Causality-driven dynamic scheduling of multipurpose batch plants

Taicheng Zhenga, Dan Lia, Jie Lia\*

a Centre for Process Integration, Department of Chemical Engineering, School of Engineering, The University of Manchester, Oxford Road, Manchester, M13 9PL, United Kingdom

\*Corresponding author email: jie.li-2.manchester.ac.uk;

Abstract

Modern multipurpose batch plants often require non-periodic rescheduling in response to random disturbances. The rescheduling strategy, that is, when to reschedule and how to reschedule, is crucial not only for reducing long-term cost, but also for the alleviation of nervousness within production environment. However, such two objectives often conflict in practice. In this study, we propose a causality-driven rescheduling strategy to address this challenge in multipurpose batch plants that widely exist in chemical, pharmaceutical, and food industries. The main idea is to use a directed Probabilistic Graphical Model (PGM) to formalize the temporal and spatial causal relationships between tasks. As the dynamic system evolves, given the evidence of observed disturbances, the PGM serves as a standby knowledge base to help scheduler infer the posterior distribution of impacted level of unobserved tasks. When the combined count of observed highly affected tasks and unobserved but highly likely to be impacted tasks exceeds a threshold, rescheduling is triggered. Also, the strategy alleviates system nervousness by fixing tasks that are less likely to be highly impacted. We compare this strategy with the classic exhaustively-minimize-cost strategy on a standard benchmark problem. Numerical results show that, comparatively, our strategy reduces 91.4% computational time and 91.2% count of task changes, at a sacrifice of 18.7% cumulative cost.

**Keywords**: dynamic scheduling, causality, directed Probabilistic Graphical Model, multipurpose batch plant.

* 1. Introduction

Practically, due to the presence of irregularly arriving disturbances, multipurpose batch plants often operate in a dynamic pattern (Ouelhadj and Petrovic, 2009). This pattern often involves a repeating process of scheduling, execution, observation, feedback, and next-round rescheduling. For example, in chemical, pharmaceutical, and food industries, during schedule execution, schedulers may need to frequently reschedule in response to various disruptions such as machine breakdown, task delay, raw material shortage, and demand variation. As indicated by the literature, in such dynamic environments, the rescheduling strategy is crucial for enhancing long-term performance. The rescheduling strategy includes two principal aspects: *when* to reschedule and *how* to reschedule. In the case of multipurpose batch plants, arguably because of the complexity of operational logic, previous studies mostly follow a periodically, completely rescheduling strategy (Gupta and Maravelias, 2019). However, as we will provide evidence later in the case study, such strategy often causes a high-level nervousness within the production environment. In plain terms, schedules before and after rescheduling are often quite different. Such nervousness often causes the plant personnel being strained to adjust to a constantly changing schedule, which has negative impacts on the production process (Pinedo, 2016).

When experienced human schedulers implement dynamic scheduling, they do not mechanically reschedule in a periodic manner. Instead, they often at first evaluate the impact of newly observed disturbances on the schedule by causal reasoning, and then adjust only the targeted part of the schedule. In multipurpose batch plants, spatially, if an upstream task was affected by a disruption, downstream tasks in the production process will be affected as well; temporally, if an earlier task is delayed, subsequent tasks on the shared machine will also be affected. Such causality-based expertise, which is mainly achieved by click-and-drag human-machine interaction in scheduling software, has been proved to be effective in many cases (Harjunkoski et al., 2014). However, to the best of our awareness, no literature has formalized this idea before. In this study, we propose a causality-driven rescheduling strategy for dynamic scheduling of multipurpose batch plants. The main idea is to use this logical causality to factorize the joint probability distribution of impacted level of to-be-executed tasks by a directed Probabilistic Graphical Model (PGM). When a disturbance is observed, the PGM performs inference algorithms to derive a posterior distribution that presents the overall impacted level of remaining unobserved part of the schedule. This posterior distribution is then used to further design the rescheduling strategy. We test the effectiveness of this idea in one of the standard benchmark problems in process scheduling community (Kallrath, 2002). The results show that the strategy can significantly reduce production nervousness in the measure of task change counts and computational time at a cost of 18.7% cumulative objective value.

* 1. Methodology



Figure 1. Flowchart of causality-driven dynamic scheduling.

In the flowchart, while squared boxes denote mathematical models or algorithms, cylindrical containers denote datasets.

We first describe the dynamic scheduling problem of multipurpose batch plants and then illustrate our methodology by going through the flowchart in Fig. 1. We consider a discrete-time dynamic system of a multipurpose plant that can be characterized by an encapsulating parameter (include plant layout, recipes, and supply/demand profile). In the system, starts from and ends at , is vectorized notation for system state; is action set that the system is scheduled to take within horizon ; is informational state that has been revealed to the scheduler; and denotes transition function. For informational state , we assume that (1) we have known information about the distribution of ; (2) we have a certain length of *vision* over . That is, at -time-periods ahead of the instantiation of disturbance , the information is revealed to the scheduler; (3) after is instantiated, it will not change. The target of the problem is to find a policy such that to minimize the cumulative cost and the cumulative action difference , where is a difference-measuring function that maps the tuple to a non-negative integer.

At the system-level, the system dynamics starts with , , , and . If or binary rescheduling decision (viii) is triggered, the system state (ii) is fed to a Mixed-Integer Linear Programming (MILP) (ix) to generate a to-be-executed schedule (x); otherwise (viii) is not triggered, the system evolves to a new system state as time advances. The essential part of the causality-driven policy is the active interaction between the rest modules. Specifically, after a schedule (x) is output by (ix), rather than directly advance the system by taking actions from (x), we first perform multiple episodes of Monte Carlo simulation (xi) to sample different realizations of unobserved disturbances in the horizon of (x). Then, for the set of tasks that are in (x) but fall beyond vision , which is denoted by , we construct a PGM , such that each node is a random variable that describes the impacted level of unobserved task and the set of directed arcs is the temporal and spatial causal relationship that described in Sec. 1. The PGM factorizes the joint probability distribution of in the semantics that , where denotes the parent nodes of . Then, by feeding the training data (xiv) that was generated from (xi) and processed by feature engineering (xiii), we can estimate a full-fledged PGM as a buffered knowledge base to help the scheduler to infer the posterior distribution from evidence (iv), where is the set of tasks that has been observed being affected by disturbance. In other words, the distribution of impacted level of remain unobserved tasks can be updated by realizations of . Finally, the framework iterates between the three different abstraction levels and advance until .

* 1. Case study

We test the effectiveness of our causality-driven policy on one of the standard benchmark problems in process scheduling community (Kallrath, 2002). The State-Task Network (STN) representation of the problem is shown in Fig. 2. The processing time, minimum and maximum batch size parameter are identical to the supplementary material of (Li et al., 2022). In addition, we assume that there is a $1 setup cost for each task and proportional cost of batch size is 0. The capacity for raw material k0 is infinity, for k11 and k21 is 25 ton, and for the rest is 15 ton. The price for each material is listed in Table 1. The inventory cost per time-period is 0.15 multiple of price and backlog cost for products is 1.5 multiple of price.

Table 1. Price for each material (unit: $)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Material | k0 | k11 | k21 | k22 | k31 | k41 | k42 | k43 | k44 | k51 |
| Price | 1 | 5 | 10 | 10 | 10 | 10 | 10 | 15 | 10 | 15 |
| Material | k52 | k61 | k62 | k63 | p1 | p2 | p3 | p4 | p5 |  |
| Price | 15 | 15 | 20 | 20 | 50 | 50 | 50 | 50 | 50 |  |



Figure 2. State-Task Network (STN) representation of the benchmark problem (Kallrath, 2002).

For system dynamics, we assume that the horizon of each schedule is 72 h (3 days), and we perform a rolling-horizon dynamic scheduling experiment over 336 h (2 weeks) timespan. For raw material k0, the regular supply is 20 ton every hour. For each product, there is a regular demand of 12 tons per 12h that starts from . We assume five types of possible disturbances (Table 2), and the scheduler has a vision of 36h over each type of disturbance.

Table 2. Assumed disturbances in the dynamic scheduling problem

|  |  |  |  |
| --- | --- | --- | --- |
| Disturbance | Raw material supply change | Demand change | Processing time change |
| Distribution | a | b | c |
| Disturbance | Machine breakdown (occurrence) | Machine breakdown (duration) | Yield change |
| Distribution | d | e | f |

a: The raw material supply will be affected by a clipped Gaussian random variable such that the maximum purchasable amount of raw material will be limited to -multiple of the regular supply. The random variable is clipped (that is, when the random variable is sampled beyond the boundary, the value will set to its closest boundary value) between [0.5, 1] for the consideration of nonnegativity. b: The unexpected arrival of demand will be count of orders (clipped Poisson random variable) multiplies amount for each order (uniform random variable). The Poisson random variable is clipped between [0, 2] to avoid infinity. c: The processing time for tasks will be affected by a clipped Gaussian random variable such that the varied processing time will be , where is the nominal processing time and is Gaussian random variable that is clipped between . d: Whether there is a breakdown disruption on a machine is a Bernoulli random variable with . In other words, for each time-period and each machine, there is 0.02 chance that there will be a machine breakdown disruption. e: When a machine breakdown is occurred, the duration of the breakdown is a uniform random variable . f: The yield of each task is -multiple of the original batch size.

* 1. Results

There are various measures to quantify how effective a rescheduling strategy performs in a dynamic environment. To set up a standard benchmark for the numerical results, we consider the following metrics:

* . The optimal objective cost of the static version of scheduling problem under *nominal* conditions. That is, we regard the dynamic scheduling problem as static by assuming that there is no disturbance through the entire dynamic scheduling timespan. Since the occurrence of a disturbance event will usually deteriorate objective, is served as a reference lower bound for the cumulative objective cost in our case study.
* . The optimal value of the dynamic scheduling problem with the assumption that the scheduler has infinite vision over uncertain events. In other words, we assume that the scheduler can foresee all realizations of uncertain events at . is obtained by solving a MILP that involves all realized uncertainties through the timespan. is served for two purposes: (1) when comparing to , reflects how unstable a system is due to the existence of disturbance. Specifically, the larger gap between and is, the more unstable the system is. (2) is the theoretical lower bound for any dynamic scheduling policies because a longer vision will at least not worsen the cumulative objective value.
* . The cumulative objective value of a dynamic scheduling policy with finite vision . is the measure for the effectiveness of dynamic scheduling policies in practice. In this study, we compare of the *causality-driven* policy with two other classic policies: *exhaustively-minimize-cost* and *periodically-minimize-cost*. To explain:
  1. . The cumulative cost of the proposed causality-driven policy. In this policy, we assume that a task is categorized as highly impacted either there is a breakdown on the execution machine, or the task is delayed over 50% of the nominal processing time, or the yield loss exceeds 40% of the original batch size. Rescheduling is triggered when the count of observed highly impacted tasks and unobserved tasks but with over 50% probability being highly impacted exceeds half of the task count in the schedule. When rescheduling is triggered, those not highly impacted tasks are fixed.
  2. . The cumulative cost of exhaustively-minimize-cost policy. That is, the objective cost is minimized through MILP every hour. For the consideration of computational time, the MILP solver is terminated either when the relative MIP gap is below 10% or runtime exceeds 300s. Also, the value of integer variables are hinted from previous solutions.
  3. . The cumulative cost of periodically-minimize-cost policy. That is, the objective cost is minimized every 24h and the count of task changes is minimized every hour. When two types of minimization problems overlap, the former takes priority. The parameter for MILP solver is set identical to that in .

In addition to cumulative cost, another metric that is of our interest is the cumulative value of difference-measuring function . is the measure for plant nervousness of a dynamic scheduling process in practice. That is, the larger value takes, more frequent schedule adjustment is required in production. However, it is not straightforward to define since there are some subtleties in the notion of task change in multipurpose batch plant. First, unlike discrete manufacturing where the number of tasks is often predetermined, multipurpose batch plants may have schedules with variable number of tasks. Second, it can be challenging to determine whether a task is shifted from previous time slots or is a completely new addition. To setup a unified metric for comparison, we define as follows: for the overlapping horizon of and , is the sum of (1) number of tasks appear in but not in , and (2) number of tasks appear in but not in .

The numerical results are presented in Table 3. In such experimental setup, we observe that is 8.94 multiple of . This large gap between and indicates that this system is highly unstable due to the existence of disturbance that described in Table 2. Also, we notice that is even slightly lower than the theoretical lower bound . This is because the computation of is terminated due to 3600s time limit and there is still 1.27% MIP gap for . Therefore, is very close to the theoretical lower bound. However, the exhaustive strategy suffers from a high value of and a very long computational time. As a compromise, if we reduce the rescheduling frequency from every hour to every 24 h, as shown by , we can see an 89.2% reduction of computational time and a 72.6% decrease of . However, is 76.2% higher than , in which is often unacceptable practice. While the trade-off between and seems hard to balance, the causality-driven strategy reduces to 830, which is even significantly lower than that in periodic strategy. Also, is only 18.7% higher than exhaustive policy. Finally, notably, the computational time for the causality-driven policy is only 8.57% of the exhaustive one.

Table 3. Numerical results for the case study

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metrics |  |  |  |  |  |
| a (unit: M$) | 0.235c | 2.102d | 2.482 | 2.091 | 3.684 |
| b | - | - | 830 | 9412 | 2581 |
| Computational time (unit: CPUs) | 3600 | 3600 | 657 | 7666 | 923 |

a: Cost. For and , denotes for the objective value of the static MILP solution. For , , and , is the cumulative cost. The lower is, the lower long-term cost of a rescheduling strategy can achieve. b: Cumulative value of difference-measuring function . The lower is, less frequent schedule adjustment is needed in dynamic scheduling. c: MILP solver terminates because 3600s time limit is reached with relative MIP gap of 20.68%. d: MILP solver terminates because 3600s time limit is reached with relative MIP gap of 1.27%.

* 1. Conclusions

In this study, we propose a causality-driven rescheduling strategy for dynamic scheduling problems in multipurpose batch plants. The main idea originates from the human expertise that schedule modification in practice is often driven by causal relationships between tasks. To formalize such idea, we use directed Probabilistic Graphical Model (PGM) to serve as a standby knowledge base to help scheduler quickly infer the posterior distribution of impacted level of unobserved tasks when disruption events occur. The causality-driven strategy is tested on the benchmark problem (Kallrath, 2002). The numerical results show that, compared to the exhaustively-minimize-cost strategy, our strategy reduces 91.4% computational time and 91.2% count of task changes, at a sacrifice of 18.7% cumulative cost.

* 1. Acknowledgements

Taicheng Zheng and Dan Li appreciate financial support from UOM-CSC (202106440020, 201908130170). Jie Li appreciates financial support from Engineering and Physical Sciences Research Council (EP/T03145X/1, EP/V051008/1).

References

Gupta, D., Maravelias, C.T., 2019. On the design of online production scheduling algorithms. Computers & Chemical Engineering 129, 106517.

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering 62, 161–193.

Kallrath, J., 2002. Planning and scheduling in the process industry. OR Spectrum 24, 219–250.

Li, D., Rakovitis, N., Zheng, T., Pan, Y., Li, J., Kopanos, G., 2022. Novel Multiple Time-grid Continuous-time Mathematical Formulation for Short-term Scheduling of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 61, 16093–16111.

Ouelhadj, D., Petrovic, S., 2009. A survey of dynamic scheduling in manufacturing systems. J Sched 12, 417–431.

Pinedo, M.L., 2016. Scheduling. Springer International Publishing, Cham.