{g}^{avg}$, $MS\_{g}^{max}$ and $MS\_{g}^{min}$ represent the average, maximum and minimum values of makespan across all individuals in generation $g$. One-point crossover is performed in each gene of the parent chromosome. And the head and tail segments in each gene individually execute the one-point mutation to generate a child from a parent.

|  |  |
| --- | --- |
| $$C\_{g}^{r}=C\_{0}^{r}⋅\left(1+0.2⋅\frac{g}{g^{max}}⋅\frac{MS\_{g}^{avg}^{2}}{\left(MS\_{g}^{max}-MS\_{g}^{min}\right)^{2}+MS\_{g}^{avg}^{2}}\right)$$ | (1) |
| $$M\_{g}^{r}=M\_{0}^{r}⋅1.2^{\left(\frac{\left(g-1\right)⋅MS\_{g}^{avg}}{\left(MS\_{g}^{max}-MS\_{g}^{min}\right)+MS\_{g}^{avg}}\right)^{0.3}}$$ | (2) |

* 1. Computational studies

To assess the performance of the proposed hybrid algorithm (denoted as HA), we conducted tests on ten examples, including seven benchmark examples (Exs 1-7) with states subject to UIS, and three examples (Exs 8-10) considering the storage policy of FIS. Specifically, the examples 1-6 correspond to the examples 1, 2, 3, 7, 8 and 9 sourced from Li et al. 2022, while examples 8-10 originate from the work of Maravelias and Grossmann (2003), involving seven instances (Ins) with varying demand requirements. We evaluate the computational performance of HA by comparing it with a unit-specific event-based MILP model from Li et al. (2022) (referred to as L&R), as it is demonstrated superior to other continuous-time MILP formulations. We also compare with the model L&M (Lee and Maravelias, 2018), recognized as an effective hybrid algorithm that combines the advantages of the discrete- and continuous-time MILP formulations.

Table 2 presents a comparison between the HA and existing approaches L&M and L&R for addressing facilities involving unlimited storage capacities. Here, the model L&R guarantees the generation of globally optimal solutions with sufficient computational resources. Remarkably, the proposed HA yield identical objective results to the L&R except the Ex 3, underscoring that our proposed HA attains the global optimum across six out of seven examples. In terms of computational performance, the HA reduces the computational time by over an order of magnitude relative to L&R for Ex 2 (i.e. 4.8 s vs. 3600 s). While comparing with L&M, solutions from the HA outperform those from L&M in Exs 1, 3, 4, 5, and 6 with the maximum improvement in makespan reduction of 21.2% (e.g. 16.5 h vs. 20.0 h in Ex 6). This highlights the superiority of HA in producing more favourable outcomes.

Table 2. Computational results for examples 1-7 from model HA, L&M and L&R

|  |  |  |  |
| --- | --- | --- | --- |
| Ex | HA | L&M | L&R |
| $MS$ (h) | CPU (s) | $MS$ (h) | CPU (s) | $MS$(h) | CPU(s) |
| 1 | 4.4 | 0.8 | 5.3 | 0.1 | 4.4 | 0.1 |
| 2 | 46.1 | 4.7 | 46.1 | 0.6 | 46.1a | 3600 |
| 3 | 14.8 | 108 | 14.6 | 0.3 | 13.4 | 0.3 |
| 4 | 193.3 | 4.7 | 197.2 | 0.6 | 193.3 | 0.2 |
| 5 | 16.5 | 2.8 | 17.0 | 0.4 | 16.5 | 0.8 |
| 6 | 16.5 | 3.9 | 20.0 | 0.2 | 16.5 | 7.3 |
| 7 | 14.3 | 1.0 | 14.3 | 0.5 | 14.3 | 0.1 |

Relative gap, a: 1.17%

As the model L&M is proposed for processes characterized by unlimited storage capacities, its implementations are tailored to solve examples 1-7. Computational results for Exs 8-10 from HA and L&R are presented in Table 3. For complicated systems (e.g. Ins 3 with enormous production demands and finite storage constraints), the advantage of the proposed hybrid algorithm primarily manifests in the reduced computational burdens. Specifically, computational times are decreased by 99% (i.e. 2.9 s vs. 3600 s), 96.5% (i.e. 125 s vs. 3600 s), 95.6% (160 s vs. 3600 s) and 99.8% (i.e. 7.4 s vs. 3600 s) in HA relative to L&R for Ins 1, 2, 3, and 5 (see Table 3), respectively.

Table 3. Computational results for examples 8-10 with FIS from the proposed HA and the MILP model L&R

|  |  |  |  |
| --- | --- | --- | --- |
| Ex | Ins | HA | L&R |
|  | $MS$-GEP(h) | $MS$-HA(h) | CPU(s) | $MS$(h) | CPU(s) |
| 1 | 1 | 37 | 37 | 2.9 | 37a | 3600 |
| 2 | 108 | 108 | 125 | 109b | 3600 |
| 3 | 217 | 217 | 160 | NA | 3600 |
| 2 | 4 | 37 | 37 | 1.7 | 37 | 0.3 |
|  | 5 | 100 | 100 | 7.4 | 100c | 3600 |
| 3 | 6 | 44 | 44 | 15.1 | 44 | 2.6 |
|  | 7 | 158 | 158 | 119.4 | 158 | 30.6 |

Relative gap, a: 1.1%, b: 1.4%, c: 9.1%

* 1. Conclusion

In this study, we introduce a hybrid algorithm designed for optimizing the short-term scheduling of multipurpose batch plants. The hybrid algorithm is proposed to harness the strengths of both gene expression programming and mathematical programming approaches on computational efficiency and solution accuracy, respectively. And it is capable to address various operational characteristics, such as various or fixed processing time and constrained storage capacity. The computational results for ten benchmark examples demonstrate that the proposed algorithm outperforms the literature hybrid approach in terms of objective results with a maximum reduction of 21.2% in makespan. Furthermore, the computational time is significantly reduced by the proposed algorithm over one order of magnitude, compared to a representative unit-specific event-based continuous-time formulation.

Acknowledgement

Dan Li and Taicheng Zheng appreciate financial support from China Scholarship Council - the University of Manchester Joint Scholarship (201908130170, 202106440020). Jie Li appreciates financial support from Engineering and Physical Sciences Research Council (EP/T03145X/1, EP/V051008/1).

References

Y. Han, and X. Gu, 2021, Improved multipopulation discrete differential evolution algorithm for the scheduling of multipurpose batch plants, Industrial & Engineering Chemistry Research, 60(15), 5530 - 5547.

Y. He, and C.W. Hui, 2010, A binary coding genetic algorithm for multi-purpose process scheduling: A case study, Chemical Engineering Science, 65(16), 4816 - 4828.

H. Lee, and C.T. Maravelias, 2018, Combining the advantages of discrete-and continuous-time scheduling models: Part 1. Framework and mathematical formulations, Computers & Chemical Engineering, 116, 176 - 190.

D. Li, D. Zhang, N. Zhang, L. Zhang, and J. Li, 2022, A novel hybrid algorithm for scheduling multipurpose batch plants, Computer Aided Chemical Engineering, 51, 961 - 966.

D. Li, N. Rakovitis, T. Zheng, Y. Pan, J. Li, and G. Kopanos, 2022, Novel Multiple Time-grid Continuous-time Mathematical Formulation for Short-term Scheduling of Multipurpose Batch Plants, Industrial & Engineering Chemistry Research, 61(43), 16093 - 16111.

C.T. Maravelias, and I.E. Grossmann, 2003, Minimization of the Makespan with a Discrete-Time State-Task Network Formulation, Industrial & engineering chemistry research, 42(24), 6252 - 6257.