12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Two-Stage Stochastic Scheduling of a Multiproduct Pipeline System using Similarity Index Decomposition

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

[*] *Department of Systems Engineering and Automatic Control, Universidad de Valladolid, Spain (e-mail: danielalberto.montes.lopez@uva.es, prada@autom.uva.es).*
[**] *Institute of Sustainable Processes, Universidad de Valladolid, Spain*
[***] *Instituto U. de Automática e Informática Industrial (ai2), Universitat Politècnica de Valencia, Spain (e-mail: jlpitarch@isa.upv.es).*

Abstract: Multiproduct pipelines are crucial for delivering substantial quantities of refined oil products from major supply centers to clients within a nearby geographical area. Despite the significant infrastructure investment, the associated transportation costs are markedly lower than those incurred with traditional delivery trucks. However, the scheduling of these systems presents a formidable challenge, requiring meticulous planning of pumping runs well in advance to meet the anticipated demands of clients. In this work, we enhance an existing literature model of a multiproduct pipeline system by introducing uncertainty in the customer demand. The problem is then addressed via a two-stage stochastic formulation. The typical drawback with stochastic formulations is the high computational burden required. To address this challenge, we adapt the so-called Similarity Index decomposition, resulting in a 28-fold improvement in CPU time while achieving equivalent solutions compared to solving the full-space problem.

Keywords: Decomposition, MILP, Uncertainty, Planning, Oil & Gas.

1. INTRODUCTION

Transportation of refined oil products constitutes a critical aspect of the energy supply chain, with several alternatives, including multiproduct pipelines, trains, and trucks. Among these, multiproduct pipelines are preferred, facilitating the transport of approximately 70% of oil products. The advantages of pipelines, such as cost-effectiveness and efficiency, make them integral to the industry logistics. However, the efficient operation of multiproduct pipelines is challenging. Sequencing batches poorly or misjudging injection volumes can lead to both contamination issues and unmet market demands. Given the intricate nature of these systems, the primary objective of pipeline scheduling emerges as a delicate balance between meeting market demands, ensuring safe operations, and minimizing costs.

Numerous studies have delved into pipeline scheduling, recognizing the need to incorporate uncertainty into the decision-making process (Li et al., 2021) in order to improve feasibility in practice. Despite its obvious need, introducing uncertainty via scenarios significantly amplifies the computational complexity of the problem. This issue leads to a considerable increment in the solution times, which poses a challenge given the need for timely decision-making in online scheduling setups (Li et al., 2021).

A usual approach to make less conservative decisions in the presence of uncertainty is the so-called two-stage stochastic optimization (TSSO) (Birge and Louveaux, 2011). In TSSO, the problem decision variables are split into two groups: the first-stage ones ($x \in \mathbb{R}$) that need to be made before actually knowing the future realization of the uncertain parameters or inputs; and the second-stage ones ($y_s \in \mathbb{R}$, where $s \in \mathcal{S}$ is the set of scenarios considered) which allow for adjusting near-future decisions by scenario once the uncertainty is revealed (e.g. actual demand will be known with precision when time arrives). As the number of variables of the optimization problem increases proportionally to the number of scenarios $|\mathcal{S}|$, it is well known that solving TSSO is NP-hard (Dyer and Stougie, 2006). Hence, TSSO can easily become computationally intractable for real industrial-scale problems, so alternative decomposition methods are often needed.

This paper tackles the computational challenges associated with two-stage stochastic pipeline scheduling. Our approach centers on a decomposition algorithm based on the similarity index (SI), aiming to streamline the decision-making process. The basis is to decompose the optimization problem into more tractable subproblems that can be solved independently (Montes et al., 2022, 2023a). Then, the SI links them all by measuring the similarity among the subproblem solutions, and an iterative procedure is set up to progressively increase it until all the subproblems are non-anticipative (the first-stage decisions are equal among all the scenarios). By breaking down the complexity into manageable components, we anticipate significant improvements in solution times, thus enhancing the applicability to industrial-size scheduling problems in

real time. Note that this proposal differs from our previous work on how the first-stage continuous variables are handled (the progressive hedging algorithm is not used). Here, an iterative procedure is established: first a non-anticipative solution to the first-stage is obtained using the SI algorithm; then such variables are fixed and the full-space problem (with non-anticipativity constraints) is solved; if the first-stage solution yields an infeasible full-space problem, such a solution is removed from the feasible region and the process is repeated.

To illustrate the proposed method, we take a literature model as the basis for the presented case study, showcasing the adaptability and applicability of our decomposition algorithm (Montes et al., 2023b, 2024) to address real-world pipeline scheduling problems. Through this research, we aim not only to contribute to the existing body of knowledge but also to provide a tangible and efficient solution to the challenges posed by uncertainty in multi-product pipeline scheduling.

The following section expands the multiproduct pipeline scheduling model to a two-stage stochastic formulation. Section 3 presents the adaptation of the SI decomposition algorithm to reduce the computational load for solving the posed problem. Section 4 summarizes the results gathered after solving an instance of the scheduling via the SI decomposition and the monolithic formulation. Finally, Section 5 provides some concluding remarks.

2. MULTIPRODUCT PIPELINE SCHEDULING MODEL

Research on multiproduct pipeline scheduling has been conducted for many years, and there are several models with different network topology and time representations (Li et al., 2021). This paper focuses on pipelines with a single source (a refinery) and multiple depots along their length that cannot supply products back to the pipeline.

Figure 1 shows a representation of the pipeline system considered, with depots, refinery, and delivering different product slugs. Moreover, every depot has several product tanks, whose levels must be always between some minimum and maximum limits. Each depots has its own product demand over time that must be fulfilled from the refinery. Note that there might be incompatibilities between products that cannot be pumped adjacently. Also, the volume of the interface between adjacent slugs depends on the products. There is a minimum and maximum length of each slug. Also note that the interface material cannot be pumped to any depot, but to the final one to avoid contamination of the products.

Cafaro and Cerdá (2004) proposed a scheduling model based on a continuous-time formulation for pipeline systems of such a topology. In brief, the set of slugs $i \in \mathcal{I}$ to be delivered to depots $j \in \mathcal{D}$ is divided into two: old slugs already inside the pipeline at the beginning of the scheduling horizon, $\mathcal{I}^{\text{old}} \subset \mathcal{I}$; and new slugs $\mathcal{I}^{\text{new}} \subset \mathcal{I}$ to be scheduled. Due to space constraints, further details on this model are not included here, and the reader is pointed to such a reference.

2.1 Two-Stage Stochastic Extension

The model proposed in Cafaro and Cerdá (2004) is deterministic. That is, the values of all parameters and demand forecasts are assumed to be known. This, however, is seldom true in actual practice. As reasoned in Section 1, recognizing and incorporating uncertainty in the decision-making process offers substantial advantages in terms of feasibility and optimality. Consequently, in this work uncertainty is considered in the product-demand forecasts ($qd_{p,j,s}$) at each depot. Hence, a set of scenarios $s \in \mathcal{S}$ associated with possible values of the product demands and their occurrence probability ($\theta_s \in (0,1]$, $\sum \theta_s = 1$) is introduced. Then, index s is added to all variables in the model of Cafaro and Cerdá (2004) to get a stochastic one.

Moreover, as actual demand will be revealed when time arrives (i.e. when a slug reaches a depot location), a two-stage stochastic formulation can be set up by choosing the following decisions (see Figure 1)

$$y_{i,p},\ x^{i'}_{i,p,j},\ A_{i,p},\ qm^{i'}_{p,j},\ ID^{i'}_{p,j},$$

as first-stage variables, where $p \in \mathcal{P}$ is the set of products (only one product per slug is allowed). The reader is referred to the nomenclature section at the end of the paper for the meaning of all model variables and parameters.

The choice is made according to the decisions that must be taken "here and now" regarding the current product to be injected, as the pipeline cannot be in standby until some future demands are revealed. Hence, the first stage in the proposed formulation concerns decisions linked to the pumping run of the next new slug i_1^{new} to be injected. The non-anticipation constraints (NACs) to fulfill are:

$$\begin{aligned}
&y_{i_1^{\text{new}},p,s} = y_{i_1^{\text{new}},p} && \forall p, s \\
&x^{i_1^{\text{new}}}_{i,j,s} = x^{i_1^{\text{new}}}_{i,j} && \forall i, j, s \\
&A_{i_1^{\text{new}},p,s} = A_{i_1^{\text{new}},p} && \forall p, s \\
&qm^{i_1^{\text{new}}}_{p,j,s} = qm^{i_1^{\text{new}}}_{p,j} && \forall p, j, s \\
&ID^{i_1^{\text{new}}}_{p,j,s} = ID^{i_1^{\text{new}}}_{p,j} && \forall p, j, s
\end{aligned} \tag{1}$$

This choice of the first and second stages variables allows for making the most imminent decision robustly, while giving a chance to adapt decisions scheduled in the future to the possible realization of the uncertain demand. This concept is analogous to the decisions a pipeline operator must face upon completing the pumping run of a slug. In this situation, the operator must navigate the immediate decision-making process, preparing subsequent actions to cover the potential future realizations of the uncertainty.

3. DECOMPOSITION ALGORITHM

3.1 Origins

The Similarity Index (SI) decomposition algorithm was recently proposed to decompose two-stage stochastic scheduling problems that are based on a discrete-time representation, and in which the first-stage only contains binary variables (Montes et al., 2022). The goal is to allow removing the non-anticipation constraints to enable a scenario-based decomposition where each scenario sub-problem is solved independently. Their solutions are later

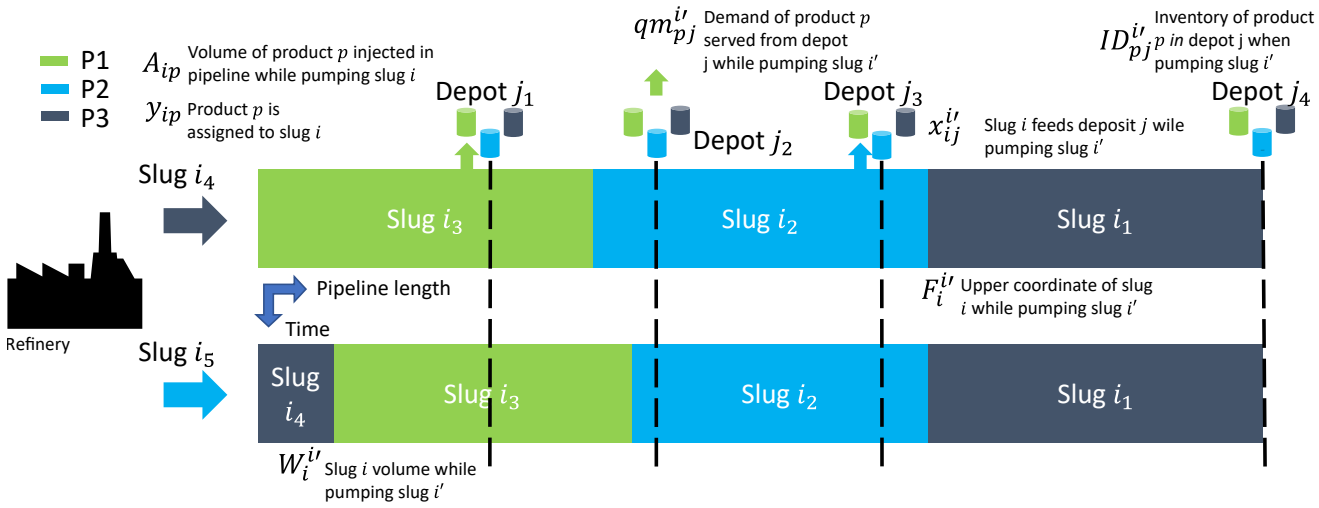

Fig. 1. Pipeline system considered and its associated model variables. Time evolution is represented by the rows, where slugs spatial location in the pipeline is depicted.

compared using the Similarity Index. An iterative procedure is set up so that the similarity of first-stage variables among subproblem solutions progressively increases until they are equal.

The Similarity Index builds upon the information theory ideas for assessing dissimilarity among sets of binarized data. Conceptually, it is defined as the size of the intersection divided by the size of the union of the sample sets. In the two-stage stochastic scheduling setting, the first-stage binary solutions from each scenario subproblem are compared via their intersection either per time period (discrete-time representation) or per time slot (continuous-time representation) in such a way. Then, the summation of such intersections over all time periods or slots belonging to the first-stage horizon is divided by the maximum intersection possible (i.e., the sets union). Therefore, the SI is always between 0 and 1 (100%). If SI=1, individual first-stage solutions are equal among all subproblems so the non-anticipation constraints are met indirectly.

Remark 1. With binary datasets, the size of the union is a fixed value, known in advance (e.g., the number of time periods in the first-stage horizon). This nice feature keeps the linearity of the mixed-integer formulation.

However, the SI algorithm as previously proposed in the authors' papers is not directly applicable to the pipeline scheduling problem because:

(i) It can only cope with binary variables in the first stage, but here $A_{i,p,s}$, $qm_{p,j,s}^{i'}$, and $ID_{p,j,s}^{i'}$ are continuous.
(ii) Despite being binary, it cannot deal either with $x_{i,p,j,s}^{i'}$, as the number of slugs that will feed a depot during a time horizon (i.e., the size of the sets union) is not known a priori nor can be fixed.

3.2 Adaptation to pipeline scheduling

Given the above listed issues, the proposed algorithm for the pipeline scheduling problem is based, in essence, on a *partial* scenario decomposition strategy, where the SI is employed to progressively enforce non-anticipativity just on a subset of the binary first-stage variables, specifically the assignment of products to slugs $y_{i_1^{\text{new}},p,s}$:

$$\text{SI} = \sum_{p\in\mathcal{P}} \min_{s\in\mathcal{S}} \left\{ y_{i_1^{\text{new}},p,s} \right\} \tag{2}$$

Algorithm 1 summarizes the iterative optimization strategy. Recall that only one product per slug is allowed.

Algorithm 1 Proposed SI Decomposition Algorithm

Require: $\rho, \text{k}_{\max}, \text{d}_{\max}$
$d \leftarrow 0$
repeat
  $k \leftarrow 0, \lambda^{(0)} \leftarrow 0$
  $\bar{y}_{i_1^{\text{new}},p} = \arg\min(3)|_{\text{s=s}_1}$
  repeat
    parallel for $s \in \mathcal{S}$ **do** ▷ Solve Subproblems
      $[y^*_{i_1^{\text{new}},p,s}; \psi^*_{i_1^{\text{new}},p}] \leftarrow \arg\min(3)$
      $SI^*_s \leftarrow \sum_p \psi^*_{i_1^{\text{new}},p}$
    end parallel for
    $\bar{s} \leftarrow \arg\min_{s\in\mathcal{S}} SI^*_s$
    $\bar{y}_{i_1^{\text{new}},p} \leftarrow y^*_{i_1^{\text{new}},p,\bar{s}}$ ▷ Worst local SI solution
    $SI \leftarrow SI(y^*_{i_1^{\text{new}},p,s})$ ▷ Global SI by Eq. (2)
    $\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \rho(SI - 1)$
    $k \leftarrow k + 1$
  until $SI = 1 \vee k = \text{k}_{\max}$
  if $SI = 1$ **then**
    $d \leftarrow d + 1$
    $\nu^*_s \leftarrow \arg\min(4)|_{y_{i_1^{\text{new}},p,s}=\bar{y}_{i_1^{\text{new}},p}}$
    if $\nu^*_s = \emptyset$ **then**
      Add (5) to Subproblems (3)
    else
      return ν^*_s ▷ Feasible solution found
    end if
  end if
until $d = \text{d}_{\max} \vee k = \text{k}_{\max}$

Initially, the non-anticipativity constraints (1) are disabled, leading to the scenario independent subproblems (3) with a modified objective function that depends of a coordination multiplier λ. Subsequently, the SI (2) is employed to progressively increase λ to enforce non-anticipativity.

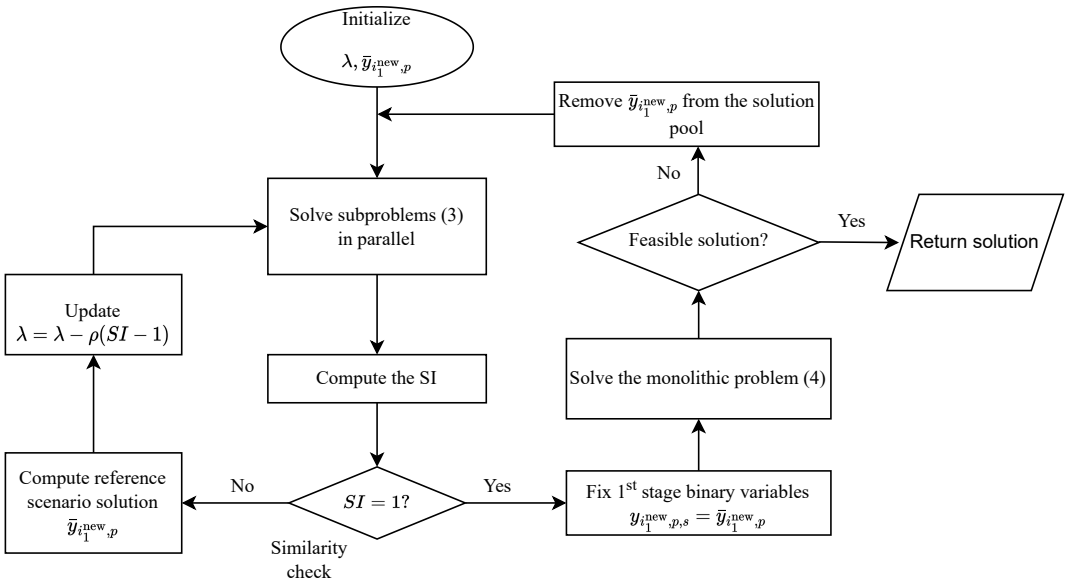

Fig. 2. Flowchart of the proposed SI decomposition algorithm.

Following this, variables $y_{i_1^{\text{new}},p,s}$ are fixed to the previously obtained values and the monolithic problem (4) is solved, now with the non-anticipativity constraints re-enabled. If feasible, it provides the reached solution values for all first and second stage variables, denoted by ν_s^*. In the event that (4) proves infeasible, the fixed values $\bar{y}_{i_1^{\text{new}},p}$ are removed from the solution pool by the addition of constraint (5) in subproblems (3), and the cycle is repeated. Figure 2 shows the described flowchart.

Each subproblem corresponding to a scenario s' is formulated as (3), where notation $\nu_{s'}$ stands for all the subproblem decision variables (first and second stage).

$$\begin{aligned} \min_{\nu_{s'},\, \psi_{i_1^{\text{new}},p}} \quad & z_{s'}(\nu_{s'}) - \lambda \cdot \text{SI}_{s'} && (3)\\ \text{s.t.} \quad & \text{Model \& process constraints,} \\ & \psi_{i_1^{\text{new}},p} \le y_{i_1^{\text{new}},p,s'} && \forall p, \\ & \psi_{i_1^{\text{new}},p} \le \bar{y}_{i_1^{\text{new}},p} && \forall p, \\ & \text{SI}_{s'} = \sum_p \psi_{i_1^{\text{new}},p} \end{aligned}$$

Where, $z_{s'} : \mathbb{R}^N \times \mathbb{B}^M \to \mathbb{R}$ is the objective function corresponding of scenario s'.

Remark 2. The SI computed by (2) cannot be incorporated into the optimization problem (3), because it requires values from all the scenarios (that would break decomposition) and the min{} operator would disrupt the problem linearity. Therefore, a *local* estimation of the SI per scenario is conducted using the set of binary slack variables denoted by $\psi_{i_1^{\text{new}},p} \in \mathbb{B}$. In this way, such a local $\text{SI}_{s'}$ only needs the binary variables specific to subproblem s' and a given *reference* solution, denoted by $\bar{y}_{i_1^{\text{new}},p}$. See Montes et al. (2022, 2023a) for a more in-depth explanation.

The monolithic counterpart can be formulated as in (4). The non-anticipation constraints are imposed on the first-stage variables that are not included in the SI computation, as discussed in the above section.

$$\begin{aligned} \min_{\nu_s} \quad & \sum_{s\in\mathcal{S}} \theta_s z_s(\nu_s) && (4)\\ \text{s.t.} \quad & \text{Model \& process constraints,} \\ & \text{Non-anticipation constraints (1)} \end{aligned}$$

To remove a given set of values $y_{i_1^{\text{new}},p,s}^{(d-1)}$ from the pool of candidate solutions (to restrict the feasible region), constraint (5) is added to (3).

$$\sum_{s\in\mathcal{S}} y_{i_1^{\text{new}},p,s}^{(d)}\Big|_{y_{i_1^{\text{new}},p,s}^{(d-1)}=0} + \sum_{s\in\mathcal{S}} \left(1 - y_{i_1^{\text{new}},p,s}^{(d)}\right)\Big|_{y_{i_1^{\text{new}},p,s}^{(d-1)}=1} \ge 1 \quad (5)$$

In the context of Algorithm 1, d is the outer iteration counter. Note that if $y_{i_1^{\text{new}},p,s}^{(d)} = y_{i_1^{\text{new}},p,s}^{(d-1)}\ \forall s,p$, then the left-hand side of (5) is zero.

To update the multiplier λ, a formula inspired in the subgradient method can be used:

$$\lambda = \lambda - \rho\,(SI - 1) \quad (6)$$

Where ρ is the *single* tuning parameter in Algorithm 1.

4. RESULTS

To evaluate the effectiveness of the proposed decomposition algorithm, we conducted tests using an instance of the model featuring five depots, four products, and four scheduled refinery production runs over a 75-hour horizon. Detailed information regarding various parameters and problem initialization can be found in Cafaro and Cerdá (2004), Example 1.

To introduce a realistic uncertainty factor, we considered future product demands at each depot to be not known with precision but varying around ±20% of their nominal values. Eleven scenarios ($s \in \mathcal{S}$) were generated to represent variations in product demand from the nominal values, each associated with specific probabilities. These scenarios, along with their probabilities, are detailed in Table 1. It is worth noting that the careful selection of

scenarios and their associated values and probabilities is pivotal in formulating a two-stage problem, with options ranging from historical data to forecasts from sales departments. Note, however, that this matter is outside the scope of this paper, so the values presented in Table 1 were arbitrarily chosen within realistic operational limits.

Table 1. Uncertainty scenarios and their associated probabilities θ_s.

Scenarios	Demand variation	Probability
s_1	0%	20%
s_2	+2.5%	2.5%
s_3	+5%	12.5%
s_4	+10%	10%
s_5	+15%	7.5%
s_6	+20%	7.5%
s_7	−2.5%	2.5%
s_8	−5%	12.5%
s_9	−10%	10%
s_{10}	−15%	7.5%
s_{11}	−20%	7.5%

The full-space problem ended up in 26220 constraints, 17821 continuous variables, and 2563 binary variables.

The model was coded in GAMS 45.2.0 and solved using Gurobi 10.0.3 with a convergence criterion set to 0.01% optimality gap in all cases. Of course, the tests were all run on the same machine, featuring an Intel i9-13900K CPU and 128GB of DDR5 RAM. The results achieved by solving the optimization problem via the standard full-space monolithic formulation and via Algorithm 1 are summarized in Table 2. Notably, the SI decomposition algorithm exhibited a remarkable performance, solving the problem in less than a minute and proving to be approximately 28 times faster than solving the full-space monolithic optimization. This swift computational time is particularly advantageous for conducting what-if studies or facilitating rescheduling efforts. Note importantly that the objective values reached by both approaches fall within the specified optimality gap, indicating that they can be considered essentially equal.

Table 2. Comparison between the full-space problem and the decomposition algorithm.

	CPU Time (s)	Objective Value
Full-Space Problem	630.92	150235.73
Decomp. Algorithm	22.87	150237.21

Furthermore, the solution obtained by Algorithm 1 was insensitive to the chosen value for ρ in (6). This fact can be explained by the few variables over which the SI is computed for this case study. However, previous results from other case studies showed minor sensitivity of the objective value concerning ρ (Montes et al., 2024).

Figure 3 shows a Gantt diagram of the solution provided by Algorithm 1 for scenarios s_1, s_6 and s_{11}. Observe that non-anticipation is evident for slugs i_1^{new} as the solution is equal across all scenarios. From there on, the decisions are different to accommodate the different customer demands considered in each scenario.

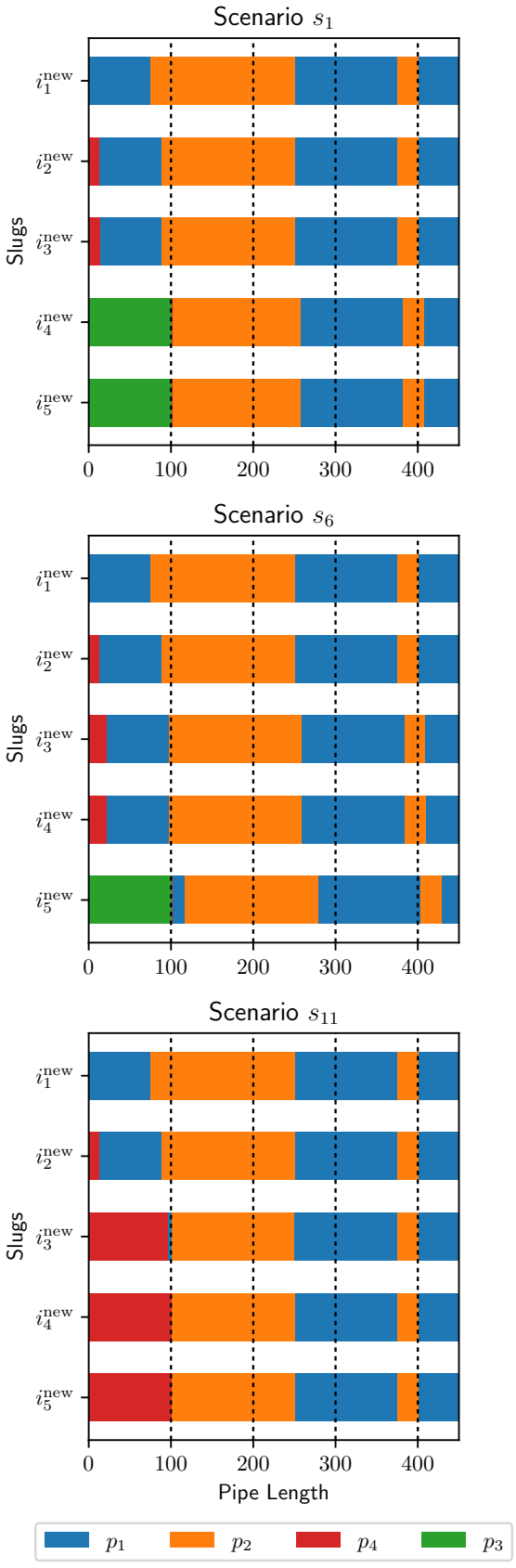

Fig. 3. Gantt diagram of the solution obtained using the decomposition algorithm. The vertical black dotted lines represent the depot locations.

5. CONCLUSIONS

The first contribution of this paper was extending a literature multiproduct pipeline scheduling model to a two-stage stochastic formulation, in which the uncertain variables are the future customer demands. As this drastically increases the computational complexity of the optimization, a second contribution is to adapt the authors' Similarity-Index decomposition to this class of problems. The key idea of this method is relying on the Similarity Index to enforce non-anticipativity on the complicating binary variables. Once this condition is met and such variables are fixed, the resulting monolithic problem is solved way faster, despite including non-anticipation in the remaining first-stage variables.

The test results showed a significant decrease in the required CPU time to get an economically identical solution to the one obtained when solving via the direct monolithic formulation. Although it is a conceptually simple idea, the results are promising for practical purposes: the method is easy to implement, and there is a single tuning parameter (whose choice does not seem critical).

Nonetheless, these preliminary results require further testing for different initialization conditions, longer time horizons, and considering more deposits and products. In addition, future work may expand the model to include truck delivery of some low-volume specialty products to the customer sites.

ACKNOWLEDGEMENTS

These results are funded by the Spanish MCIN/AEI/ 10.13039 /501100011033/, as part of the a-CIDiT (PID2021-123654OB-C31) and LOCPU (PID2020-116585GB-I00) research projects. The first author has received financial support from the 2020 call of pre-doctoral contracts of the University of Valladolid and Banco Santander.

NOMENCLATURE

Sets

$i \in \mathcal{I}$ Slugs. Can either be old $i^{\text{old}} \in \mathcal{I}^{\text{old}}$ (inside of the pipe at the beginning of the horizon, or new $i^{\text{new}} \in \mathcal{I}^{\text{new}}$. Hence, $\mathcal{I}^{\text{old}} \cup \mathcal{I}^{\text{new}} = \mathcal{I}$.
$j \in \mathcal{J}$ Depots.
$p \in \mathcal{P}$ Products.
$s \in \mathcal{S}$ Scenarios.

Parameters

ρ Step size for updating λ.
θ_s Ocurrence probability of scenario s.
$qd_{p,j,s}$ Demand of product p in depot j in scenario s.

Decision Variables

ν_s Full set of decision variables of scenario s.
$A_{i,p,s}$ Volume of product p injected in the pipeline while pumping slug i.
$ID^{i'}_{p,j,s}$ Inventory of product p in depot j when pumping slug i'.
$qm^{i'}_{p,j,s}$ Demand of product p served from depot j while pumping slug i'.
$x^{i'}_{i,j,s}$ 0-1 variable to represent that slug i feeds depot j in scenario s while pumping slug i'.
$y_{i,p,s}$ 0-1 variable to assign product p to slug i in s.
z_s Objective value of scenario s.

Other Variables

$\bar{y}_{i_1^{\text{new}},p}$ Reference solution towards which the SI is computed.
λ Similarity Index weigh in the objective function.
SI_s Local Similarity Index for scenario s.
SI Similarity Index.
$\psi_{i_1^{\text{new}},p}$ Slack variable for computing the local SI.
d Outer iteration counter.
k Inner iteration counter.

REFERENCES

Birge, J.R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer, New York, 2nd edition.

Cafaro, D.C. and Cerdá, J. (2004). Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. *Computers & Chemical Engineering*, 28(10), 2053–2068. doi:10.1016/j.compchemeng.2004.03.010.

Dyer, M. and Stougie, L. (2006). Computational complexity of stochastic programming problems. *Math. Programming*, 106, 423–432. doi:10.1007/s10107-005-0597-0.

Li, Z., Liang, Y., Liao, Q., Zhang, B., and Zhang, H. (2021). A review of multiproduct pipeline scheduling: from bibliometric analysis to research framework and future research directions. *Journal of Pipeline Science and Engineering*, 1(4), 395–406. doi:https://doi.org/10.1016/j.jpse.2021.08.001. Special Issue on Smart Operation and Management of Pipelines.

Montes, D., Pitarch, J.L., and De Prada, C. (2022). The Similarity Index to Decompose Two-Stage Stochastic Scheduling Problems. *IFAC-PapersOnLine*, 55(7), 821–826. doi:10.1016/j.ifacol.2022.07.546.

Montes, D., Pitarch, J.L., and De Prada, C. (2023a). Decomposing Two-Stage Stochastic Scheduling Problems on a continuous-time basis via Slot Similarity. *IFAC-PapersOnLine*, 56(2), 10168–10173. doi:10.1016/j.ifacol.2023.10.892.

Montes, D., Pitarch, J.L., and De Prada, C. (2023b). Extending the si decomposition to continuous-time two-stage scheduling problems. volume 52 of *Computer Aided Chemical Engineering*, 499–504. doi:10.1016/B978-0-443-15274-0.50080-9.

Montes, D., Pitarch, J.L., and de Prada, C. (2024). Similarity-based Decomposition Algorithm for Two-stage Stochastic Scheduling. *Optimization Online*. URL `https://optimization-online.org/?p=25958`.