Computers and Chemical Engineering 193 (2025) 108924

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Data-driven robust optimization for pipeline scheduling under flow rate uncertainty

Amir Baghban [a], Pedro M. Castro [b], Fabricio Oliveira [*,c]

[a] Department of Mathematics, Azarbaijan Shahid Madani University, 5375171379, Tabriz, Iran
[b] CERENA. Department of Chemical Engineering, Instituto Superior Técnico, University of Lisbon, 1049-001 Lisboa, Portugal
[c] Department of Mathematics and Systems Analysis, School of Science, Aalto University, P.O. Box 11100, 00076 AALTO, Finland

ARTICLE INFO

Keywords:
Straight liquid pipelines
Continuous-time formulation
Mixed-integer linear programming
Support vector clustering
Robust optimization

ABSTRACT

Frequently, parameters in optimization models are subject to a high level of uncertainty coming from several sources and, as such, assuming them to be deterministic can lead to solutions that are infeasible in practice. Robust optimization is a computationally efficient approach that generates solutions that are feasible for realizations of uncertain parameters near the nominal value. This paper develops a data-driven robust optimization approach for the scheduling of a straight pipeline connecting a single refinery with multiple distribution centers, considering uncertainty in the injection rate. For that, we apply support vector clustering to learn an uncertainty set for the robust version of the deterministic model. We compare the performance of our proposed robust model against one utilizing a standard robust optimization approach and conclude that data-driven robust solutions are less conservative.

1. Introduction

Around the clock, refined product pipelines ship huge volumes of oil derivatives over long distances from production to distribution sites, making transportation planning crucial for the oil supply chain. Pipelines bring multiple benefits to the oil industry because they are the safest, cleanest, and most economical means of transportation, conveying roughly 70 % of oil products in the US (Cafaro and Cerdá, 2008a). They also contribute to sustainability as they generate fewer CO_2 emissions than rail and road. Furthermore, refined product pipelines are multiproduct systems, transporting different types of oil derivatives (e.g., gasoline, diesel, home heating oil, and kerosene) in the same duct, in batches. Unique features of pipelines compared to other transportation means are that they are always full and there is no physical separation between adjacent products. The latter is responsible for volume contamination at the interface that grows with the distance travelled. Some interfaces are directed to depots that contain low-degree products (e.g., the interface between premium gasoline and regular gasoline is directed to regular gasoline depots), whereas contaminated volumes are discharged into separate tanks and sent back for reprocessing (by tanker trucks).

In the pipeline scheduling problem (PSP), the goal is to generate a detailed plan that meets product demand at the distribution centers (DCs) at the lowest cost. A detailed scheduling answers the following questions that have been addressed by Mostafaei et al. (2021a) and others: When/what/how much to inject into the pipeline? When/what/how much to deliver from the pipeline to a specific DC? What should the inventory profiles at tanks be to avoid overflow or running out of stock? In the different segments of the pipeline, flow rate limitations must be respected and since restart costs are high, stoppages should be avoided. To prevent schedules with large interfaces, forbidden product sequences are often enforced (e.g., gasoline cannot be next to gasoil).

The PSP can be classified according to the topology, ranging from straight pipelines with a single input and output node studied by, e.g., Dimas et al. (2018), Kirschstein (2018) and Cafaro and Cerdá (2008a) to tree-like pipelines with multiple input and output nodes, which were studied by e.g., Castro and Mostafaei (2019) and Mostafaei et al. (2015a). Another criterion is the flow direction. Most articles assume unidirectional flow, while Cafaro and Cerdá (2014) and Castro (2017a) tackled bidirectional pipelines. Most nodes along the pipeline system are single-purpose, but Castro and Mostafaei (2017b) and Cafaro et al. (2015) studied dual-purpose intermediate nodes that increase flexibility by alternating the reception and delivery of material from/into the pipeline. Time representation is also important. In discrete-time models by Chen et al. (2017), Herran et al. (2010), Magatão et al. (2004) and Rejowski and Pinto (2003), the time horizon is divided into time periods

* Corresponding author.
E-mail address: fabricio.oliveira@aalto.fi (F. Oliveira).

https://doi.org/10.1016/j.compchemeng.2024.108924
Received 15 August 2024; Received in revised form 5 October 2024; Accepted 10 November 2024
Available online 14 November 2024

Nomenclature

Sets/indices

I	Batches
K	Pumping runs
J	Distribution centers
P	Products
A	Set indexing the variables of Constraint (32)
B	Set indexing support vectors

Parameters

$\delta_{\|J\|}$	Total volume of the pipeline
δ_j	Location of distribution center j on the pipeline
vr^{min}	Minimum allowable volume of a batch injected into the pipeline during a run
vr^{max}	Maximum allowable volume of a batch injected into the pipeline during a run
$vs_j^{\min}$	Minimum allowable flow rate through segment j
$vs_j^{\max}$	Maximum allowable flow rate through segment j
M	A sufficiently large constant
v_p^{min}	Minimum injection rate for product p
v_p^{max}	Maximum injection rate for product p
$dem_{p,j}$	Demand for product p in depot j
$BM_{p,p'}$	Binary parameter indicating whether product p and p' can be injected consecutively
Q	Weighting matrix of the generated data
θ	A parameter introduced in the robust counterpart model
M_p	The mean value for the $v_1v_2\ldots v_{p-1}v_{p+1}\ldots v_{\|P\|}$ component of the data points
G	The mean value for the $-v_1v_2\ldots v_{\|P\|}$ component (first component) of the data points
$DI1_p$	Difference between minimum and maximum values for $v_1v_2\ldots\widehat{v}_p\ldots v_{\|P\|}$ component in data set
$DI2$	Difference between minimum and maximum values for $-v_1v_2\ldots v_{\|P\|}$ component in data set

Continuous Variables (nonnegative)

MS	Makespan
S_k	Start time of pumping run k
L_k	Duration of pumping run k
$F_{i,k}$	Right coordinate of batch i at the completion time of run k
$W_{i,k}$	Volume of batch i at the completion time of run k
$VPR_{i,p,k}$	Volume of product p injected in batch i during run k
$VPD_{i,p,j,k}$	Volume of product p delivered from batch i to depot j during run k
$VR_{i,k}$	Volume of batch i injected during run k
$VD_{i,j,k}$	Volume of batch i delivered to depot j during run k
$ID_{p,j,k}$	Volume of product p in depot j at time k
$LC_{p,j,k}$	Volume of product p shipped from depot j to local markets during run k
$\mu_{a,b,k}$	Variable of data-driven robust model
$\lambda_{a,b,k}$	Variable of data-driven robust model
η_k	Variable of data-driven robust model
$CV1_{i,p,k}$	Variable of the Γ-robust model
$CV2_k$	Variable of the Γ-robust model
$CV3_k$	Variable of the Γ-robust model
$CV4_{i,p}$	Variable of the Γ-robust model
$CV5$	Variable of the Γ-robust model

Binary variables

$Y_{i,p}$	Binary variable indicating batch i contains product p
$X_{i,k}^{ref}$	Binary variable indicating batch i is injected into the pipeline during run k
$X_{i,j,k}^{dep}$	Binary variable indicating batch i is delivered to distribution center j during run k
$X_{j,k}^{seg}$	Binary variable indicating pipeline segment j is active during run k

of equal size and the events (e.g., the start time of a new product injection) coincide with a subset of the interval boundaries. In continuous-time models developed by Mostafaei et al. (2021a) and Moradi et al. (2019), the number of events is specified by the user, with the optimization determining their preferred location. The main drawback of discrete-time models is their larger size, whereas their main advantage is that some model constraints can be handled linearly.

The first PSP formulation was developed by Rejowski and Pinto (2003), consisting of a discrete-time MILP model suitable for a straight system with one refinery and multiple terminals. Cafaro and Cerdá (2004, 2008b) presented a continuous-time model for the same topology, which was later extended to systems with multiple refineries and DCs by Cafaro and Cerdá (2009, 2010), structured as a tree (Cafaro and Cerdá, 2011), or as a mesh (Cafaro and Cerdá, 2012). Over the last two decades, researchers such as Castro and Mostafaei (2017b, 2019), Liao et al. (2019b), Dimas et al. (2018), Ghaffari-Hadigheh and Mostafaei (2015), Mostafaei et al. (2015a, 2015b, 2016, 2021a, 2021b), Zaghian and Mostafaei (2016) have extended the scope of PSP by developing a variety of formulations that can handle real-life constraints in a computationally efficient manner.

The above articles share the assumption that all data is deterministic, whereas in real-world applications, parameters are subject to uncertainty. The realization of the uncertainty may render an operational plan useless, forcing the decision-makers to reschedule operations. For instance, a drop in the injection flow rate, which has a probability of occurrence of 17 %, according to Muhlbauer (2004), may lead to major delays in demand fulfillment. Incorporating flow rate uncertainty in optimization models will lead to more robust solutions, mitigate the need for rescheduling efforts and improve service levels.

The two main adopted methodologies to address uncertainty are stochastic and robust optimization. In stochastic optimization, uncertainty is incorporated by using either a discrete (known as scenario-based approach) or a continuous probability distribution, despite the real distribution function being often unknown. Chatterjee and Chowdhury (2017) showed that for the discrete distribution, the number of scenarios enormously increases with the number of uncertain parameters. The disadvantage of using a continuous probability distribution is that it leads to intractable nonlinear optimization problems. In robust optimization, uncertain parameters take their values from an uncertainty set, with each realization corresponding to a different optimization problem. The aim is then to find a solution that remains feasible for all possible realizations and optimal for the worst-case value of the objective function. However, such a solution may be too conservative to implement in practice.

Asl and MirHassani (2019) tackled the uncertainty in pipeline injection flow rate due to pump failure, with a two-stage stochastic programming approach. The case study is a straight line composed of a single refinery and a receiving node and the aim is to schedule it for long horizon times: from 10 to 30 days. Due to a long scheduling horizon and the approach chosen to model the uncertainty being based on multiple scenarios, the problem becomes computationally hard to be dealt with.

While robust optimization is a computationally efficient methodology to deal with uncertainty, its application for pipeline scheduling is rare (Li et al., 2021). Moradi and Mirhassani (2016) applied

Γ-robustness (Bertsimas and Sim, 2004) to schedule a pipeline comprised of a single input and output node, aiming at finding feasible solutions for most of the demand scenarios by varying the budget parameter. Their results show that robustness can be increased up to a desired level without significantly affecting the complexity of the model. However, the Γ-robustness approach does not leverage the underlying statistical information of the historical data as much as possible. Therefore, in this paper, we aim at developing a data-driven approach to construct the uncertainty set.

From the machine learning (ML) perspective, building such a data-driven uncertainty set falls into unsupervised and pattern recognition problems. ML has itself diverse learning models (e.g., kernel density estimation (Bishop (2006)) but because the conventional kernels contain nonlinear terms, major computational challenges arise. For instance, the radial basis (RBF) and the sigmoid functions may lead to intractable robust counterpart formulations. To overcome this problem, Shang et al. (2017) used linear kernel-based Support Vector Clustering (SVC). Their approach has a few crucial benefits from a practical standpoint. It: (i) ensures that the uncertainty set is convex; (ii) preserves the linearity of the deterministic problem in the case of linear programming (LP) problems; and (iii) considers the correlation between uncertain parameters.

Shang et al. (2017) approach has been adopted by several authors in their LP or MILP models. Mohseni and Pishvaee (2020) used it to deliver robust decisions at a lower cost, compared to conventional methods, when considering the multiperiod optimization of a wastewater sludge-to-biodiesel supply chain. Qiu et al. (2020) developed a mathematical model for multi-product inventory optimization under demand uncertainty. They concluded that the proposed data-driven robust optimization approach offers improved protection against demand uncertainty compared to box and ellipsoid uncertainty sets.

In this paper, we use a data-driven robust optimization approach to handle flowrate uncertainty in pipelines. The formation of the uncertainty set is automatically constructed with the linear kernel SVC and is interpretable against the number of data points it covers. Then, the formulation of the robust counterpart is automatically done by implementing the algorithm in Shang et al. (2017). Unlike the Γ-robust method that uses a single predefined shape for every problem and dataset, our data-driven robust approach uses historical data points to automatically and efficiently define an uncertainty set. It ensures both robustness and optimality by tuning a hyperparameter, which adjusts the uncertainty set size around the mean value. This data-fitting uncertainty set ensures the worst-case scenario is one of the historical data points, providing better solutions at all robustness levels.

The article is organized as follows: Section 2 outlines the problem, followed by the development of a deterministic continuous-time model in Section 3. Section 4 discusses synthesizing data for uncertain injection rates and introduces the support vector clustering method for forming a data-driven uncertainty set. This section also gives the robust counterparts of Γ-robust and data-driven optimization approaches for pipeline scheduling with uncertain injection rates. Section 5 validates the deterministic model by comparing it with two other continuous time models from the literature. Then, Section 6 demonstrates the efficiency of the robust solutions generated by the data-driven approach compared to those of the Γ-robust approach, before the conclusions in section 7.

2. Problem statement

In the pipeline scheduling problem (PSP), refined oil products $p \in P$ are transported from a single refinery (input node, located at one end of the pipeline) to multiple distribution centers $j \in J$ (output nodes) along the line, at volumetric coordinates δ_j. The goal is to meet product demand $dem_{p,j}$ as quickly as possible, i.e., to minimize the makespan. The following assumptions are made:

1) The pipeline is always full. Since we are dealing with incompressible fluids (liquids), the volume injected by the refinery must be equal to the total volume delivered to distribution centers.
2) Simultaneous delivery to multiple distribution centers, not necessarily involving the same product, is possible.
3) The flow rate inside the pipeline is subject to given lower vs_j^{min} and upper bounds $vs_j^{\max}$, which may vary between segments. Note that the number of segments is equal to the number of output nodes and so the same index is used.
4) The initial inventory level at the refinery, In_p, is known for all products.
5) Certain product sequences are forbidden ($BM_{p,p'} = 0$).

3. Deterministic scheduling formulation

The deterministic formulation below can be seen as an improvement of the continuous-time batch-centric formulation of Ghaffari-Hadigheh and Mostafaei (2015), which models the transportation of products as the movement of batches $i \in I$. Following recent developments in the literature (Liao et al., 2019a), we propose an alternative version of the model that allows for multiple batches to be injected by the refinery during a pumping run $k \in K$ (which represents a time slot) and delivered to the farthest active output node. The model constraints are briefly described in the next sections.

3.1. Objective function

The objective function is to minimize the makespan:

$$\min MS \tag{1}$$

3.2. Timing constraints

The start of pumping run k, S_k, should occur only after completion of the previous run. In Eq. (2), L_{k-1} represents the duration of run $k-1$ and $S_0 = 0$. Eq. (3) states that the completion time of the last pumping run cannot exceed the makespan.

$$S_k \geq S_{k-1} + L_{k-1} \;\forall\; k \in K \tag{2}$$

$$S_{|K|} + L_{|K|} \leq MS \tag{3}$$

3.3. Batch coordinates

We assume that the movement is from left to right on the diagrams and that batch $i+1$ enters the pipeline after batch i. Let $F_{i,k}$ be the right coordinate of batch i and $W_{i,k}$ the batch volume inside the pipeline at the completion time of run k. Then, the right coordinate of batch $i+1$ is equal to the right coordinate of i minus the volume of i, as stated by Eq. (4). Another way to compute the right coordinate of i is to add the volumes of all batches $i' \geq i$, as can be seen in Eq. (5).

$$F_{i+1,k} = F_{i,k} - W_{i,k},\ \forall\ i \in I, i < |I|,\ k \in K \tag{4}$$

$$F_{i,k} = \sum_{i' \geq i} W_{i',k},\ \forall\ i \in I,\ k \in K \tag{5}$$

3.4. Relation between batches and products

Batch i is assigned to product p by making binary variable $Y_{i,p} = 1$ (note that the volumes and batch-product assignments of batches initially inside the pipeline are known a priori). Eq. (6) declares that each batch can be of a single product, while Eq. (7) prevents forbidden sequences at the injection point. If the batch does not hold product p, then no volume of product p can be injected nor delivered through that batch, as stated by Eqs. (8)-(9), where M is a sufficiently large volume.

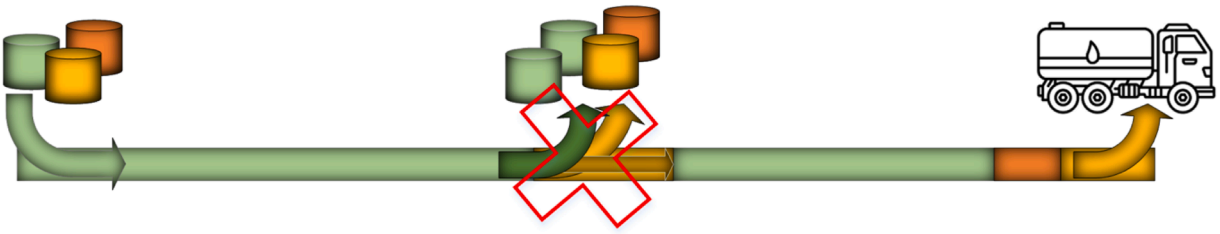

Fig. 1. Batch centric models like the current one, do not allow an intermediate depot to receive multiple batches during an injection run if the next segment is active.

$$\sum_{p\in P} Y_{i,p} = 1,\ \forall\, i \in I \tag{6}$$

$$Y_{i,p} + Y_{i+1,p'} \le 1 + BM_{p,p'},\ \forall\, i \in I, i < |I|,\ p,p' \in P, p \neq p' \tag{7}$$

$$\sum_{k\in K} VPR_{i,p,k} \le M.Y_{i,p},\ \forall\, i \in I,\ p \in P \tag{8}$$

$$\sum_{k\in K}\sum_{j\in J} VPD_{i,p,j,k} \le M.Y_{i,p},\ \forall\, i \in I,\ p \in P \tag{9}$$

Eq. (10) says that the total volume injected in the pipeline from the refinery storage tanks cannot exceed the initial inventory levels In_p. Since a batch will hold a single product, the volume leaving the refinery through batch i during run k can be computed by summing all product volumes, as seen in Eq. (11). The same can be said for the volume entering the tanks of a distribution center j, Eq. (12). The volume inside the dedicated product tanks of a distribution center increases when receiving material from the pipeline and decreases when shipping the product to local markets ($LC_{p,j,k}$), as stated by Eq. (13). Then, the total shipments must be sufficient to meet demand.

$$\sum_{i\in I}\sum_{k\in K} VPR_{i,p,k} \le In_p,\ \forall\, p \in P \tag{10}$$

$$VR_{i,k} = \sum_{p\in P} VPR_{i,p,k},\ \forall\, i \in I,\ k \in K \tag{11}$$

$$VD_{i,j,k} = \sum_{p\in P} VPD_{i,p,j,k},\ \forall i \in I,\ j \in J, k \in K \tag{12}$$

$$ID_{p,j,k} = ID_{p,j,k-1} + \sum_{i} VPD_{i,p,j,k} - LC_{p,j,k},\ \forall p \in P,\ j \in J, k \in K \tag{13}$$

$$\sum_{k} LC_{p,j,k} \ge dem_{p,j},\ \forall p \in P,\ j \in J \tag{14}$$

3.5. Volumetric balances

The global volumetric balance in Eq. (15) tells us that the volume injected is equal to the sum of the volumes delivered. Eq. (16) states that the total volume of batches inside the pipeline at the end of every run k is equal to the pipeline volume $\delta_{|J|}$. The injection of batch i during run k will increase its size, whereas delivery to a depot will reduce it, as can be seen in Eq. (17).

$$\sum_{i\in I} VR_{i,k} = \sum_{i\in I}\sum_{j\in J} VD_{i,j,k},\ \forall\, k \in K \tag{15}$$

$$\sum_{i\in I} W_{i,k} = \delta_{|J|},\ \forall\, k \in K \tag{16}$$

$$W_{i,k} = W_{i,k-1} + VR_{i,k} - \sum_{j\in J} VD_{i,j,k}\ \forall\, i \in I,\ k \in K \tag{17}$$

3.6. Triggering batch injection

Let binary variable $X^{ref}_{i,k} = 1$ indicate that batch i is injected in the pipeline during run k. For this injection to take place, the left coordinate of i at the start of k (computed as $F_{i,k-1} - W_{i,k-1}$) must be equal to zero, as stated by Eq. (18). Notice that $X^{ref}_{i,k} = 1$ implies $F_{i,k-1} - W_{i,k-1} = 0$, with Eq. (19) forcing the injected volume $VR_{i,k}$ to belong to the given interval $[vr^{min}, vr^{max}]$. In contrast, $X^{ref}_{i,k} = 0$ relaxes the constraint, making it possible for the left coordinate to take any value between 0 and the pipeline volume $\delta_{|J|}$.

$$F_{i,k-1} - W_{i,k-1} \le \delta_{|J|}\left(1 - X^{ref}_{i,k}\right),\ \forall\, i \in I,\ k \in K \tag{18}$$

$$vr^{min} X^{ref}_{i,k} \le VR_{i,k} \le vr^{max} X^{ref}_{i,k},\ \forall\, i \in I,\ k \in K \tag{19}$$

3.7. Triggering batch delivery

Let binary variable $X^{dep}_{i,j,k} = 1$ indicate that batch i is delivered to distribution center j during run k. This is possible only if the left coordinate of i at the end of k has not passed (isn't to the right of) the distribution center, and the right coordinate of i has reached it, as can be seen in Eqs. (20)-(21), respectively. Activating the depot will allow for the delivery of a certain batch volume, Eq. (22).

$$F_{i,k} - W_{i,k} \le \delta_j + \left(\delta_{|J|} - \delta_j\right)\left(1 - X^{dep}_{i,j,k}\right),\ \forall\, i \in I,\ j \in J,\ k \in K \tag{20}$$

$$F_{i,k} \ge \delta_j X^{dep}_{i,j,k},\ \forall\, i \in I,\ j \in J,\ k \in K \tag{21}$$

$$vr^{min} X^{dep}_{i,j,k} \le VD_{i,j,k} \le vr^{max} X^{dep}_{i,j,k},\ \forall\, i \in I,\ j \in J,\ k \in K \tag{22}$$

Eqs. (20)-(21) will allow multiple batches to be delivered to a distribution center during a run. As discussed in Liao et al. (2019a) and illustrated in Fig. 1, this may result in infeasible solutions unless nonlinear equations are used or the downstream segment is idle. Eq. (23) ensures that only the farthest active center is allowed to receive material from multiple batches during a run.

$$F_{i,k-1} \ge \delta_j X^{dep}_{i,j,k} - \delta_{|J|}\left(1 - X^{seg}_{j+1,k}\right),\ \forall\, i \in I,\ j \in J,\ k \in K \tag{23}$$

Eq. (24) limits the volume of batch i that can be delivered to depot j during run k.

$$VD_{i,j,k} \le VR_{i,k} + \left(\delta_j - F_{i+1,k-1}\right) + vr^{\max}\left(1 - X^{dep}_{i,j,k}\right),\ \forall i \in I, j \in J, k \in K \tag{24}$$

3.8. Logic constraints

If the refinery is injecting batch i, then the first pipeline segment (connecting the refinery to the first distribution center) must be active. This logic proposition can be reformulated into the inequality constraint in Eq. (25). On the other hand, if no batch of i is being injected, then the first segment will be idle, leading to Eq. (26). Since the only input node is

the refinery, if segment $j+1$ is active, so will segment j, as seen in Eq. (27). Eq. (28) states that if the distribution center located at the end of segment j is receiving a batch during run k, then the segment must be active. Finally, if segment j is active, then at least one batch must be delivered to a downstream distribution center.

$$X^{seg}_{1,k} \geq X^{ref}_{i,k},\ \forall\, i \in I,\ k \in K \tag{25}$$

$$X^{seg}_{1,k} \leq \sum_{i \in I} X^{ref}_{i,k},\ \forall\, k \in K \tag{26}$$

$$X^{seg}_{j,k} \geq X^{seg}_{j+1,k},\ \forall\, j \in J, j > 1,\ k \in K \tag{27}$$

$$X^{seg}_{j,k} \geq X^{dep}_{i,j,k},\ \forall\, i \in I, j \in J,\ k \in K \tag{28}$$

$$X^{seg}_{j,k} \leq \sum_{i \in I} \sum_{j' \geq j} X^{dep}_{i,j',k},\ \forall\, j \in J,\ k \in K \tag{29}$$

3.9. Flow rate limitations

During run k, the volume going through segment j is equal to the total volume delivered to downstream depots. Considering that run k lasts L_k hours, the minimum and maximum flow rate limits of operation can be enforced through Eq. (30).

$$L_k\, vs^{min}_j - M\left(1 - X^{seg}_{j,k}\right) \leq \sum_{i \in I} \sum_{j' \geq j} VD_{i,j',k} \leq L_k\, vs^{max}_j,\ \forall j \in J, k \in K \tag{30}$$

Let us now assume that product p is injected into the pipeline at a certain flow rate v_p. Since there are no constraints related to these variables, it suffices to guarantee that the actual flow rate is within given lower v^{min}_p and upper bounds v^{max}_p. This is ensured by Eq. (31), where the summation accounts for the possibility of multiple batch injections per pumping run.

$$\sum_{i \in I, p \in P} \frac{VPR_{i,p,k}}{v^{max}_p} \leq L_k \leq \sum_{i \in I,\ p \in P} \frac{VPR_{i,p,k}}{v^{min}_p},\ \forall\, k \in K \tag{31}$$

4. Synthetic data set generation

The product flow rates at the injection points are subject to uncertainty due to a number of reasons, including: (i) the electric motors powering the pumps may fail or need to undergo maintenance operations; (ii) demand variability and their influence on product inventory inside storage tanks at the refinery or distribution centers can limit the amount of the product being injected; (iii) the volumes of the products inside the pipeline (of different densities) and the number of active segments also impact the maximum flow rate that can be achieved.

The only set of constraints involving injection rates are (31), which are crucial to consider during the data set generation. If we enforce that the generated data set will never fall below the minimum flow rate ($\frac{1}{v_p} \leq \frac{1}{v^{min}_p}$), to guarantee turbulent flow and low volumes for the interfaces between consecutive batches, then, when seeking the robust solution, one does not need to account for the inequality on the right-hand side (RHS). Two reasons support this observation. The first is that since the objective is to minimize the makespan, the optimization will naturally drive the system towards higher pumping rates, as it will be apparent later, when solving the deterministic examples. So, it is reasonable to assume that the pumping rate will be close to its mean value, which is the maximum of the flow rate interval (achieved when all pumps are working at full capacity). Thus, while the flow rate can fluctuate between the maximum and minimum allowable levels, our primary concern is the uncertainty surrounding the maximum flow rate.

The second reason is that the data-driven robust solution derived for the left-hand side inequality remains feasible for the RHS ($L_k = \sum_{i \in I,\ p \in P} \frac{VPR_{i,p,k}}{v_p} \leq \sum_{i \in I,\ p \in P} \frac{VPR_{i,p,k}}{v^{min}_p}$). Nevertheless, the parameters in (31) are all in the form of $\frac{1}{v_p}$, which can pose computational challenges. To mitigate this, we opt to normalize (31) by multiplying both sides of it by $v_1 v_2 \ldots v_{|p|}$, leading to:

$$\begin{aligned} &-v_1 v_2 \ldots v_{|P|} L_k + \sum_{i \in I} v_2 v_3 \ldots v_{|P|}\, VPR_{i,p_1,k} + \sum_{i \in I} v_1 v_3 \ldots v_{|P|}\, VPR_{i,p_2,k} + \ldots \\ &\quad + \sum_{i \in I} v_2 v_3 \ldots v_{|P|-1}\, VPR_{i,p_{|P|},k} \\ &\quad \leq 0 \end{aligned} \tag{32}$$

For example, if we assume four products and two batches to be injected into the pipeline, the reformulated constraint introduces nine uncertain parameters. To generate the synthetic data, we employ two multivariate Normal distributions to generate a data set that is representative of the problem at hand. This is because the pumping rates present not only correlation but also dependency on external factors, e.g., product viscosity.

Table 1 presents the mean values used for the uncertain parameters of the reformulated constraint. Out of the 2000 data points generated, represented in Fig. 2, we randomly selected 300 as our historical dataset to form the uncertainty set, reserving the remaining points for testing purposes. This set is then used to create the robust counterpart, and then the robust solutions are tested against the remaining 1700 data points, to determine the number of constraint violations. The vector $[v_1v_2v_3v_4,\ v_2v_3v_4, v_1v_3v_4,\ v_1v_2v_4,\ v_1v_2v_3,\ v_2v_3v_4, v_1v_3v_4,\ v_1v_2v_4,\ v_1v_2v_3]$ indicates the order in which the values of the nine parameters appear in each data point. To compute the injection rates for each product, it suffices to divide the first component by the others. Note, however, that in order to scale our model, instead of using $v_1v_2v_3v_4 = 1200^4 = 20736{\cdot}10^8$ and $v_1v_2\ldots\widehat{v_i}\ldots v_4 = 1200^3 = 17.28{\cdot}10^8$, we divided them by 10^8 and used the smaller values.

Table 1
Mean values of nine uncertain flowrates (data for the examples in Section 6).

First use of multivariate Normal distribution		Second use of multivariate Normal distribution	
$v_1v_2v_3v_4$	20,732	$v_1v_2v_3v_4$	20,737
$v_2v_3v_4$	16	$v_2v_3v_4$	18
$v_1v_3v_4$	16	$v_1v_3v_4$	18
$v_1v_2v_4$	16	$v_1v_2v_4$	18
$v_1v_2v_3$	16	$v_1v_2v_3$	18

4.1. The Γ-robustness approach

Bertsimas and Sim (2004) proposed a robust optimization approach aimed at mitigating the over-conservatism prevalent in earlier methods. Their work addresses the strict conservatism of the primary approach by Soyster (1973) by strategically selecting a subset of uncertain parameters to take their worst-case values. In essence, if the optimization problem is formulated as:

$$\max\ c^t x$$

$$s.t.\ Ax \leq b,\ x \geq 0,$$

and if J_i represents the index of all uncertain parameters for the i^{th} constraint, and Γ_i is a real number within the interval $[0, |J_i|]$, the following optimization problem provides its robust counterpart:

$$\max\ c^t x$$

$$s.t.\ \sum_j a_{ij} x_j + \max_{\{S_i \cup \{t_i\} | S_i \subseteq J_i,\ |S_i| = \lfloor \Gamma_i \rfloor,\ t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \widehat{a}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor) \widehat{a}_{it_i} y_t \right\} \leq b_i,\ \forall i$$

$$-y_j \leq x_j \leq y_j,\ \forall j$$

$$l \leq x \leq u,\ y \geq 0,$$

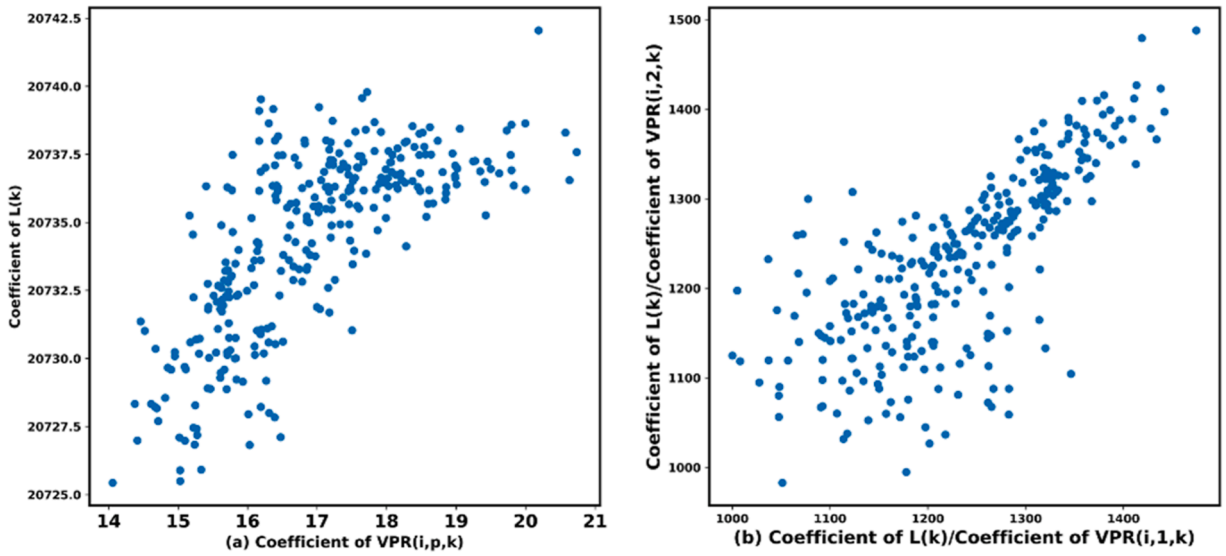

Fig. 2. First two components of the 300 data points (a) and $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ injection rates (b).

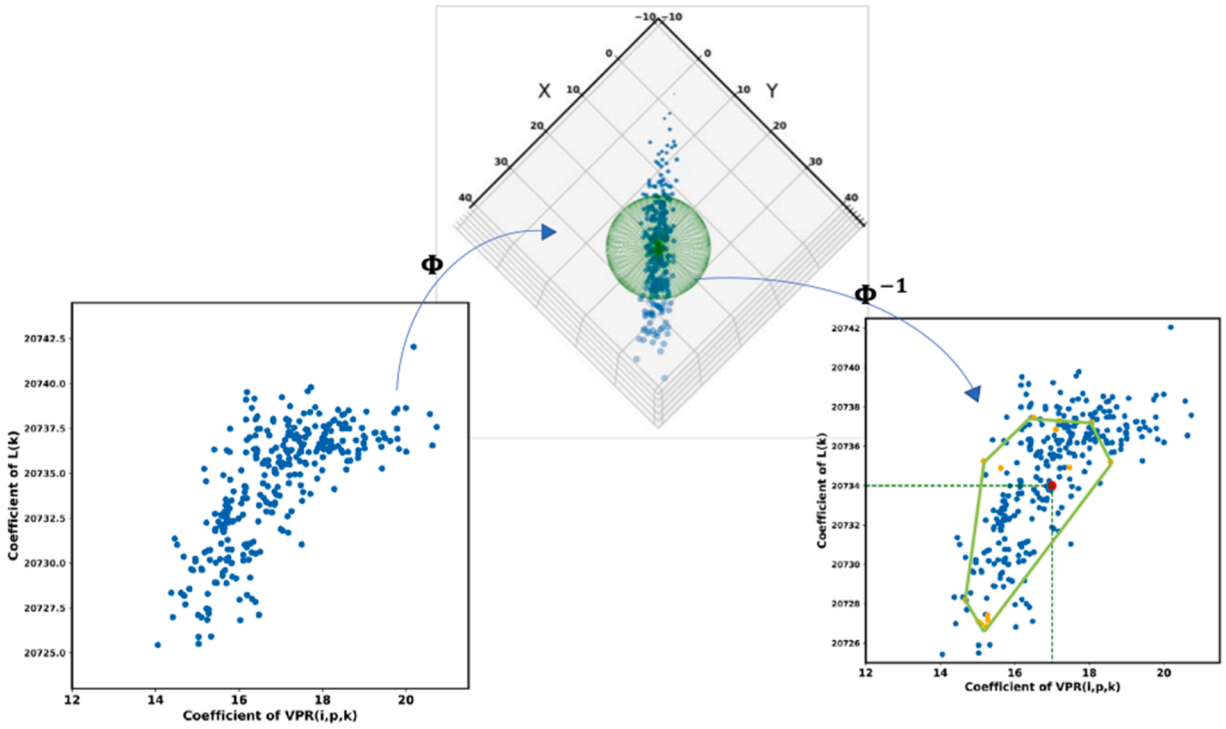

Fig. 3. Process of finding the data-driven uncertainty set with support vector clustering. The orange points represent the projection of the support boundary vectors — the points located on the boundary of the uncertainty set — from the original 9-dimensional space to the Cartesian plane. The convex hull of the outer orange points then creates the projection of the uncertainty set into this two-dimensional space.

where $\widehat{a}_{ij}$ represents the extent to which uncertain parameter j in constraint i can change and a_{ij} is its mean value. Using linear optimization duality, the above formulation can be transformed into the following linear mathematical optimization problem:

$$\max\ c^t x$$

$$s.t.\ \sum_j a_{ij}x_j + z_i\Gamma_i + \sum_{j\in J_i} p_{ij} \le b_i$$

$$z_i + p_{ij} \ge \widehat{a}_{ij}y_j,\ \forall i, j \in J_i$$

$$-y_j \le x_j \le y_j,\ \forall j$$

$$l_j \le x_j \le u_j\ \forall j$$

$$p_{ij} \ge 0,\ \forall i,\ \forall j \in J_i$$

$$y_j \ge 0,\ \forall j$$

$$z_i \ge 0,\ \forall i.$$

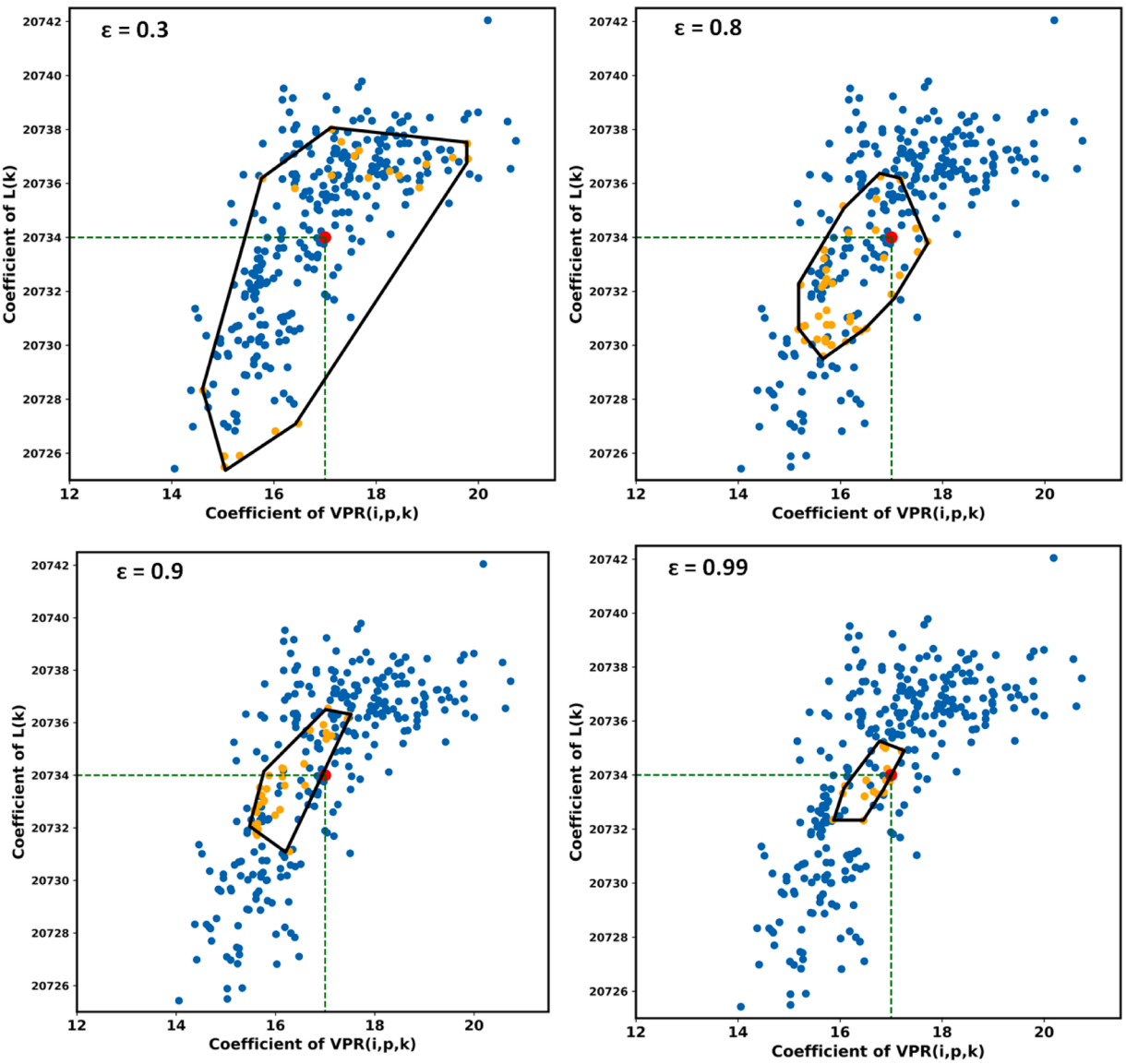

Fig. 4. In the relaxed problem of SVC, conservativeness decreases by increasing ε. Note that as $\varepsilon \to 1$, the uncertainty set concentrates around the mean (17.20734).

4.2. Support vector clustering (SVC)

Support Vector Clustering (SVC), introduced by Ben-Hur et al. (2001), is a machine learning approach designed to identify patterns and cluster data points into groups with shared similarities and is known for excelling in solving a wide range of complex clustering or outlier detection problems. In the SVC algorithm, input data is transformed into a high-dimensional feature space using an unknown function ϕ, with the goal of locating the smallest sphere that encompasses the mapped data (for a visualization, refer to Fig. 3).

Let $\mathscr{D} = \{u^{(i)}\}_{i=1}^{N}$ be our dataset derived from historical data. SVC explores the smallest sphere of radius R that covers all mapped data. This is conceptualized as the following optimization problem:

$$\min_{R,P} R^2$$

$$s.t. \quad \| \phi(u^{(i)}) - P\|^2 \leq R^2, \ \forall i,$$

where $\phi : \mathbb{R}^r \to \mathbb{R}^t$ is a non-linear unknown transformation, mapping the dataset into a higher-dimensional feature space (with r being the number of uncertain parameters and t being unknown), P is the center of the sphere to be determined, and $\| \quad \|$ is the Euclidean norm. To counter the fact that the model incorporates a hard margin, and to prevent outliers from being enclosed by the sphere, we introduce the following relaxed problem:

$$\min_{P,R,\xi} R^2 + \frac{1}{N\varepsilon}\sum_{i}^{N} \xi_i,$$

$$s.t. \quad \| \phi(u^{(i)}) - P\|^2 \leq R^2 + \xi_i \ \forall i = 1, \ldots, N$$

$$\xi_i \geq 0, \ \forall i = 1, \ldots, N,$$

which includes additional slack variables ξ_i, $i = 1,\ldots,N$. Note that $\varepsilon < 1$ is a positive hyperparameter that aids in balancing robustness and optimality by regulating the impact of slack variables, thus controlling the inclusion of outliers (refer to Fig. 4).

Solving the original problem is impractical due to the unknown function ϕ. To solve it, we proceed by formulating its Lagrangian dual and applying the kernel trick to remove the mapping ϕ. The Lagrangian function is stated as follows:

$$L(P,R,\xi,\alpha,\beta) = R^2 + \frac{1}{N\varepsilon}\sum_i^N \xi_i - \sum_i^N \alpha_i\left(R^2 + \xi_i - \| \phi(u^i) - P\|^2\right) - \sum_i^N \beta_i\xi_i, \tag{33}$$

where $\alpha_i, \beta_i,\ i = 1,\ldots,N$ are the Lagrange multipliers. By using the first-order optimality conditions, we obtain:

$$\sum_i^N \alpha_i = 1,\ P = \sum_i^N \alpha_i\phi(u^{(i)}),\ \alpha_i + \beta_i = \frac{1}{N\varepsilon}. \tag{34}$$

To obtain a tractable formulation to the Lagrange function, we combine Eqs. (34) into (33) and obtain

$$L(P,R,\xi,\alpha,\beta) = \sum_i \alpha_i\ \| \Phi(u^i) - \sum_j \alpha_j\Phi(u^j)\Big|^2,$$

with the conditions of $\sum_i \alpha_i = 1$ and $0 \le \alpha_i \le \frac{1}{N\varepsilon}$ for all i. Expanding the norm $\| \Phi(u^i) - \sum_j \alpha_j\Phi(u^j)\|^2$ gives:

$$\sum_i \alpha_i\ \| \Phi(u^i) - \sum_j \alpha_j\Phi(u^j)\Big|^2 = \sum_i \alpha_i\langle\Phi(u^i),\Phi(u^i)\rangle - \sum_i\sum_j \alpha_i\alpha_j\langle\Phi(u^i),\Phi(u^j)\rangle.$$

Now, using the kernel trick, one can replace the inner products by the kernel K (K will be explicitly introduced later) using the equality $\langle\Phi(u^i), \Phi(u^j)\rangle = K(u^i,u^j)$, which leads to:

$$L(P,R,\xi,\alpha,\beta) = \sum_{i=1}^N \alpha_i K(u^{(i)},u^{(i)}) - \sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j K(u^{(i)},u^{(j)}).$$

We then solve the following dual problem and then use the complementary slackness to find the radius R (all the computations leading to radius R are given in the next paragraphs).

$$\max_{\alpha} \sum_{i=1}^N \alpha_i K(u^{(i)},u^{(i)}) - \sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j K(u^{(i)},u^{(j)})$$

$$s.t.\ 0 \le \alpha_i \le \frac{1}{N\varepsilon},\ i = 1,\ldots,N,$$

$$\sum_{i=1}^N \alpha_i = 1.$$

Suppose the above problem is solved to optimality with a given kernel function K. The complementary slackness equations:

$$\alpha_i\left(R^2 + \xi_i - \| \phi(u^i) - P\|^2\right) = 0,\ \forall i,$$

$$\beta_i\xi_i = 0,\ \forall i,$$

lead to the following results (notice that in the primal problem the constraints are $\| \phi(u^{(i)}) - P\|^2 \le R^2 + \xi_i$ and $\xi_i \ge 0$ for all i and moreover, we have $\alpha_i + \beta_i = \frac{1}{N\varepsilon}$ for dual variables):

1) $\alpha_i = 0, \beta_i = \frac{1}{N\varepsilon}$ if and only if $\| \phi(u^{(i)}) - P\| < R$ (note that $\beta_i > 0$ and so $\xi_i = 0$);
2) $0 < \alpha_i < \frac{1}{N\varepsilon}$ *and* $0 < \beta < \frac{1}{N\varepsilon}$ if and only if $\| \phi(u^{(i)}) - P\| = R$;
3) $\alpha_i = \frac{1}{N\varepsilon}$ *and* $\beta_i = 0$ if and only if $\| \phi(u^{(i)}) - P\| > R$.

The second item shows that after solving the dual problem, one can choose an arbitrary index i for which $0 < \alpha_i < \frac{1}{N\varepsilon}$ and calculate the radius as follows:

$$R^2 = \| \phi(u^{(i)}) - P\|^2 = K(u^{(i)},u^{(i)}) - \sum_j \alpha_j K(u^{(i)},u^{(j)}), \tag{35}$$

Now, everything is ready to introduce the uncertainty set but before going ahead, let us define three different types of data points in $\mathscr{D}$ based on the conditions for α_i and β_i listed above.

The first group, with $\alpha_i = 0$ and $\beta_i = \frac{1}{N\varepsilon}$, resides inside the sphere ($\| \phi(u^{(i)}) - P\| < R$); the second lies exactly on the sphere ($\| \phi(u^{(i)}) - P\| = R$) with $0 < \alpha_i < \frac{1}{N\varepsilon}$; and the third is situated outside of the sphere ($\| \phi(u^{(i)}) - P\| > R$), for which α_i attains its maximum value. The points with non-zero α_i are referred to as *support vectors*. Among them, those with $0 < \alpha_i < \frac{1}{N\varepsilon}$ are referred to as *boundary support vectors*, residing exactly on the sphere, while those with $\alpha_i = \frac{1}{N\varepsilon}$ are labeled as *outliers* and lie outside of the sphere. Therefore, we can introduce the indexing set for the set of support vectors (B) and boundary support vectors (BSV) as follows:

$$B = \{1 \le i \le N\ |\alpha_i > 0\},$$

$$BSV = \left\{1 \le i \le N \middle| 0 < \alpha_i < \frac{1}{N\varepsilon}\right\}.$$

Excluding the outliers, we define the set of all data inside and on the sphere as the uncertainty set ($U(\mathfrak{H})$). In other words, it includes all points u in the domain of ϕ, such that the distance between $\phi(u)$ and the center is at most R (recall Fig. 3), i.e.,

$$U(\mathfrak{H}) = \left\{u\middle|\ \|\Phi(u) - P\|^2 \le R^2\right\},$$

where it simplifies to the following set:

$$U(\mathfrak{H}) = \left\{u\middle|\ K(u,u) - 2\sum_{i=1}^N \alpha_i K(u,u^{(i)}) + \sum_i^h\sum_l^f \alpha_i\alpha_j K(u^{(i)},u^{(j)}) \le R^2\right\}. \tag{36}$$

The set in (36) represents the general form of a data-driven uncertainty set using the kernel function K. However, it is crucial to select a kernel function that simplifies the definition of set $U(\mathfrak{H})$ with linear bounds. Conventional kernel functions, such as polynomial, sigmoid, and RBF, introduce complexity to the uncertainty set by adding nonlinearity, rendering the robust counterpart computationally intractable. Moreover, these functions often overlook correlations among uncertain parameters. To address these challenges, we opt for the Weighted Generalized Intersection Kernel (WGIK) introduced by Shang et al. (2017). Denoted by K, this function is defined as:

$$K(d,d') = \sum_k^N o_k - |Q(d-d')|_1, \tag{37}$$

where o_k for $1 \le k \le N$ is the kernel parameter satisfying $o_k > \max_{1\le i\le N} Q_k^T u^{(i)} - \min_{1\le i\le N} Q_k^T u^{(i)}$, Q_k is the k^{th} row of the weighting matrix computed via the estimated covariance matrix and $|\ |_1$ is the L_1. WGIK is best tailored for construction of uncertainty sets, especially when the

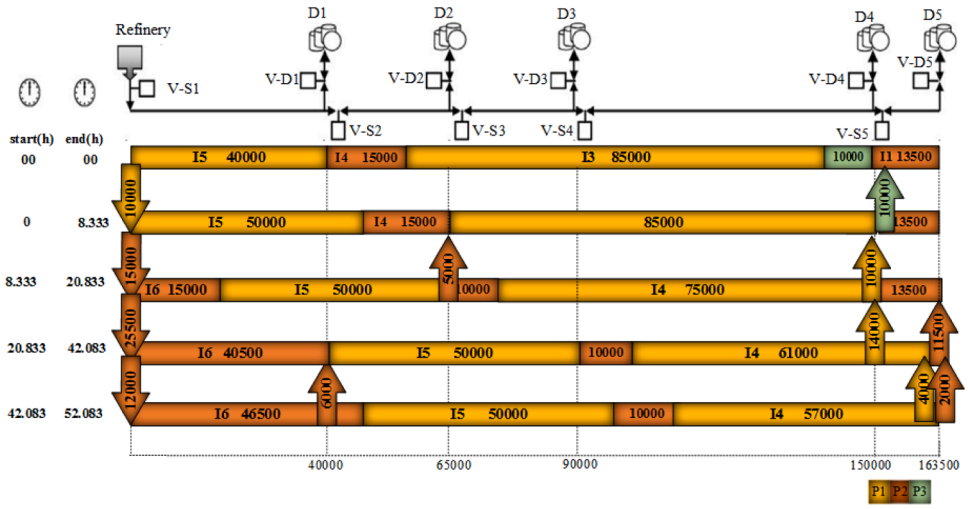

Fig. 5. Optimal deterministic schedule for Example 1.

original problem is a linear programming problem since: (1) it is concave and the resulting uncertainty set is a convex polyhedral set; (2) it incorporates the correlation between uncertain parameters through weighting matrix Q in the definition of the kernel function; and (3) its definition is in a way that each parameter has the same impact on the kernel expression.

By inserting (35) and substituting the formula of K in (36), we can explicitly write the SVC-based uncertainty set $U_\varepsilon(\mathfrak{H})$ (for $\varepsilon < 1$) as:

$$U_\varepsilon(\mathfrak{H}) = \left\{ u \,|\, \sum_{b\in B} \alpha_b \left|Q\left(u-u^{(b)}\right)\right|_1 \le \sum_{b\in B} \alpha_b \left|Q\left(u^{(i')}-u^{(b)}\right)\right|_1, for some i' \in BSV \right\}. \tag{38}$$

We let $\theta = \sum_{b\in B} \alpha_b \left|Q\left(u^{(i')} - u^{(b)}\right)\right|_1$, which is a constant. Introducing auxiliary variables a_i for $b \in B$ satisfying:

$$-a_b \le Q\left(u-u^{(b)}\right) \le a_b \text{ and } \left(\sum_{b\in B} \alpha_b \left(1^T \cdot a_b\right)\right) \le \theta$$

leads to a linear expression for our uncertainty set. Here, the auxiliary variables a_b ($b \in B$) form a vector whose length is the number of uncertain parameters, 1^T is a vector of ones, and $1^T \cdot a_b$ is the inner product of the vectors. Thus, the uncertainty set can be expressed as:

$$U_\varepsilon(\mathfrak{H}) = \left\{ u \,\middle|\, \sum_{b\in B} \alpha_b \left(1^T \cdot a_b\right) \le \theta \text{ and } -a_b \le Q\left(u-u^{(b)}\right) \le a_b \; \forall b \in B \right\} \tag{39}$$

4.3. Robust counterparts for pipeline scheduling

If the uncertainty set has been formulated as $U_\varepsilon(\mathfrak{H})$, then, according to Shang et al. (2017), one can express the robust counterpart of the uncertain constraint in the form of the following constraints:

$$\begin{cases} \sum_{b\in SV(B)} \sum_{a\in A} (\mu_{abk} - \lambda_{abk}) Q\left(u^{(b)}\right)_a + \eta_k \theta \le 0, \; \forall\, k \in K, \\ \sum_{b\in B} \sum_{a'\in A} Q_{aa'} (\lambda_{rbk} - \mu_{a'bk}) + VvC_{ka} = 0, \; \forall a \in A, \; k \in K, \\ \lambda_{abk} + \mu_{abk} = \eta_k \alpha_b, \; \forall\, b \in B, \; k \in K, \; a \in A, \\ \lambda_{abk}, \; \mu_{abk}, \eta_k \ge 0, \; \forall\, b \in B, \; k \in K, \; a \in A, \\ VvC_k \gtrless 0 \; \forall\, k \in K, \end{cases} \tag{40}$$

where we set $VvC_k = \left(L_k, \; VPR_{1,1,k}, VPR_{2,1,k}, \ldots, VPR_{1,2,k}, VPR_{2,2,k}, \ldots, VPR_{|I|,|p|,k}\right)$ as the vector of variables from (32) for a fixed $k \in K$. Moreover, the set $A = \{1, \ldots, |I \,\|\, P| + 1\}$ is an indexing set associated with VvC_k (note that $|I \,\|\, P| + 1$ is the length of VvC_k). Moreover, Q is the weighting matrix, B is the set of indices for support vectors, α_b for all $b \in B$ comes from the solution of the dual problem and θ is defined as before.

The Γ-robust approach counterpart for (32) is formed as follows:

$$\begin{cases} G \cdot L_k + \sum_{i,p} \left(M_p \; VPR_{i,p,k} + CV1_{i,p,k}\right) + \Gamma \cdot CV2_k + CV3_k \le 0 \; \forall k \in K, \\ CV2_k + CV1_{i,p,k} \ge \mathrm{DI1}_p \times CV4_{i,p} \; \forall k \in K, \; i \in I, \; p \in P, \\ CV2_k + CV3_k \ge DI2 \; \times CV5 \; \forall k \in K, \\ -CV4_{i,p} \le VPR_{i,p,k} \le CV4_{i,p} \; \forall k \in K, \; i \in I, \; p \in P, \\ -CV5 \le L_k \le CV5 \; \forall k \in K, \end{cases} \tag{41}$$

where M_p is the mean value for the $v_1 v_2 \ldots v_{p-1} v_{p+1} \ldots v_{|P|}$ component of the data set, G is the mean value for the $-v_1 v_2 \ldots v_{|P|}$ component of the data set, Γ is a real number between 0 and the number of uncertain parameters, $CV1_{i,p,k}$, $CV2_k, CV3_k$, $CV4_{i,p}$, $CV5$ are the variables introduced when the robust counterpart is formed, $DI1_p$ is the difference between the minimum and maximum values of $v_1 v_2 \ldots \hat{v}_p \ldots v_{|P|}$ and $DI2$ is the difference between the minimum and maximum of values of $-v_1 v_2 \ldots v_{|P|}$ in the data set. Note that all the variables are nonnegative.

In summary, the robust counterparts include Eqs. (1) to (30) of the deterministic MILP model and either Eq. (40), for the data-driven approach, or Eq. (41), for the Γ-robust approach.

Table 2
Problem data related to pipeline segments.

Segment	Volume (m^3)	Flowrate range (m^3/h)
Refinery-D1	40,000	800–1200
D1-D2	25,000	600–1200
D2-D3	25,000	600–1200
D3-D4	60,000	600–1200
D4-D5	13,500	400–800

5. Computational results for deterministic scheduling

In this section, we aim to compare the proposed deterministic model with the models developed by Ghaffari-Hadigheh and Mostafaei (2015), denoted as GM, and Liao et al. (2019a), denoted as LCLZ, by solving two real life examples. The MILP mathematical formulation is implemented in GAMS 43 and the resulting problems are solved to optimality by solver CPLEX 22.1, using default options. The hardware consisted of an

Table 3
Problem data related to inventory levels at the refinery and product demand at the depots.

		Inventory (m^3)	Demand (m^3)				
Example	Product/Depot	Refinery	D1	D2	D3	D4	D5
Example 1	P1	30,000				24,000	4000
	P2	72,000	6000	5000			13,500
	P3	10,000				10,000	
Example 2	P1			9000	6000	49,000	26,000
	P2	135,600			13,600	15,200	38,300
	P3					13,000	7000
	P4	42,500				1000	

Table 4
Computational statistics for deterministic problems.

Example	Model	Makespan	Binary variables	Continuous variables	Equations	CPUs
Example 1	GM	52.083	193	1083	1702	0.046
	LCLZ		454	1684	1790	0.031
	This work		141	874	1352	0.032
Example 2	GM	150.083	456	2665	3968	8.51
	LCLZ	148.417	548	1731	2194	5.77
	This work		225	1434	2172	2.37

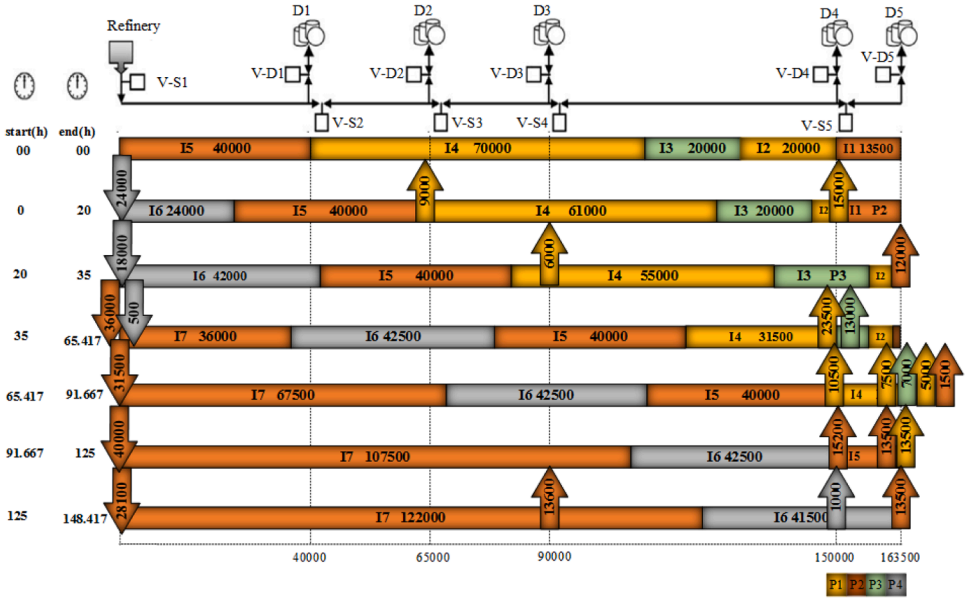

Fig. 6. Optimal schedule generated by our model for Example 2.

Intel Core i7 @ 2.70 GHz Processor, 16 GB of RAM running Windows 11 Pro.

5.1. Example 1

Example 1 corresponds to the first example in Ghaffari-Hadigheh and Mostafaei (2015) and involves a straight pipeline network with one input node and five output nodes (D1-D5), as depicted in the first line of Fig. 5. The refinery injects three products—gasoline (P1), diesel (P2), LPG (P3)—at a flow rate within the interval [800, 1200] m^3/h. Note that as given in Table 2, the flowrate lower and/or upper bounds decrease when going through the segments. The objective is to fulfill product demand as fast as possible (see details in Table 3).

An important feature of the proposed continuous-time model is that it allows multiple batch injections at the refinery as well as multiple delivery operations at the distribution centers, during a single time slot. An example of the latter can be seen in Fig. 5: in the last pumping run, occurring from time 42.083 to 52.083 h (last line), depot D5 receives material from two batches (I_1 and I_3; the first and third batches initially in the pipeline). Note that simultaneous deliveries to different depots are also allowed (e.g., to D2 and D4 during [8.333, 20,833] h).

Table 4 reports the statistics of the three models, with ours generating a significantly smaller problem size, particularly with respect to the LCLZ model, for about the same computational time. Still, example 1 can be solved almost instantaneously and so a more complex example is required to validate the proposed formulation.

5.2. Example 2

Example 2 corresponds to the second example of Ghaffari-Hadigheh and Mostafaei (2015), and is a variant of Example 1, since it features the same pipeline system but with different initial conditions (please refer to the first row of Fig. 6), and larger values for product demand at the depots and inventory at the refinery (Table 3). Note that one more product is being injected: jet-fuel (P4).

The results in Table 4 show that the smaller problem size is now

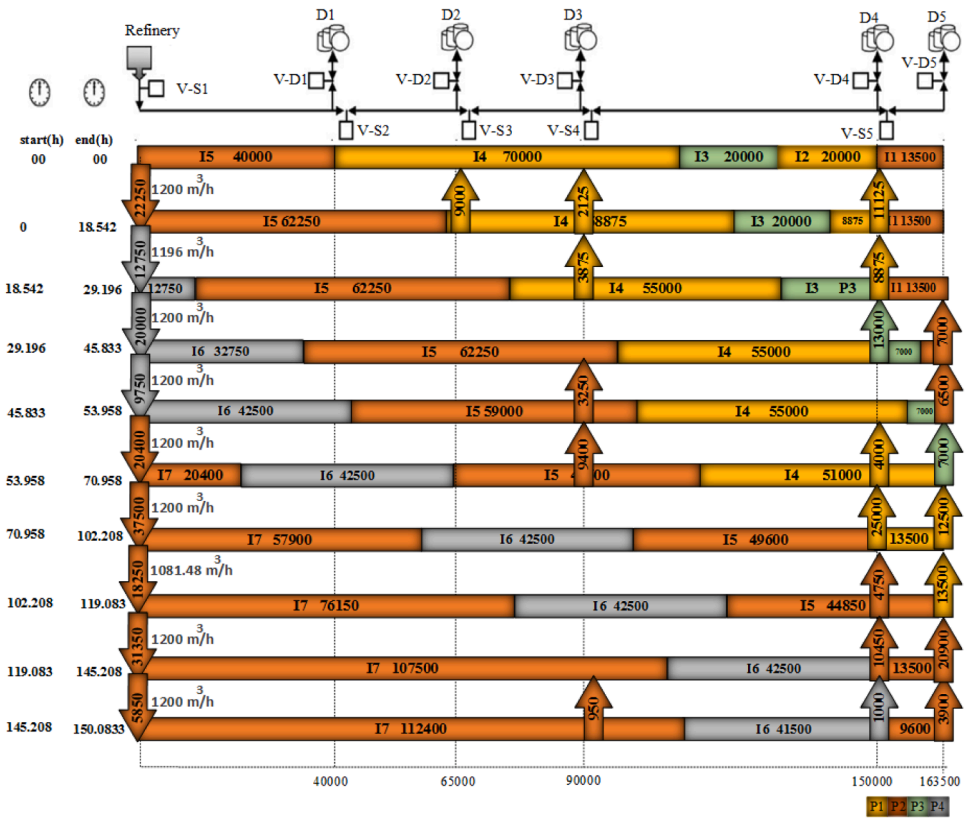

Fig. 7. Optimal schedule reported by GM model for Example 2, which is a suboptimal solution.

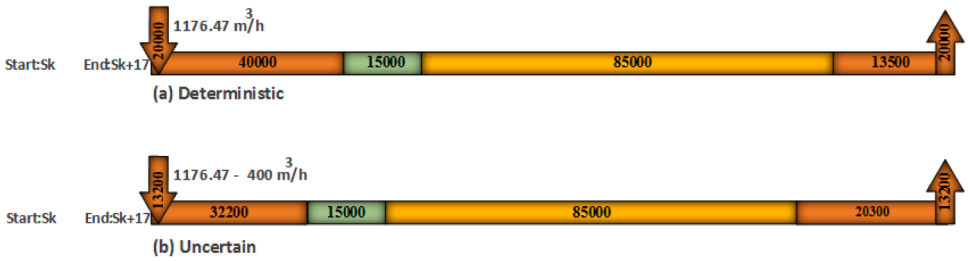

Fig. 8. Comparison between deterministic (a) and robust (b) schedules for a pumping run (k) lasting 17 h Due to the failure of one pumping facility, only 13,200 of the 20,000 m3 injected in the deterministic schedule can be processed.

reflected in a much faster computational time, about 41 % of LCLZ and 28 % of GM. Furthermore, due to the novel feature of multiple product injections/deliveries compared to the GM model, the makespan has been reduced from 150.083 to 148.417, which is the optimal value reported by Liao et al. (2019a). The corresponding schedule is given in Fig. 6. Notice that the injection rate is 1200 m^3/h for all the runs, whereas in Fig. 7, showing the best solution reported in Ghaffari-Hadigheh and Mostafaei (2015), there are two injections below such maximum rate, most notably the third from last, at 1081.48 m^3/h.

6. Comparison between the kernel-based and Γ-robust approaches

The dataset in Table 1 is used in this section to compare Γ-robust and data-driven approaches. The aim of the robust optimization approach is to ensure that the obtained optimal solution remains feasible under any parameter realization that lies within the uncertainty set. To illustrate this concept in pipeline scheduling, let us consider a scenario with three pumping facilities at the input station, each featuring a pumping rate of 400 m^3/h. Two possibilities may occur:

(1) All pumping facilities are operational (Fig. 8a). The optimal injection rate is equal to 1176.5 m^3/h (20,000 m^3 pumped over 17 h) below the maximum of 1200 m^3/h, and the inequality on the LHS of Eq. (31) holds: $\frac{20,000}{1200} \leq \frac{20,000}{1176.5} = 17$.

(2) Suddenly, one of the pumping facilities becomes unavailable, resulting in a drop of 400 m^3/h in the injection rate. If the operator continues injection operations with the information obtained from deterministic optimization, the operator faces the inequality $\frac{20,000}{1176.5-400} > 17$, indicating that Eq. (31) no longer holds, resulting in infeasibility. Therefore, the flow rate of 776.5 m^3/h must be considered as one of the possible realizations (Fig. 8b).

While robust optimization protects the optimal schedule from uncertainty, there is the drawback of overconservatism. The Γ-robust approach constructs an uncertainty set that is symmetric around the mean value and therefore covers areas far away from the historical data points (see Fig. 9 on the left). This implies that the robust solution loses quality (in comparison with the deterministic optimal solution) to gain robustness. On the other hand, the data-driven approach efficiently defines an uncertainty set covering the historical data points, and

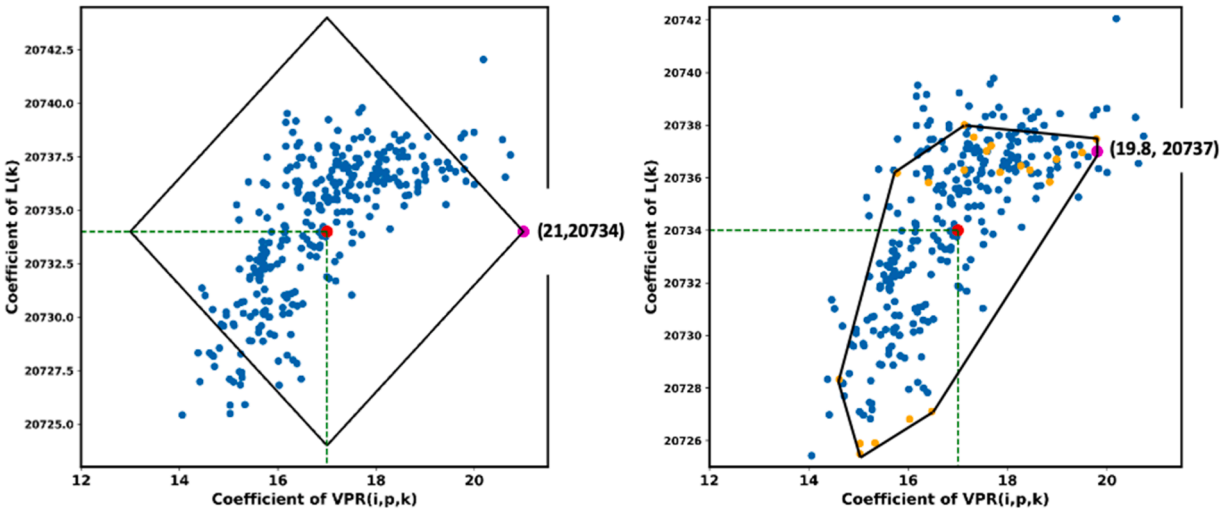

Fig. 9. Comparison between Γ-robust (on the left) and data-driven approach (on the right). For about the same level of data coverage, the lowest injection rate on the left (20,734/21=987 m3/h) is distant from the data points, whereas the lowest injection rate for the SVC approach is one of the data points (20,737/19.8 = 1047 m3/h).

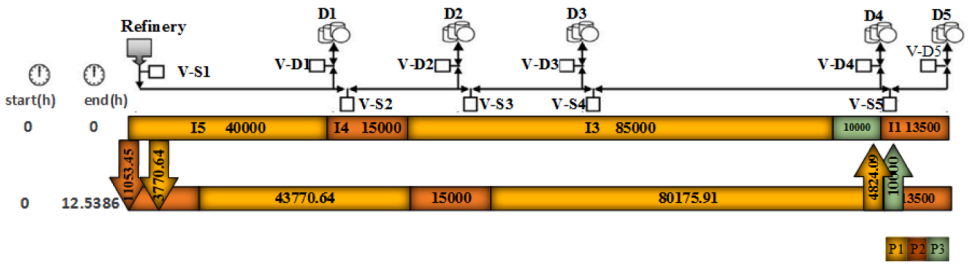

Fig. 10. Robust solution for Example 1 from data-driven approach when using $\varepsilon = 0.9$ (first pumping run).

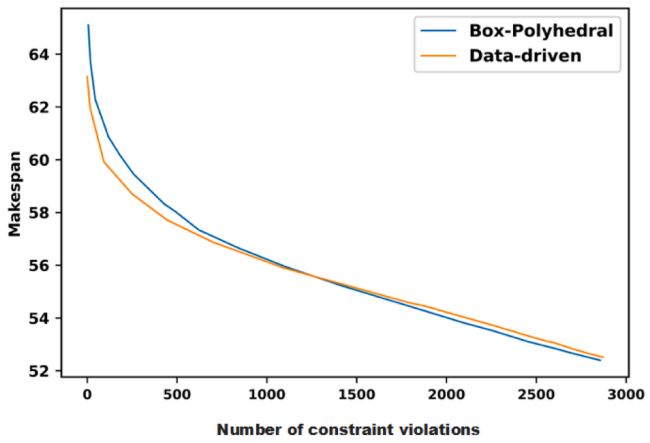

Fig. 11. Comparison of the conservativeness level of robust optimization approaches for Example 1. To determine the number of constraint violations of Eq. (31) for each robust solution (characterized by a certain value of makespan), we have used a test set with 1700 data points, where counting is done for each of the pumping runs.

therefore, the worst-case scenario is one of the data points itself (see Fig. 9 on the right).

To compare the two robust optimization approaches for scheduling under uncertainty, we now revisit the first example of Section 5.

6.1. Example 1

Example 1 features 4 products and 1 new batch, and part of the robust solution for the data-driven approach is shown in Fig. 10. During the first pumping run, L_1= 12.538 h, $VPR_{5,1,1}$= 3770.64 and $VPR_{6,2,1}$= 11,053.45 m^3 of products P1 and P2 enter the pipeline as batches I5 and

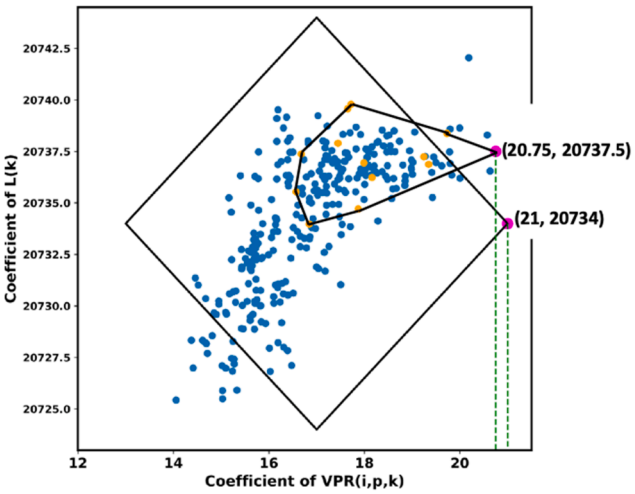

Fig. 12. Uncertainty sets obtained by Γ-robust (diamond shape) and data-driven (other polygon) approaches for example 1, featuring the same level of robustness. The worst cases correspond to injection rates of 20,734/21=987 m^3/h and 20,737.5/20.75= 999 m^3/h, respectively.

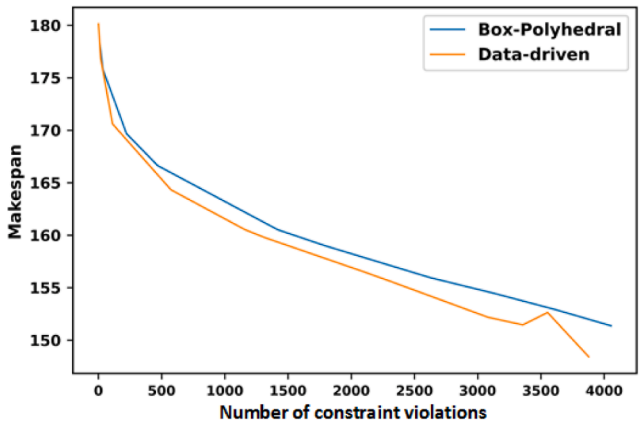

Fig. 13. Comparison of the conservativeness level of robust optimization approaches for Example 3.

I6, respectively. No other products enter the pipeline, and so $VPR_{5,p,1} = 0\ \forall p \neq$ P1 and $VPR_{6,p,1} = 0\ \forall p \neq$ P2. These values can be replaced in Eq. (32), leading to (for $k = 1$):

$-v_1v_2v_3v_4L_1 + v_2v_3v_4VPR_{5,1,1} + v_1v_3v_4VPR_{5,2,1} + v_1v_2v_4VPR_{5,3,1} + v_1v_2v_3VPR_{5,4,1} + v_2v_3v_4VPR_{6,1,1} + v_1v_3v_4VPR_{6,2,1} + v_1v_2v_4VPR_{6,3,1} + v_1v_2v_3VPR_{6,4,1} \leq 0$, which has 9 uncertain parameters (the terms multiplying the VPR variables). Note that the number of non-zero variables may decrease in subsequent pumping runs, leading to fewer uncertain parameters. Specifically, if batch I5 is not injected in later runs, there will be just 5 uncertain parameters. This is the case of run $k = 6$, which only injects batch I6 into the pipeline.

After replacing the set of constraints (for all k) by their robust counterparts, robust solutions can be generated for both approaches. Then, different robust solutions can be generated by varying Γ from 0 to 9 (Γ-robustness approach), and ε between 0 and 1 (SVC approach).

If we replace the flowrate values $v_1, \ldots, v_4$ of the first test data point, we can check if the above constraint is violated (i.e., LHS > 0). We can then do the same for all other test data points to get to the total number of constraint violations for the first run, which is equal to 612. The total number over all runs is 2312, which is greater than the number of data points in the test set (1700). The reason for so many violations is because optimality dominates robustness for values of ε close to 1 (that is, the uncertainty set has been reduced considerably).

Fig. 11 compares the two approaches by representing makespan as a function of the total number of constraint violations, with the best performer being the one fulfilling product demand in the shortest time. In the region between 0 and 500 violations (high level of robustness), the orange curve is below the blue curve, indicating the better performance of the proposed data-driven approach. Then, as the level of robustness is decreased to about 2900 violations (low level of robustness, approaching the deterministic solution), a similar performance is observed.

The question may be raised why this graph is suitable for illustrating the differences. Suppose that U_1 and U_2 are two uncertainty sets obtained by data driven and Γ-robust approaches, respectively, leading to the same number of constraint violations for their robust solutions. This is the case of Fig. 12, where the worst cases correspond to injection flowrates of 999 and 987 m^3/h, respectively. The higher flowrate allows product demand to be fulfilled faster, leading to a lower makespan. Thus, U_1 is a more efficient uncertainty set.

6.2. Example 3

The number of extra variables added to the robust counterpart of the data-driven approach is obtained by multiplying the number of support vectors and the number of pumping runs. Therefore, the more complex the model (e.g., a longer time horizon with several depots demanding multiple products), the more difficult it is to solve it. Preliminary results showed no solution being found after a couple of days of computational time, and so, to validate the data-driven approach, we consider a simpler instance than Example 2. In Example 3, the time horizon is 160 h, the initial condition of the pipeline is in Fig. 6, there is unlimited inventory of P1 in the refinery and only the first depot has product demand (1.781 $\times$ 10^5 m^3 of P1).

The number of pumping runs needed is 6 and only one new batch is pumped. Fig. 13 compares both approaches regarding their conservatism through this example. This figure shows that for larger examples (larger demand), the data-driven approach performs better, particularly in the less conservative region of the graph.

6.3. Computational statistics

Reflecting on the advantages and disadvantages, one must mention that the SVC data-driven approach generates larger mathematical problems and has steeper computational resource requirements, as seen in Table 5. Note that both approaches share the number of binary variables with the deterministic formulation.

7. Conclusions

Firstly, a continuous-time MILP model was developed to optimize the transportation of refined oil products from a single refinery to multiple distribution centers using a straight pipeline. The numerical results have shown that our model generates smaller problem sizes and is computationally faster than two others from the literature. This is because it requires fewer pumping runs to represent a schedule since it can handle multiple batch injections and deliveries to the farthest active depot, per run.

Secondly, this study delved into the realm of pipeline scheduling optimization under uncertainty by developing the robust counterpart of the deterministic formulation. Two distinctive approaches were explored and compared: the Γ-robust and a data-driven approach using support vector clustering. Through a couple of examples, it became evident that the data-driven approach offers a more efficient and less conservative solution, capturing better the information of historical data points, and leading to robust solutions that align closely with real-world conditions, particularly in the case with a longer planning horizon. The drawback is that it is more demanding computationally, which highlights the need for more efficient solution methods. Developing such computationally efficient methods is, therefore, a direction that merits

Table 5
Computational statistics for robust optimization approaches.

Example	Data coverage parameter	Makespan (h)	Binary variables	Continuous variables	Equations	CPUs
Example 1	$\Gamma = 0.25$	54	141	923	1457	2
	$\Gamma = 0.3$	56				2
	$\varepsilon = 0.8$	54		19,094	10,492	61
	$\varepsilon = 0.5$	63		12,254	7072	51
Example 3	$\Gamma = 0.25$	153	197	1337	2119	2
	$\Gamma = 0.3$	155				2
	$\varepsilon = 0.8$	155		28,598	15,672	175
	$\varepsilon = 0.5$	180		18,338	10,542	83

further research.

CRediT authorship contribution statement

Amir Baghban: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Pedro M. Castro:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Fabricio Oliveira:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Pedro M. Castro acknowledges the financial support from Fundação para a Ciência e Tecnologia (FCT) through project UIDB/04028/2020. Fabricio Oliveira acknowledges the financial support of the Research Council of Finland (decision number 348094)

Data availability

Data will be made available on request.

References

Asl, N.B., MirHassani, S.A., 2019. Benders decomposition with integer sub-problem applied to pipeline scheduling problem under flow rate uncertainty. Comput. Chem. Eng. 123, 222–235.
Ben-Hur, A., Horn, D., Siegelmann, H.Y., Vapnik, V., 2001. Support vector clustering. JMLR. 2, 125–137.
Bertsimas, D., Sim, M., 2004. The price of robustness. Oper. Res. 52, 35–53.
Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer, Berlin.
Cafaro, D.C., Cerdá, J., 2004. Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. Comput. Chem. Eng. 28 (10), 2053–2068.
Cafaro, D.C., Cerdá, J., 2008a. Efficient tool for the scheduling of multiproduct pipeline and terminal operation. Ind. Eng. Chem. Res. 47, 9941–9956.
Cafaro, D.C., Cerdá, J., 2008b. Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. Comput. Chem. Eng. 32 (4), 728–753.
Cafaro, D.C., Cerdá, J., 2009. Optimal scheduling of refined products pipelines with multiple sources. Ind. Eng. Chem. Res. 48 (14), 6675–6689.
Cafaro, D.C., Cerdá, J., 2010. Operational scheduling of refined products pipeline networks with simultaneous batch injections. Comput. Chem. Eng. 34 (10), 1687–1704.
Cafaro, D.C., Cerdá, J., 2011. A rigorous mathematical formulation for the scheduling of tree-structure pipeline networks. Ind. Eng. Chem. Res. 50 (9), 5064–5085.
Cafaro, D.C., Cerdá, J., 2012. Rigorous scheduling of mesh-structure refined petroleum pipeline networks. Comput. Chem. Eng. 38, 185–203.
Cafaro, D.C., Cerdá, J., 2014. Rigorous formulation for the scheduling of reversible-flow multiproduct pipelines. Comput. Chem. Eng. 61, 59–76.
Cafaro, V.G., Cafaro, D.C., Mendéz, C.A., Cerdá, J., 2015. Optimization model for the detailed scheduling of multi-source pipelines. Comput. Ind. Eng. 88, 395–409.
Castro, P.M., 2017a. Optimal scheduling of multiproduct pipelines in networks with reversible flow. Ind. Eng. Chem. Res. 56, 9638–9656.
Castro, P.M., Mostafaei, H., 2017b. Product-centric continuous-time formulation for pipeline scheduling. Comput. Chem. Eng. 104, 283–295.
Castro, P.M., Mostafaei, H., 2019. Batch-centric scheduling formulation for treelike pipeline systems with forbidden product sequences. Comput. Ind. Eng. 112, 2–18.
Chatterjee, T., Chowdhury, R., 2017. Improved Sparse Approximation Models For Stochastic computations. Handbook of Neural Computation. Elsevier, pp. 201–223.
Chen, H., Zou, L., Wu, C., Wang, L., Diao, F., Chen, J., Huang, Y., 2017. Optimizing detailed schedules of a multiproduct pipeline by a monolithic MILP formulation. J. Petrol. Sci. Eng. 159, 148–163.
Dimas, D., Murata, V.V., Neiro, S.M.S., Relvas, S., Barbosa-Póvoa, A.P., 2018. Multiproduct pipeline scheduling integrating for inbound and outbound inventory management. Comput. Chem. Eng. 115, 377–396.
Ghaffari-Hadigheh, A., Mostafaei, H., 2015. On the scheduling of real-world multiproduct pipelines with simultaneous delivery. Optim. Eng. 16 (3), 571–604.
Herran, A., Cruz, J.M., Andres, B., 2010. Mathematical model for planning transportation of multiple petroleum products in a multi-pipeline system. Comput. Chem. Eng. 34, 401–413.
Kirschstein, T., 2018. Planning of multi-product pipelines by economic lot scheduling models. Eur. J. Oper. Res. 264, 327–339.
Liao, Q., Castro, P.M., Liang, Y., Zhang, H., 2019a. Computationally efficient MILP model for scheduling a branched pipeline system. Ind. Eng. Chem. Res. 58, 5236–5251.
Liao, Q., Castro, P.M., Liang, Y., Zhang, H., 2019b. Batch-centric model for scheduling straight multi-source pipelines. AIChE J. 65, e16712.
Li, Z., Liang, Y., Liao, Q., Zhang, B., Zhang, H., 2021. A review of multiproduct pipeline scheduling: from bibliometric analysis to research framework and future research directions. J. Pipel. Sci. Eng. 1 (4), 395–406.
Magatão, L., Arruda, L.V.R., Neves, F.A., 2004. A mixed integer programming approach for scheduling commodities in a pipeline. Comput. Chem. Eng. 28, 171–185.
Mohseni, S., Pishvaee, M., 2020. Data-driven robust optimization for wastewater sludge-to-biodiesel supply chain design. Comp. Ind. Eng. 139, 105944.
Moradi, S., Mirhassani, S.A., 2016. Robust scheduling for multi-product pipelines under demand uncertainty. J. Adv. Manuf. Technol. 87, 2541–2549.
Moradi, S., MirHassani, S.A., Hooshmand, F., 2019. Efficient decomposition-based algorithm to solve long-term pipeline scheduling problem. Pet. Sci. 16, 1159–1175.
Mostafaei, H., Alipouri, Y., Zadahmad, M., 2015a. A mathematical model for scheduling of real-world tree-structured multi-product pipeline system. Math. Oper. Res. 81, 53–81.
Mostafaei, H., Castro, P.M., Ghaffari-Hadigheh, A., 2015b. A novel monolithic MILP framework for lot-sizing and scheduling of multiproduct treelike pipeline networks. Ind. Eng. Chem. Res. 54, 9202–9221.
Mostafaei, H., Castro, P.M., Ghaffari-Hadigheh, A., 2016. Short-term scheduling of multiple source pipelines with simultaneous injections and deliveries. Comput. Chem. Eng. 73, 27–42.
Mostafaei, H., Castro, P.M., Oliveira, F., Harjunkoski, I., 2021a. Efficient formulation for transportation scheduling of single refinery multi-product pipelines. Eur. J. Oper. Res. 293 (2), 731–747.
Mostafaei, H., Castro, P.M., Relvas, S., Harjunkoski, I., 2021b. A holistic MILP model for scheduling and inventory management of a multiproduct oil distribution system. Omega (Westport) 98, e102108.
Muhlbauer, W., 2004. Pipeline Risk Management manual: Ideas, Techniques and Resources. Elsevier, USA.
Qiu, R., Sun, Y., Fan, Z., Sun, M., 2020. Robust multi-product inventory optimization under support vector clustering-based data-driven demand uncertainty set. Soft Comput. 24, 6259–6275.
Rejowski, R., Pinto, J, 2003. Scheduling of a multiproduct pipeline system. Comput. Chem. Eng. 27, 1229–1246.
Shang, C., Huang, X., You, F., 2017. Data-driven robust optimization based on kernel learning. Comput. Chem. Eng. 106, 464–479.
Soyster, A.L., 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. Oper. Res. 21, 1154–1157.
Zaghian, A., Mostafaei, H., 2016. An MILP model for scheduling the operation of a refined petroleum products distribution system. Oper. Res. Int. J. 16, 513–542.