Extending the SI Decomposition to Continuous-Time Two-Stage Scheduling Problems

Daniel Montes[a*], José Luis Pitarch[b], César de Prada,[a,c]

[a]*Department of Systems Engineering and Automatic Control, Universidad de Valladolid, C/ Prado de la Mergelina 3-5, Valladolid (47011), Spain.*

[b]*Instituto Univ. de Atutomática e Informática Industrial (ai2), Universitat Politècnica de Valencia, Camino de Vera s/n, Valencia (46022), Spain.*

[c]*Institute of Sustainable Processes, Universidad de Valladolid, C/ Dr. Mergelina s/n, Valladolid (47011), Spain.*

*danielalberto.montes.lopez@uva.es

Abstract

Scheduling often involves making decisions in presence of uncertainty, which governs the pricing of raw materials, energy, resource availability, demands, etc. A common approach to incorporate uncertainty in the decision-making process is using two-stage stochastic formulations. Unfortunately, the mathematical complexity of the resulting problems grows exponentially with the number of uncertainty scenarios, which is further complicated by the presence of binary variables The authors have recently proposed a method using the so-called Similarity Index for discrete-time two-stage scheduling problems that enable scenario-based decomposition. This paper extends this method for scheduling problems formulated on a continuous-time basis. The fundamental idea is to use the Similarity Index to meet non-anticipation in the binary variables and Progressive Hedging on the continuous ones. The proposal is tested on a literature case study that consists of a multiproduct plant with a single processing unit. The combined SI-PH decomposition managed to solve the problem much faster than its monolithic counterpart.

Keywords: Similarity Index, Progressive Hedging, Optimization under uncertainty.

1. Introduction

The authors recently presented a decomposition method based on the idea of the Similarity Index (SI) to efficiently solve two-stage stochastic scheduling problems (TSSP) formulated using a discrete-time basis (Montes et al., 2022a, 2022b). TSSP handles uncertainty by discretizing the underlying probability distribution of uncertain parameters in a set of scenarios with associated probabilities. Then, the decision variables of the problem are grouped into two sets: first-stage and second-stage ones. In the first stage, the actual value that uncertain parameters will have is still unknown, but some decisions must be made regardless. Hence, the schedule within this first stage cannot be tailored to each scenario, as the uncertainty is not revealed yet. This is known as non-anticipation, i.e., decisions in the first stage must be unique. In the second stage, the values of the uncertain parameters are realized and assumed known, so the decisions can be adjusted to each realization accordingly.

Two-stage formulations are challenging to solve, as the problem size grows exponentially with the number of scenarios considered. This fact is further complicated by the presence of many binary variables, usual in scheduling formulations. One approach to tackle this issue is to solve each scenario as an independent subproblem of reduced size, but the non-anticipation criterion for the first stage must be satisfied. The Progressive Hedging Algorithm (PHA) was originally proposed for convex problems involving only continuous variables and has convergence and optimality guarantees (Rockafellar & Wets, 1991). However, although the assumptions and guarantees vanish in presence of discrete variables, this method has been used successfully as a heuristics for mixed-integer problems (Bashiri et al., 2021; Peng et al., 2019). In the PHA, the non-anticipation constraints are relaxed, which enables scenario decomposition. Non-anticipation is then progressively enforced by penalizing the deviation from the average schedule in the first stage, i.e., the decision-variables average values computed from the subproblems solutions in a previous iteration.

Likewise, the Similarity-Index decomposition also enables scenario decomposition by approximating, or estimating, the degree of non-anticipation in the discrete decisions of the first stage. The scenario subproblems are solved independently and the similarity among their solutions is then measured using the SI. A weighing parameter is used to progressively increase the importance of maximizing the similarity. Eventually, all scenario solutions are equal in the first stage so non-anticipation is met. However, the SI decomposition was originally devised for discrete-time scheduling formulations. In this work, the authors aim to combine the SI decomposition with Progressive Hedging to decompose TSSP based on continuous-time formulations. The SI handles the binary variables of the problem, while Progressive Hedging handles the continuous ones. The proposal is tested successfully on a literature case study.

The paper is organized as follows: Section 2 presents a summary of the SI concept and its extension to continuous-time scheduling formulations. Section 3 describes the proposed SI-PH decomposition algorithm. The case study is presented in Section 4. A comparison between the proposed approach and the standard monolithic approach is summarized in Section 5. Conclusions and open issues are outlined in Section 6.

2. Similarity Index in Continuous-Time Formulations

In discrete-time basis, the Similarity Index is relies on the idea of fuzzifying the binary decision variables along the surrounding time periods. The generated areas from the fuzzification for each scenario are compared and their overlap is used as a measure of similarity. Consequently, even if the decisions are not made in the same period among scenarios, one could quantify their similarity by temporal proximity. As the non-anticipation constraints (NAC) require the scenario schedules to be equal in the first stage, the SI can be used to remove the NACs from the formulation, enabling scenario decomposition. The SI is can be pushed up through adding it to the objective function.

Two main difficulties arise for extending the Similarity-Index Decomposition to continuous-time TSSP: Time synchronization of decisions among scenarios; and weighing the slot duration as part of the SI computation. To get around these limitations, we propose: 1) not to consider the slot duration in the fuzzification process, and 2) to fuzzify in a slots basis instead of time periods. Then, the SI formula becomes:

$$SI := \sum_{l \in \mathcal{L}_1} \frac{\min_{e \in \mathcal{E}} \left\| y_{le} + 0.5 y_{(l+1)e} + 0.5 y_{(l-1)e} \right\|_1}{2|\mathcal{L}| - 1} \tag{1}$$

Where the time slots l belong to an ordered set $\mathcal{L}$. The subset $\mathcal{L}_1 \subset \mathcal{L}$ implies that the Similarity Index is only computed for the decisions made in the first stage of the problem. y_{le} are the decision variables indexed by the slots and the scenarios $e \in \mathcal{E}$. In essence, the Similarity Index is computed as the intersection of the generated areas from fuzzifying the decisions divided by the maximum intersection possible. Formula (1) implies that the similarity is only weighed in the two surrounding slots. As in the time-basis case (Montes et al., 2022b), this formula can be generalized for different fuzzification lengths.

Note that (1) does not cover the slot duration and other continuous variables that might be present in the problem (such as inventory levels, production levels, etc.). We propose to use Progressive Hedging (PH) to handle these variables, which results in a hybrid SI-PH method. Incorporating both the SI and the PH in a general TSSP, each subproblem formulation is given by:

$$
\begin{aligned}
&\min_{x,y} J_e - \lambda\, SI_e + \sum_{l \in \mathcal{L}_1} \left(\omega_{le} x_{le} + \frac{\rho}{2} \left(\frac{x_{le} - \bar{x}_l}{\bar{x}_l} \right)^2 \right) \\
&\text{s.t.:} \\
&\text{Specific process constraints} \\
&SI_e = \frac{\sum_{l \in \mathcal{L}_1} |s_{le}|_1}{2|\mathcal{L}| - 1} \\
&s_{le} \le y_{le} + 0.5 y_{(l+1)e} + 0.5 y_{(l-1)e} \quad \forall l \in \mathcal{L}_1 \\
&s_{le} \le \overline{y}_l + 0.5 \overline{y}_{l+1} + 0.5 \overline{y}_{l-1} \quad \forall l \in \mathcal{L}_1 \\
&x \in \mathbb{R}, y \in \{0,1\}
\end{aligned} \tag{2}
$$

Note that the exact/global SI cannot be computed inside the subproblems to be solved independently, and that the element-wise operator min{·} in (1) is non-linear, so it cannot be used in the usual MILP scheduling formulations. Thus, the SI is estimated locally in each subproblem (2) by replacing the intersection of the fuzzified variables with a set of slack variables that are upper-bounded. The slack variables s_{le} are maximized in the objective function so this bound is tight. The SI_e is then an approximatin of the local SI by scenario, computed as the similarity of the scenario schedule with a reference schedule defined by fixed values $\overline{y}_l$. For details about this estimation refer to (Montes et al., 2022b). λ and ω_{le} are the weighing parameters for the SI and the PH. They are updated in each iteration k from the scenario solutions, as follows, where $\bar{x}_l$ is the expected value of x, using probabilities p_e, in the first stage:

$$
\begin{gathered}
\lambda^{(k+1)} = \lambda^{(k)} - \alpha(SI - 1); \;\; \omega_{le}^{(k+1)} = \omega_{le}^{(k)} + \rho \left(\frac{x_{le} - \bar{x}_l}{\bar{x}_l} \right) \; \forall l \in \mathcal{L}_1, e \in \mathcal{E} \\
\bar{x}_l = \sum_{e \in \mathcal{E}} p_e x_{le} \;\; \forall l \in \mathcal{L}_1
\end{gathered} \tag{3}
$$

The parameters α and ρ are the step-size for λ and ω. They greatly influence the algorithm behavior, both in terms of optimality and convergence, as discussed later.

3. SI-PH Decomposition

Non-anticipation can be enforced progressively using the SI and PH ideas. Hence, the original problem can be decomposed into scenario subproblems (2) by dropping the non-anticipation constraints. Each subproblem is solved independently and in parallel. The subproblems solutions are later used to update the required parameters for the SI and PH with (3). The procedure is repeated until either the convergence criteria are met, or the maximum number of iterations is reached. Algorithm 1 below shows the pseudo-code for the SI-PH decomposition.

Algorithm 1

Require $\alpha, \rho, k_{\max}, \Delta, \mathrm{tol}$

$k \leftarrow 0, \lambda^{(0)} \leftarrow 0, \omega_{le}^{(0)} \leftarrow 0, \bar{y}_l \leftarrow 0, \bar{x}_l \leftarrow 0$

repeat

  for $e \in \mathcal{E}$ **do**

    $y_{le}^*, x_{le}^* \leftarrow \underset{y_{le}, x_{le}, s_{le}}{\arg\min} J_e - \lambda^{(k)} SI_e + \sum_{l \in \mathcal{L}_1} \left(\omega_{le}^{(k)} x_{le} + \frac{\rho}{2} \left\| x_{le} - \bar{x}_l^{(k)} \right\|_2^2 \right)$

    $SI_e^* \leftarrow SI_e(s_{le}^*)$

  end for

  $SI \leftarrow SI(y_{le}^*)$ using (1)

  $\bar{y}_l \leftarrow \underset{y_{le}^*}{\arg\min} SI_e^*$

  $\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \alpha\left(SI^{(k)} - 1\right)$

  $\bar{x}_l^{(k+1)} \leftarrow \sum_{e \in \mathcal{E}} \mathrm{p_e}\, x_{le}^{(k)} \quad \forall l \in \mathcal{L}_1$

  $\omega_{le}^{(k+1)} \leftarrow \omega_{le}^{(k)} + \rho\left(x_{le}^{(k)} - \bar{x}_l^{(k+1)} \right) \quad \forall l \in \mathcal{L}_1, e \in \mathcal{E}$

  $k \leftarrow k + 1$

until $\left(SI = 1 \wedge \left\| x_{le}^{(k)} - \bar{x}_l^{(k+1)} \right\|_2 \leq \mathrm{tol} \right) \vee k = \mathrm{k}_{max}$

return y_{le}^*, x_{le}^*

The convergence criteria need to check that non-anticipation is met. For binary variables, that is easy to verify, as the SI needs to reach the value of 1. For continuous variables, non-anticipation can be confirmed if the 2-norm of the difference between the scenario solutions and the reference value is beyond a small enough tolerance.

4. Case Study

The SI-PH decomposition was tested on a case study originally presented by Dogan & Grossmann (2006). The model uses a continuous-time representation for simultaneously integrating both the scheduling and planning of a continuous multiproduct plant with a single processing unit. The planning horizon is divided into fixed-duration time periods $t \in \mathcal{T}$ (weeks), which are further subdivided into variable-duration time slots $l \in \mathcal{L}$. Products $\mathrm{i} \in \mathcal{I}$ are committed to each slot using binary variables W_{ilt}. Only the modifications to the original formulations are listed in this work for brevity and not the original equations.

The original formulation of Dogan & Grossmann (2006) is here extended to consider the production rates r_i as uncertain. Consequently, an additional index $e \in \mathcal{E}$, corresponding to the set of uncertainty realization scenarios, is added to all variables and equations of their model. As the time horizon is divided into both weeks and slots, additional variables W_{ije}^{aux}, with an additional auxiliary slot index j, are introduced to compute the Similarity Index as in (1). This variable is equal to the product assignment W_{ilte} but indexed by j. The index j enumerates the slots from 1 to $N \cdot |\mathcal{T}_1|$. Where N is the number of slots per time period and $\mathcal{T}_1$ is the subset of time periods belonging to the first stage. For instance, given time periods made up of three slots, the first slot of the second time period corresponds $j = 4$.

$$W_{ije}^{\text{aux}} = W_{ilte} \qquad \forall j = l + |\mathcal{L}| \cdot (t-1), l, t, e \tag{4}$$

The original formulation of the problem did not explicitly consider the slot duration ϕ_{lte}. As the non-anticipation criterion of the two-stage formulation requires the slot duration variables to be equal among the scenarios in the first stage, some additional constraints are added to compute them from the starting time of the slots Ts_{lte}:

$$\phi_{lte} = Ts_{l+1,te} - Ts_{lte} \qquad \forall l \neq l_N, t \in \mathcal{T}_1, e \tag{5}$$

$$\phi_{l_N te} = \mathrm{HT_t} - Ts_{l_N te} \qquad \forall t \in \mathcal{T}_1, e \tag{6}$$

Where $\mathrm{HT_t}$ is the elapsed time (in hours) from the beginning of the time horizon, at the end of each time period.

In addition to the economic function of the problem z^P, the objective function shall include the terms corresponding to the SI and the PH. The SI is computed for the product assignment variables W_{ije}^{aux} while the PH is applied to the slot duration ϕ_{lte}, and the inventory levels INV_{ite}. The objective function to minimize in each subproblem is:

$$\begin{aligned} J_e \coloneqq z_e^P - \lambda SI_e + \sum_{i,t \in \mathcal{T}_1, e} \omega_{ite} \frac{INV_{ite}}{\overline{\mathrm{INV}}_{it}} + \sum_{l,t \in \mathcal{T}_1, e} \mu_{lte} \frac{\phi_{lte}}{\overline{\phi}_{lt}} \\ + \frac{\rho_1}{2} \sum_{i,t \in \mathcal{T}_1, e} \left(\frac{INV_{ite} - \overline{\mathrm{INV}}_{it}}{\overline{\mathrm{INV}}_{it}} \right)^2 + \frac{\rho_2}{2} \sum_{l,t \in \mathcal{T}_1, e} \left(\frac{\phi_{lte} - \overline{\phi}_{lt}}{\overline{\phi}_{lt}} \right)^2 \bar{x}_l \end{aligned} \tag{7}$$

After solving each subproblem, the multipliers λ, ω_{ite} and μ_{lte}, and the expected values $\overline{\mathrm{INV}}_{it}$ and $\overline{\phi}_{lt}$ need to be updated, as follows:

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha(SI - 1)\,; \qquad \overline{\mathrm{INV}}_{it} = \sum_{e \in \mathcal{E}} p_e INV_{ite} \;\; \forall i, t \in \mathcal{T}_1;$$

$$\omega_{ite}^{(k+1)} = \omega_{ite}^{(k)} + \rho_1 \frac{INV_{ite} - \overline{INV}_{it}}{\overline{\mathrm{INV}}_{it}} \;\; \forall i, t \in \mathcal{T}_1, e; \quad \overline{\phi}_{lt} = \sum_{e \in \mathcal{E}} \pi_e \phi_{lte} \;\; \forall l, t \in \mathcal{T}_1;$$

$$\mu_{lte}^{(k+1)} = \mu_{lte}^{(k)} + \rho_2 \frac{\phi_{lte} - \overline{\phi}_{lt}}{\overline{\phi}_{lt}} \quad \forall l, t \in \mathcal{T}_1, e \tag{8}$$

Values α, ρ_1, and ρ_2 are the tuning parameters of the algorithm.

5. Preliminary results

The combined SI-PH decomposition algorithm was tested on a case study instance of five products (A, B, C, D, E), eight weeks, and five slots. A total of eight uncertainty realization scenarios were considered. Problem 2a parameters from (Dogan & Grossmann, 2006) were used as a basis for the instance. The resulting problem had 17601 variables, 2240 of which were binary, and 17827 equations. GAMS 40.2.0 was used to code the model, and Gurobi 10.0 to solve it. All calculations were performed on a 2-CPU Xeon Gold 6130 computer. The optimality gap for Gurobi was set to 0.5% in all cases.

The monolithic instance was assigned 32 threads. The solver failed to provide an optimal solution after 10 hours of computation. The reported objective value of the best feasible solution found till that moment was $z^P = 57643$, reporting a 6.06% optimality gap.

Each of the subproblems in the decomposed instance was assigned 8 threads. Note that although the machine only has 32 cores, some scenarios are solved much faster than others, which frees cores for other threads. Note that, if too few threads are assigned per subproblem, the overall CPU capacity may be underutilized.

The tuning parameters greatly affected the quality of the solution. The values $\rho_1 = 95$, $\rho_2 = 120$, and $\alpha = 190$ provided the highest quality solution. The decomposition approach arrived at an objective value of $z^P = 54682$ in 1552 seconds. Compared to the objective value of the monolithic approach, this solution is around 5% worse. However, just setting $\rho_2 = 115$ reaches a much suboptimal $z^P = 48717$, and with $\rho_2 = 125$ the algorithm did not converge after 2000 iterations.

6. Summary

This paper presented a hybrid decomposition method for TSSPs formulated on a continuous-time basis. The non-anticipation constraints are replaced by a combination of the Similarity Index and Progressive Hedging. This allows decomposing the original problem into smaller subproblems that are easier to solve. Their solutions are combined to iteratively build a high-quality feasible solution to the original problem.

Unfortunately, tuning the algorithm is not smooth due to the PH part, and small parameter variations lead to very different objective values. Some reported solutions are even far from optimality. Future work will focus on rethinking the PH decomposition strategy.

Acknowledgments

These results are funded by the Spanish MCIN/AEI as part of the a-CIDiT (PID2021-123654OB-C31, PID2021-123654OB-C32) and LOCPU (PID2020-116585GB-I00) research projects. The first author has received financial support from the 2020 call for pre-doctoral contracts of the University of Valladolid and Banco Santander.

References

Bashiri, M., Nikzad, E., Eberhard, A., Hearne, J., & Oliveira, F. (2021). A two stage stochastic programming for asset protection routing and a solution algorithm based on the Progressive Hedging algorithm. *Omega (United Kingdom), 104.*

Dogan, M. E., & Grossmann, I. E. (2006). A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants. *Industrial and Engineering Chemistry Research*, *45*(1), 299–315.

Montes, D., Pitarch, J. L., & de Prada, C. (2022a). Decomposition of Two-stage Stochastic Scheduling Problems via Similarity Index. *Computer Aided Chemical Engineering, 51,* 985–990.

Montes, D., Pitarch, J. L., & de Prada, C. (2022b). The Similarity Index to Decompose Two-Stage Stochastic Scheduling Problems. *IFAC-PapersOnLine, 55*(7), 821–826.

Peng, Z., Zhang, Y., Feng, Y., Rong, G., & Su, H. (2019). A Progressive Hedging-Based Solution Approach for Integrated Planning and Scheduling Problems under Demand Uncertainty. *Industrial and Engineering Chemistry Research*, *58*(32), 14880–14896.

Rockafellar, R. T., & Wets, R. J. B. (1991). Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Mathematics of Operations Research*, *16*(1), 119–147.