Integrating MILP, Discrete-Event Simulation, and Data-Driven Models for Distributed Flow Shop Scheduling using Benders Cuts

Roderich Wallrath * and Meik B. Franke

Posted Date: 23 July 2024

doi: 10.20944/preprints202407.1794.v1

Keywords: Benders decomposition; distributed flow shop scheduling; model integration; mixed-integer programming; discrete-event simulation; data-driven optimization

Article

Integrating MILP, Discrete-Event Simulation, and Data-Driven Models for Distributed Flow Shop Scheduling Using Benders Cuts

Roderich Wallrath [1,2,*] and Meik B. Franke [1]

[1] University of Twente, Faculty of Science and Technology, Sustainable Process Technology, Drienerlolaan 5, 7522 NB Enschede, The Netherlands
[2] Bayer AG, Kaiser-Wilhelm-Allee 1, 51373 Leverkusen, Germany; m.b.franke@utwente.nl
* Correspondence: r.wallrath@utwente.nl

Abstract: Digitalization plays a crucial role in improving the performance of chemical companies. In this context, different modeling, simulation, and optimization techniques like MILP, discrete-event simulation (DES), and data-driven (DD) models are being used. Due to their heterogeneity, these techniques must be executed individually, and holistic optimization is a manual and time-consuming process. We propose Benders decomposition to combine these techniques into one rigorous optimization procedure. The main idea is that heterogeneous models can simultaneously be optimized as Benders subproblems. We illustrate this concept with the distributed permutation flow shop scheduling problem (DPFSP) and assume that a MILP, DES, and DD model exist for three flow shops. Our approach can compute bounds and report gap information on the optimal makespan for five medium-sized literature instances. The approach is promising because it enables the optimization of heterogeneous models and makes it possible to build optimization capabilities on an existing model and tool landscape in chemical companies.

Keywords: benders decomposition; distributed flow shop scheduling; model integration; mixed-integer programming; discrete-event simulation; data-driven optimization

1. Introduction

In the chemical industry, advanced modeling and optimization techniques are crucial for operational decision-making. Their relevance is growing with the digital transformation of production, increasing sustainability requirements, and tougher global competition. Chemical companies often use a variety of different modeling and optimization techniques, each adapted for specific purposes and each with their own strengths and limitations. For example, mixed integer linear programming (MILP) can capture the discrete and continuous aspects of scheduling and planning problems, but it requires problem abstraction and simplification to have acceptable solution times. Discrete-event simulation (DES) can represent the dynamic behavior from the supply chain to the shop floor level in detail and can include stochastic effects, but it is costly to optimize DES models, and the solution quality can be inferior. Data-driven (DD) methods can exploit large amounts of data and bridge the gap if first principles are not known or are too complicated, but solutions may lack interpretability and causality, and the solution reliability can be inferior.

As a result, modeling and optimization techniques may exist in many successful, but siloed applications. A unifying, rigorous optimization framework is lacking, which would enable chemical companies to move from traditional, component-oriented optimization to system-wide optimization. Different research attempts on this topic have already been made. The need for integrated methods in optimization is studied in Hooker (2012) and unified treatment of mathematical programming, constraint programming, and global optimization is suggested. However, simulation techniques and data-driven approaches are not addressed. In Azevedo et. al. (2024), a bibliometric analysis of hybrid approaches involving optimization and machine learning algorithms is conducted. It shows that there is a growing interest in combining metaheuristic approaches, such as particle swarm

 doi:10.20944/preprints202407.1794.v1

optimization, with machine learning techniques. However, the integration of rigorous optimization techniques is not covered in the analysis. In Figueira and Almada-Lobo (2014), a comprehensive overview and taxonomy of simulation-optimization methods are given and the potential for cross-fertilization between the methods is discussed. However, the authors conclude that there are gaps in the simulation-optimization interaction matrix, and they specifically suggest investigating the AME-OSI combination. In this combination, the optimization proceeds in iterations that contain one or more simulation runs (OSI), and each run is used to refine and enhance the analytical model (AME). Furthermore, unifying optimization and decision-making frameworks have already been suggested for industrial settings, for example, to support collaborative supply chain decision-making in the chemical industry (García-Flores and Wang, 2002), to optimize supplier selection for green and sustainable supply chains (Luthra et. al., 2017) and to improve operational strategies and overall decision orchestration in pharmaceutical companies (Marques et. al., 2020). However, these frameworks operate on a conceptual level, and they lack a rigorous, algorithmic integration layer that interlinks the different methods, models, and tools.

Herein, we propose a novel framework that integrates MILP, DES, and DD models into one rigorous optimization framework for production scheduling in the chemical industry. The main idea is based on Zhang et. al. (2017), Forbes et. al. (2024), and Wallrath and Franke (2024), which suggest using Benders decomposition for a MILP-DES integration. The advantage of Benders decomposition is that different model types can be considered in the different subproblems as long as Benders cuts can be generated from the subproblem solutions. In this work, we use this concept to integrate three different model types into one optimization framework. We show that the integration of MILP, DES, and DD models is possible and that the Benders master problem can guide the search process in these models efficiently while yielding dual bounds on the overall problem. We apply our approach to literature instances of the well-known distributed permutation flow shop scheduling problem (DPFSP) with makespan minimization objective, as presented in Naderi and Ruiz (2010). The DPFSP is an important real-world problem in production and supply chain management in chemical companies and is one of the fastest-growing topics in the scheduling and operations research literature (Perez-Gonzalez and Framinan, 2024). Production and supply chain management are very often related to heterogeneous methods, models, and tools that originate from the different regions, manufacturing sites, and production plants of chemical companies. To this end, we study DPFSP instances featuring three factories ($K = 3$) and assume that a MILP, a DES, and a DD model exist for these factories. To the best of our knowledge, this is the first approach that integrates different simulation and optimization methods rigorously and provides a comprehensive and flexible solution framework for the pre-existing and specialized models and tools in chemical companies.

The rest of the article is structured as follows: In Section 2, we review current approaches to distributed flow shop (DFS) optimization, and Benders decomposition. In Section 3, we present the optimization framework featuring the Benders master problem, Benders cuts, and different types of subproblems. In Section 4, we apply our approach to a literature case study of a DPFSP featuring three factories, represented by a MILP, DES, and DD model, and discuss the progression of the upper and lower bounds. In Section 5, we discuss the approach from an industrial perspective and conclude with an outlook for future research in Section 6.

2. Literature

In the following, we briefly review current techniques for DFS optimization as well as Benders decomposition.

2.1. Distributed Flow Shop Optimization

A DFS consists of a number of flow shops arranged in parallel. DFS scheduling is the task of assigning each job to one of the flow shops and sequencing all jobs in each flow shop. The decision variables, therefore, include assignment and sequence variables. Each flow shop consists of multiple machines, and the sequence of the jobs can either change between the machines (distributed non-permutation flow shop) or be fixed (distributed permutation flow shop). DFS scheduling, as initially

 doi:10.20944/preprints202407.1794.v1

mentioned in Cao and Chen (2003), can therefore be seen as a combination of parallel machine scheduling (Unlu and Manson, 2010, Vallada and Ruiz, 2011, Edis et. al., 2013) and flow shop scheduling (Johnson, 1954, Zheng and Wang, 2003, Komaki et. al., 2019), which are both well-studied problems. Reviews and classifications of DFS scheduling problems have recently been proposed (Duan et. al., 2023, Perez-Gonzalez and Framinan, 2024).

DFS scheduling problems are most commonly solved with the makespan minimization objective (Mraihi et. al., 2023), which translates into the minimization of the maximum completion time among all jobs. A standard method for DFS scheduling is mathematical programming, such as MILP (Naderi and Ruiz, 2010) or constraint programming (CP) (Gogos, 2023). A general advantage of mathematical programming lies in its ability to use the structure of closed-form models to optimize the solution process, for example, by generating cuts, solving the dual problem, and pruning the search space. However, a mathematical programming model must be set up, which is often a difficult and time-consuming task. Especially for complex systems, such as real-world DFS, building accurate and tractable mathematical models quickly becomes an impossible task from a practical point of view. In this context, heuristics and metaheuristics become important alternatives because they do not require mathematical programming models. Since they are based on the systematic evaluation of a black-box model, simulation and DD models can be used. Heuristic approaches have traditionally held a very strong position for DFS scheduling due to their simplicity and effectiveness. Based on the well-known Nawaz-Enscore-Ham (NEH) heuristic for simple flow shop problems (Nawaz et. al., 1983), an improved heuristic, that implements a novel insertion rule is proposed by Gao and Chen (2011). Closely related are iterated greedy methods, as initially proposed for permutation flow shops in Ruiz and Stützle (2007). The basic idea is to repeatedly construct an initial solution, to destruct it by removing some jobs, and then reconstruct it by re-inserting the removed jobs in a new sequence to find a potentially better solution. This iterative process, combined with local search heuristics and optimization techniques, is one of the most successful approaches to DFS scheduling and has led to multiple research contributions. In Lin et. al. (2013), a modified iterated greedy procedure is adopted for the DPFSP. To this end, a bounded search iterated greedy algorithm is presented in Fernandez-Viagas and Framinan (2015), which exploits the specific structure of DPFSP. The authors claim that their heuristic improved the state-of-the-art of existing heuristics at that time. In Ruiz et. al. (2019), an enhanced two-stage iterated greedy method is suggested, which again led to a significant performance improvement. In addition to heuristic approaches, different metaheuristics, such as scatter search (Naderi and Ruiz, 2014), taboo search (Gao et. al., 2013), and genetic algorithms (Gao and Chen, 2011) have been adopted successfully. Although they are not always as efficient as heuristic approaches, metaheuristics generally require less customization as they are based on general and versatile search strategies. In addition, many standard packages are available as open-source codes for metaheuristics. Both heuristic and metaheuristic methods are compatible with a variety of different model types, as long as they can be evaluated as a black-box model. Therefore, they could also be used to optimize heterogeneous models.

However, heuristics and metaheuristics have two crucial drawbacks. Since they are primal search methods, they do not use dual bounds to reduce the search space and do not provide optimality gap information. Hence, finding good solutions relies on smart, but ultimately brute-force, primal search strategies. The solution quality cannot be assessed by gap information, and the user must decide when to terminate the search process, leaving the potential solution improvement unknown. This work presents an alternative mathematical optimization framework that is capable of integrating heterogeneous models, uses dual bounds, and reports optimality gap information. The framework is based on Benders decomposition, which we review in the following.

2.2. Benders Decomposition

Benders decomposition (Benders, 1962) is a technique in mathematical programming that allows the splitting of the variables of an optimization problem into complicating and non-complicating variables to obtain a master problem with the complicating variables and independent subproblems with the non-complicating variables. The master and subproblems are solved individually in an

 doi:10.20944/preprints202407.1794.v1

iterative procedure in which constraints (Benders cuts) are generated from the dual information of subproblem solutions and added to the master problem. The advantage of Benders decomposition is that smaller, tractable subproblems are obtained, which can be treated, for example, using specialized algorithms. In Hamzadayı (2020), Benders decomposition is used to solve an improved, position-based MILP model of the DPSFP. The master problem contains the binary decision variables that encode the factory assignment and position of jobs. The subproblems contain the continuous decision variables, which are the completion times of jobs. A hybrid Benders decomposition algorithm is presented, which applies the local search algorithm (LS3) (Ruiz et. al., 2019) to improve each master problem solution heuristically before sending it to the subproblems. With the improved and original master problem solution, two cuts are generated in each Benders iteration, which accelerates the convergence of the algorithm. In Bektaş et. al. (2020), the mixed no-idle permutation flow shop scheduling problem is solved using a similar concept. The referenced local search heuristic (RLS) (Pan et. al., 2007) is used to generate a number of neighborhood master solutions, which lead to multiple Benders cuts. To the best of our knowledge, these are the only reported applications of Benders decomposition for (distributed) flow shop scheduling problems. Furthermore, in these applications, the subproblems are linear programs (LPs), which result from fixing the integer variables of the original MILP model. Therefore, we extend our literature analysis to Benders decomposition applied to scheduling problems in general, in which the subproblems do not directly come from the original MILP model but are of different types.

In Hooker and Ottosson (2003), the Benders decomposition concept is generalized, and logic-based Benders cuts are proposed, which may be derived from any type of subproblem as long as the cuts represent valid inequalities. The combination of a MILP master problem and CP subproblem is well-known and shows superior performance over monolithic MILP or CP models for many types of scheduling problems, such as project scheduling with multi-skilled personnel (Li and Womer, 2009), single batching machine scheduling (Emde et. al., 2020), preemptive flexible job-shop scheduling (Juvin et. al., 2023), and two-stage flexible flow shop scheduling with unrelated parallel machines (Tan and Terekhov, 2018). The idea of using a DES model as a subproblem in the Benders decomposition framework is relatively new. In one of the first examples (Zhang et. al. 2017), this concept is suggested to optimize the buffer allocation problem (BAP) in multi-stage serial-parallel manufacturing systems. The authors derive Benders cuts from a DES model to find the optimal design parameters of a joint workstation, workload, and buffer allocation system. In Forbes et. al. (2024), this concept is further developed, and improved logic-based Benders cuts are proposed for the master problem to solve operational scheduling tasks. They apply their approach to a shift scheduling problem as well as a personnel allocation problem and can solve realistic-sized instances exactly in reasonable computing times.

This work presents a novel use case for Benders decomposition. In contrast to Zhang et. al. (2017), Forbes et. al. (2024), and Wallrath and Franke (2024), we apply Benders decomposition to the DPFSP. In contrast to Hamzadayı (2020), we use three different model types in the subproblems and thereby leverage Benders decomposition as an integration framework, which can be used to combine pre-existing heterogeneous models in the chemical industry.

3. Methodology

We first introduce the concept of using Benders decomposition with heterogeneous subproblems and then adopt the concept for distributed permutation flow shops.

3.1. Benders Decomposition with Heterogeneous Subproblems

We consider the minimization problem shown in Equations (1) involving continuous variables $x \in \mathbb{R}^n$ and binary variables $y \in \{0,1\}^n$. Benders decomposition can be applied with respect to x and y such that (1) decomposes into a part (1d), which contains the combinatorial complexity and encodes the logical behaviour, and a part (1c), which involves continuous decisions, which are linked to the binary decision.

 doi:10.20944/preprints202407.1794.v1

$$\begin{aligned} &\min z && (1a)\\ &z = c^T x && (1b)\\ s.t.\quad &Ax + By \le b && (1c)\\ &Cy \le d && (1d)\\ &x \ge 0 && (1e)\\ &y \in \{0,1\}^n && (1f) \end{aligned}$$

This yields the MILP Benders master problem in (2), which computes lower bounds μ on z. The master problem consists of the constraints block (1d) of the original problem, optimality cuts $\mu \ge u^T(b - Bx)$ with $u \in O' \subset O$, and feasibility cuts $0 \ge u^T(b - Bx)$ with $u \in F' \subset F$, that are generated based on solutions of the subproblems (1c). The set O contains all dual solutions of (1), for which the dual problem is feasible and bounded, while the set F contains all dual solutions, for which the dual problem is unbounded. However, we only need to generate cuts from the subsets O' and F', since only a finite number of cuts are active constraints in the optimum of (2).

$$\begin{aligned} &\min \mu && (2)\\ &\mu \ge u^T(b - By) \quad \forall u \in O' && (2)\\ &0 \ge u^T(b - By) \quad \forall u \in F' && (2)\\ \text{s.t.}\quad &Cy \le d && (2)\\ &\mu \in \mathbb{R} && (2)\\ &y \in \{0,1\}^n && (2) \end{aligned}$$

In each Benders iteration, we obtain a solution vector $\hat{y}$ from (2). Based on $\hat{y}$, we solve the individual subproblems of the form (3), where $\hat{y}_s$ is the part of the solution vector $\hat{y}$ that corresponds to subproblem $s \in S$, x_s is the corresponding part of the decision vector x, and A_s, B_s, b_s, and c_s^T describe subproblem s.

$$\begin{aligned} &\min z_s && (3)\\ &z_s = c_s^T x_s && (3)\\ \text{s.t.}\quad &A_s x_s + B_s \hat{y}_s \le b_s && (3)\\ &x_s \ge 0 && (3) \end{aligned}$$

We argue that this classical Benders decomposition framework can be used to integrate LP, DES, and DD models as illustrated in Figure 1. The basic idea is to generate Benders cuts from an LP model f_{LP}, a DES model f_{DES}, and a DD model f_{DD} which act as the Benders subproblems, such that $S = \{LP, DES, DD\}$. For the LP subproblem, the dual multipliers u_{LP} can be directly retrieved from the dual solution. For the DES subproblem, the dual multipliers u_{DES} correspond to the sensitivity information of the DES solutions and can be derived using a dual mapping. For the DD subproblem, no dual multipliers can be derived and only the objective value information z_{DD} can be used to generate logic-based Benders cuts of the form $\mu \ge E_{DD} y_{DD} + e_{DD}$, where the cut parameters E_{DD} and e_{DD} are determined through the evaluation of the DD model. For all three models feasibility cuts can be formulated which exclude $\hat{y}_s$ from the feasible set of the master problem. In the following, we apply this concept to the DPFSP.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 July 2024 doi:10.20944/preprints202407.1794.v1

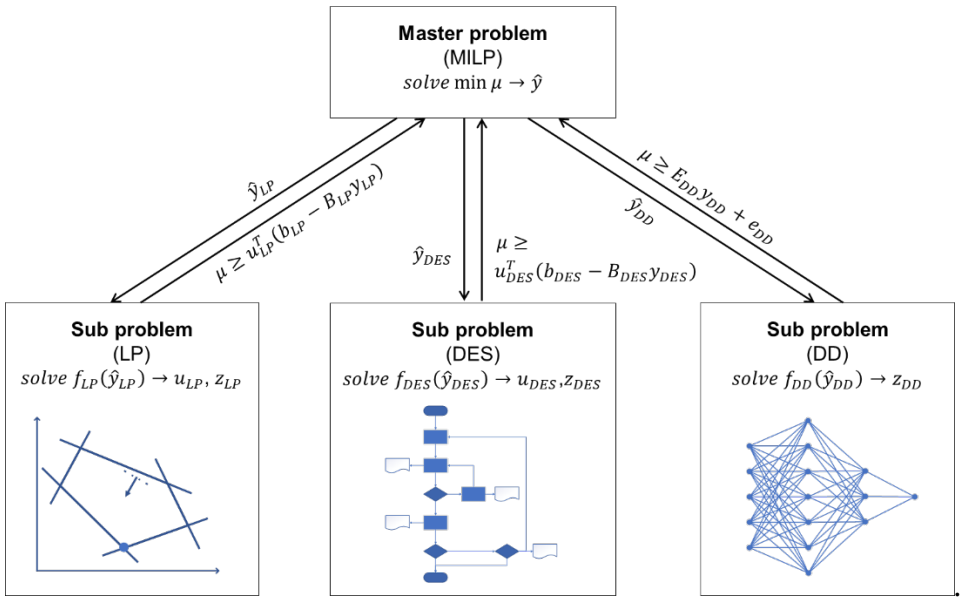

Figure 1. Integration concept of (MI)LP, DES, and DD using Benders decomposition.

3.2. Distributed Permutation Flow Shop Model

We consider the DPFSP as reviewed in Section 2.1 and assume the makespan minimization objective. The Benders master problem (4) is based on job sequencing variables y_{ijf}, which are equal to one if job i precedes job j on factory f and are zero otherwise, as well as job assignment variables a_{if}, which are equal to one if job i is assigned to factory f and are zero otherwise. Constraints (4e) ensure that each job is assigned to exactly one factory, and constraints (4f) enforce the precedence relation between two jobs that are assigned to the same factory. Constraints (4g) and (4h) assign an artificial job with index zero to each factory and schedule it as the first job. As a result, it is possible to express the precedence relation of all real jobs i, j by $y_{ijf} \neq 0$, which is needed to build the Benders optimality cuts. We do not need to consider feasibility cuts for the DPSFP because the flow shop subproblems always return primal feasible solutions when a set of jobs and permutation is assigned to them. Constraints (4b) are the optimality cuts from the LP subproblem, constraints (4c) the optimality cuts from the DES subproblem, and constraints (4d) the approximative optimality cuts from the DD subproblem. The parameter vector t_s contains the processing times of all jobs assigned to subproblem s, and T is a Big-M factor. The underlying assumption of the cuts (4d) is that each job assigned to the DD subproblem contributes to the makespan z_{DD} equally, where N_{DD} is the number of jobs assigned to the DD subproblem. Therefore, we compute the average makespan contribution per job $\frac{z_{DD}}{N_{DD}}$ and assume that a change in the job permutation causes a relaxation by $\frac{z_{DD}}{N_{DD}}$. While this approximation may exclude combinatorial effects on the makespan and therefore may cut off the global optimum, it strengthens the cuts. Furthermore, we select these approximative optimality cuts for the DD subproblem, because we introduce some model inaccuracies compared to the LP and DES model, as we use a deep neural network to approximate the flow shop of the DD subproblem.

$$\min \mu \tag{4a}$$

$$\text{s.t.} \quad \mu \geq u_{LP}^T(t_{LP} - T(1 - y_{LP})) \forall\ u_{LP} \in O'_{LP} \tag{4b}$$

$$\mu \geq u_{DES}^T(t_{DES} - T(1 - y_{DES})) \forall\ u_{DES} \in O'_{DES} \tag{4c}$$

$$\mu \geq y_{DD}^T \left(\frac{z_{DD}}{N_{DD}}, \dots, \frac{z_{DD}}{N_{DD}}\right) \forall\ N_{DD}, z_{DD} \in O'_{DD} \tag{4d}$$

$$\sum_{f=1}^{K} a_{if} = 1 \ \forall\ i \in \{1, \dots, N\} \tag{4e}$$

$$y_{ijf} + y_{jif} = a_{if} a_{jf} \ \forall\ i, j \in \{1, \dots, N\} \wedge i \neq j, \qquad f \in \{1, \dots, F\} \tag{4f}$$

 doi:10.20944/preprints202407.1794.v1

$$a_{0f} = 0 \; \forall \, f \in \{1, \dots, K\} \quad (4g)$$

$$y_{i0f} = 0 \; \forall \, i \in \{1, \dots, N\}, f \in \{1, \dots, K\} \quad (4h)$$

In each Benders iteration, the master problem (4) is solved, which yields a lower bound on the makespan c_{max} of the DPFSP as well as a job assignment and permutation for each factory subproblem. The subproblems compute the dual multipliers $\{u_{LP}, u_{DES}\}$ and individual makespans $\{z_{LP}, z_{DES}, z_{DD}\}$. Based on this primal and dual feedback, Benders cuts are formulated and added to (4), and an upper bound on c_{max} is computed. The resulting algorithm is shown in Appendix A.3, Algorithm 1. In the following, we introduce the flow shop subproblems, which are represented by different model types.

LP subproblem. The LP subproblem represents factory one and consists of a MILP model (with fixed integer variables) of a permutation flow shop featuring m machines as shown in (5). Given an input job permutation $\hat{y}_{LP}$, the competition time c_{im1} of each job i assigned to this factory is computed for each machine m. The objective is to minimize the makespan $z_{LP} = c_{max,1}$. The intra-machine constraints (5b) enforce the job sequence induced by $\hat{y}_{LP}$, where t_{jm1} is the processing time of job i on machine m in factory 1 and T is a Big-M factor. The inter-machine constraints (5c) enforce valid competition times as jobs move from one machine m to the next $m+1$ in the flow shop. From each solution of this model, dual multipliers u_{LP} are obtained, which correspond to the active constraints (5b) and (5c) given the job permutation $\hat{y}_{LP}$. Based on u_{LP}, the Benders cuts (4b) are built.

$$min \; c_{max,1} \quad (5a)$$

$$s.t. \quad c_{im1} + t_{jm1} \leq c_{jm1} + T\left(1 - \hat{y}_{ij1}\right) \forall \, i, j \in \{1, \dots, N\} \wedge i \neq j, m \in \{1, \dots, M\} \quad (5b)$$

$$c_{im1} + t_{im1} \leq c_{i(m+1)1} + T(1 - \hat{a}_{i1}) \; \forall \; i \in \{1, \dots, N\}, m \in \{1, \dots, M-1\} \quad (5c)$$

$$c_{max,1} \geq c_{iM1} \qquad \forall \, i \in \{1, \dots, N\} \quad (5d)$$

DES subproblem. The DES subproblem represents factory two and consists of a DES model of a permutation flow shop featuring m machines as shown in Appendix A.4, Figure 4. Given an input job permutation $\hat{y}_{DES}$, the completion times c_{im2} of each job i assigned to this factory are computed for each machine m. The objective is to minimize the makespan $z_{DES} = c_{max,2}$ of this factory. From each solution of the DES model, dual multipliers u_{DES} are computed using the concept of critical path mapping as described in Wallrath and Franke (2024). Based on u_{DES}, the Benders cuts (4c) are built.

DD subproblem. The DD subproblem represents factory three and consists of a deep neural network model as specified in Table 1. The model input is a job permutation $\hat{y}_{DD}$ and the output is the makespan $z_{DD} = c_{max,3}$. The model was trained on 400000 randomly generated job permutations, based on which makespans were computed using a DES model. The training progression of one literature instance is shown in Figure 2. Note that we had to build a DD model artificially by sampling the DES model because we want to illustrate the functionality of Benders decomposition for heterogeneous models. In real-world settings, DD models often pre-exist and are the only way to query complex production data sets for KPIs like the makespan.

Table 1. Specifications of deep neural network model of subproblem three.

Parameter	Value
Input	job permutation $\hat{y}_{DD}$
Output	makespan $c_{max,3}$
Number of samples	400000
Sampling mode	random uniform
Number of hidden layers	5
Number of nodes in hidden layers	600, 500, 400, 300, 100
max_iter	1000
DD modeling package	sklearn.neural_network, MLPRegressor

 doi:10.20944/preprints202407.1794.v1

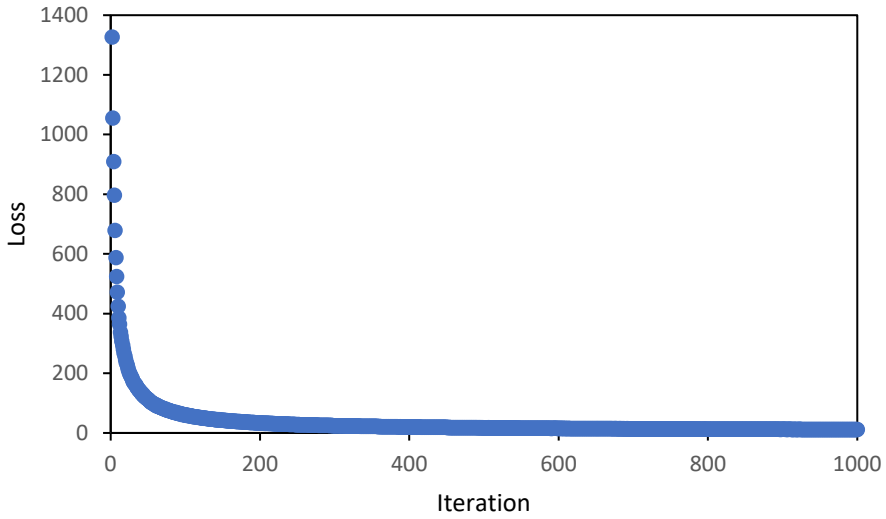

Figure 2. Training progression over 1000 iterations of the deep neural network model for subproblem three, literature DPFSP instance 85.

4. Computational Studies

We consider the five DPFSP instances[1], as presented in Naderi and Ruiz (2010) featuring $K = 3$ factories, $M = 5$ machines, and $N = 10$ jobs. The literature IDs of the instances we used are given in Appendix A.2, Table 3. According to Naderi and Ruiz (2010), the number of possible solutions for instances of this size is $66 * 10! = 239500800$. We analyze the upper and lower bound progression reported by the Benders algorithm with respect to the Benders iterations. Figure 3 shows that for all instances, meaningful lower bounds are found. Both lower bounds (LB) and upper bounds (UB) improve as the algorithm progresses. For four of the five instances, the Benders algorithm approximates the best-known literature solution[1] through LBs and UBs. This confirms our concept of rigorous optimization for heterogeneous models. For instance 86, a better upper bound than the best-known literature solution is found.

However, for all instances, the remaining optimality gap $\delta = (UB - LB)/UB$ after 3000 Benders iterations and approximately ten hours of solution time is still more than 29 percent as shown in Appendix A.1, Table 2. This is due to the large number of possible solutions compared to the relatively small number of iterations and the fact that the Benders cuts cannot cover the large, combinatorial decision space sufficiently. We build the Benders cuts based on precedence variables y_{ijf} and Big-M factors T for the LP and DES subproblem to obtain a strong relaxation mechanism in the cuts. This mechanism is necessary, because a master solution that differs in one element y_{ijf} from a previously evaluated master solution, might lead to a completely different makespan. Therefore a relaxation by T is required to account for combinatorial effects for the exact LP and DES models. Furthermore, as the processing times of jobs are not factory-specific and we only use simple Benders cuts, symmetric solutions arise, which may add to the slow convergence. They could be addressed using symmetry-breaking constraints which could lead to an improvement in the effectiveness of the cuts.

[1] Instances and solutions taken from http://soa.iti.es/problem-instances

 doi:10.20944/preprints202407.1794.v1

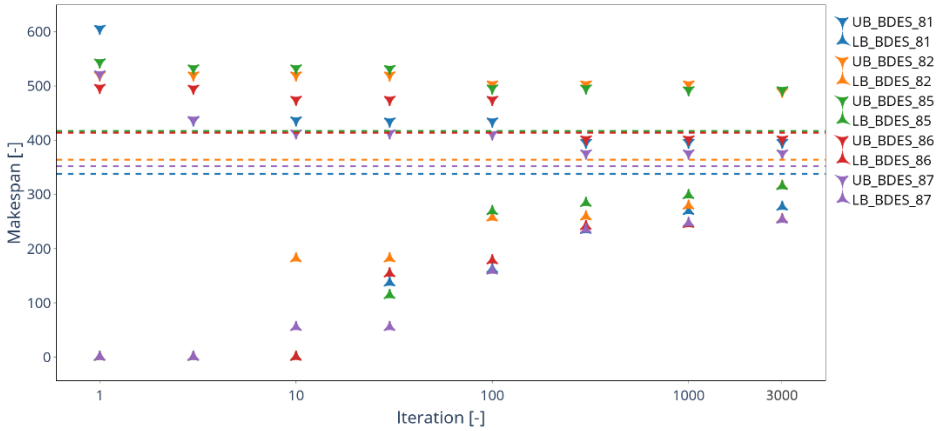

Figure 3. Upper bounds (UB) and lower bounds (LB) progression of Benders algorithm compared to the best literature upper bounds (dashed lines) for five literature DSFSP instances.

5. Discussion

The results show that Benders decomposition can be used to integrate heterogeneous flow shop problems and to solve DFS scheduling problems with makespan minimization objective. In the following, we discuss this approach from an industrial perspective and highlight two implications for supply chain and operations management.

In the decision-making hierarchy of chemical companies, the link between supply chain planning and site manufacturing is critical for optimal operations. A common approach is to optimize a global supply chain model resulting in setpoints and production targets for sites and plants. The shop floor level is not considered in detail in the global model and usually must be flexible enough to fulfill the propagated targets. On the other hand, detailed process models may exist on the site or plant level, which include all relevant shop floor constraints, such as process utilities, personnel, logistics, purchasing, maintenance, and downtime constraints. The Benders algorithm can be used to integrate these models with the global supply-chain model ensuring optimized and feasible operations. Assessing the impact of changes at the supply chain level on production can then help to reduce the cost of flexibility in production.

In chemical companies, often a range of different, heterogeneous models and tools pre-exist. The integration of these models into a single optimization framework is one advantage of the proposed approach. In addition, the ability to combine different model types gives modeling experts the freedom to select the best-fitting modeling technique for a given problem. While DES models may be suitable to describe large-scale systems with many constraints in continuous time, MILP can be used when systems are easy to describe but difficult to optimize. Similarly, DD or hybrid models might be the best-fitting solution for specific problems. As long as these models can be queried as black-box models and Benders cuts can be formulated, potentially based on dual information, they can be used in the proposed framework.

6. Conclusion and Outlook

In this study, we showed how Benders decomposition can be used as an integration concept for heterogeneous models, that is, to combine models of different types into one rigorous optimization framework. We illustrated this concept with the class of the DPFSP and assumed that a MILP, DES, and DD model exist for three different flow shops. To this end, we solved five literature instances of the DPFSP from Naderi and Ruiz (2010) and observed that the proposed algorithm can report upper and lower bounds as well as remaining gap information on the optimal makespan.

This work is early, conceptual research along with the first computational studies. There are areas of improvement and potential for future work. The remaining gap after 3000 Benders iterations and approximately ten hours of solution time is still more than 29 %, which highlights the issues of

 doi:10.20944/preprints202407.1794.v1

the convergence that still need to be addressed. Therefore, future work should focus on cut generation, for example by combining the approach with a branch and Benders cut algorithm (Gendron et. al., 2016) or by deriving more no-good cuts in each Benders iteration using similar procedures as the CB separation heuristic in Codato and Fischetti (2006). In addition, with the topic of DFS scheduling gaining more attention over the last few years, it would be interesting to apply the approach to a real-world industrial case study. The ability to perform rigorous optimization on systems with heterogeneous models also has interesting practical advantages, such as optimality gap information and system integration, which could be explored further from this basis.

Symbol list

Symbol	Description
K	Number of factories
M	Number of machines
N	Number of jobs
x	Continuous decision variables
y	Binary decision variables
z	Generic objective function
c	Generic objective function coefficient vector
A, B, b, d	Generic constraint coefficient matrices and right-hand sides
μ	Generic objective function of Benders master problem
u	Generic dual multiplier
O'	Set of specific dual vectors of optimality cuts of Benders master problem
O	Set of all dual vectors of optimality cuts of Benders master problem
F'	Set of specific dual vectors of feasibility cuts of Benders master problem
F	Set of all dual vectors of feasibility cuts of Benders master problem
s	Specific subproblem of all considered subproblems S
f_s	Function notation of a specific subproblem $s \in S$
E_{DD}, e_{DD}	Logic-based Benders cuts parameters from DD subproblem
y_{ijf}	Job sequence variable
a_{if}	Job assignment variable
T	Big-M factor
N_{DD}	Number of jobs assigned to DD subproblem
c_{imf}	Completion time of a job
t_{imf}	Processing time of a job
δ	Current optimality gap

Appendix

A. 1

Table 2. Current gap of Benders algorithm after 3000 iterations.

Instance	Remaining gap δ
81	0.29
82	0.35
85	0.36
86	0.44
87	0.32

 doi:10.20944/preprints202407.1794.v1

A. 2

Table 3. Literature ID of instances from http://soa.iti.es/problem-instances.

Instance	Literature name
81	l_3_10_5_1.txt
82	l_3_10_5_2.txt
85	l_3_10_5_3.txt
86	l_3_10_5_4.txt
87	l_3_10_5_5.txt

A. 3

Algorithm 1: Benders Algorithm

Input: DPFSP featuring MILP, DES, and DD subproblem

Output: Makespan-optimized schedule (c_{max}, δ) containing job assignments a_{if} and job sequences y_{ijf}

Initialize: $c_{max} = \infty,\ \delta = \infty,\ k = 0, t = 0$

while (k < maximum iterations and t < timeout and δ > 0) do

solve Benders master problem and obtain μ^k

solve subproblems with current master solution a_{if}^k, y_{ijf}^k

obtain makespan $c_{max}^k = \max(c_{max,1}^k, c_{max,2}^k, c_{max,3}^k)$

derive dual multipliers u_{LP}^k, u_{DES}^k and parameters E_{DD}, e_{DD}

build Benders cut for MILP, DES, and DD subproblem

add Benders cuts to the master problem

if ($c_{max}^k < c_{max}$) do

update upper bound: $c_{max} \leftarrow c_{max}^k$

update current gap: $\delta \leftarrow \frac{c_{max} - \mu^k}{c_{max}}$

increment k and t

A. 4

 doi:10.20944/preprints202407.1794.v1

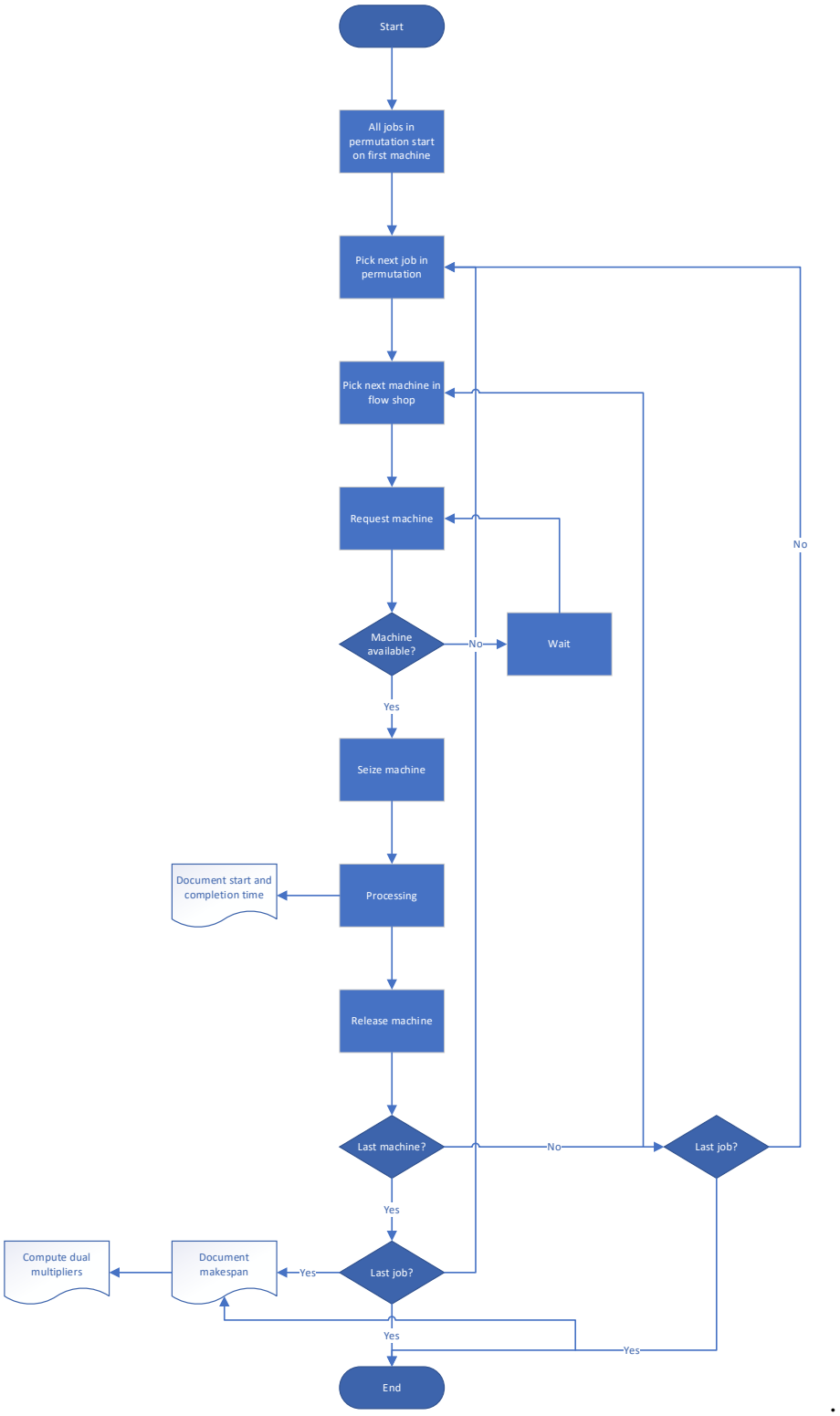

Figure 4. Block diagram of permutation flow shop DES model of factory two.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

References

1. Hooker, John. (2012). Integrated Methods for Optimization. 10.1007/978-1-4614-1900-6.
2. Azevedo, B.F., Rocha, A.M.A.C. & Pereira, A.I. Hybrid approaches to optimization and machine learning methods: a systematic literature review. Mach Learn (2024).

3. Gonçalo Figueira, Bernardo Almada-Lobo, Hybrid simulation–optimization methods: A taxonomy and discussion, Simulation Modelling Practice and Theory, Volume 46, 2014, Pages 118-134, ISSN 1569-190X
4. Benders, J.F., 1962, Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252
5. Wallrath, Roderich, and Meik B. Franke. "Integration of MILP and Discrete-Event Simulation for Flow Shop Scheduling Using Benders Cuts." Computers & Chemical Engineering (2024): 108809.
6. McNaughton, R. (1959). Scheduling with Deadlines and Loss Functions. Management Science, 6(1), 1–12. http://www.jstor.org/stable/2627472
7. B. Naderi, Rubén Ruiz, The distributed permutation flowshop scheduling problem, Computers Operations Research, Volume 37, Issue 4, 2010, Pages 754-768, ISSN 0305-0548
8. Paz Perez-Gonzalez, Jose M. Framinan, A review and classification on distributed permutation flowshop scheduling problems, European Journal of Operational Research, Volume 312, Issue 1, 2024, Pages 1-21, ISSN 0377-2217
9. Yasin Unlu, Scott J. Mason, Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems, Computers & Industrial Engineering, Volume 58, Issue 4, 2010, Pages 785-800, ISSN 0360-8352
10. Eva Vallada, Rubén Ruiz, A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times, European Journal of Operational Research, Volume 211, Issue 3, 2011, Pages 612-622, ISSN 0377-2217
11. Emrah B. Edis, Ceyda Oguz, Irem Ozkarahan, Parallel machine scheduling with additional resources: Notation, classification, models and solution methods, European Journal of Operational Research, Volume 230, Issue 3, 2013, Pages 449-463, ISSN 0377-2217
12. S. M. Johnson, 1954. "Optimal two- and three-stage production schedules with setup times included," Naval Research Logistics Quarterly, John Wiley & Sons, vol. 1(1), pages 61-68, March.
13. Zheng, DZ., Wang, L. An Effective Hybrid Heuristic for Flow Shop Scheduling. Int J Adv Manuf Technol 21, 38–44 (2003)
14. Komaki, G. M., Sheikh, S., & Malakooti, B. (2019). Flow shop scheduling problems with assembly operations: a review and new trends. International Journal of Production Research, 57(10), 2926–2955
15. Gogos C. Solving the Distributed Permutation Flow-Shop Scheduling Problem Using Constrained Programming. Applied Sciences. 2023; 13(23):12562
16. Bahman Naderi, Rubén Ruiz, A scatter search algorithm for the distributed permutation flowshop scheduling problem, European Journal of Operational Research, Volume 239, Issue 2, 2014, Pages 323-334, ISSN 0377-2217
17. Gao, J., Chen, R., & Deng, W. (2013). An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. International Journal of Production Research, 51(3), 641–651
18. Gao, J., & Chen, R. (2011). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. International Journal of Computational Intelligence Systems, (4), 497–508
19. Hamzadayı, A. (2020). An effective benders decomposition algorithm for solving the distributed permutation flowshop scheduling problem. Computers & Operations Research, 123, 105006.
20. Cao, D., & Chen, M. (2003). Parallel flowshop scheduling using Tabu search. International journal of production research, 41(13), 3059-3073.
21. Duan, J., Wang, M., Zhang, Q., & Qin, J. (2023). Distributed shop scheduling: A comprehensive review on classifications, models and algorithms. Mathematical Biosciences and Engineering, 20(8), 15265-15308.
22. Lin, S. W., Ying, K. C., & Huang, C. Y. (2013). Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. International Journal of Production Research, 51(16), 5029–5038. https://doi.org/10.1080/00207543.2013.790571
23. Fernandez-Viagas, V., & Framinan, J. M. (2015). A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem. International Journal of Production Research, 53(4), 1111–1123.
24. Gao, J., & Chen, R. (2011). An NEH-based heuristic algorithm for distributed permutation flowshop scheduling problems. Scientific Research and Essays, 6(14), 3094-3100.
25. Muhammad Nawaz, E Emory Enscore, Inyong Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega, Volume 11, Issue 1, 1983, Pages 91-95, ISSN 0305-0483
26. Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European journal of operational research, 177(3), 2033-2049.
27. Mraihi, T., Driss, O. B., & El-Haouzi, H. B. (2023). Distributed permutation flow shop scheduling problem with worker flexibility: Review, trends and model proposition. Expert Systems with Applications, 121947.
28. Ruiz, R., Pan, Q. K., & Naderi, B. (2019). Iterated Greedy methods for the distributed permutation flowshop scheduling problem. Omega, 83, 213-222.
29. Bektaş, T., Hamzadayı, A. & Ruiz, R. Benders decomposition for the mixed no-idle permutation flowshop scheduling problem. J Sched 23, 513–523 (2020).

30. Pan, Q. Q., Tasgetiren, M. F., & Liang, Y. C. (2007, July). A discrete differential evolution algorithm for the permutation flowshop scheduling problem. In Proceedings of the 9th annual conference on Genetic and evolutionary computation (pp. 126-133).
31. Juvin, C., Houssin, L., & Lopez, P. (2023). Logic-based Benders decomposition for the preemptive flexible job-shop scheduling problem. Computers & Operations Research, 152, 106156.
32. Tan, Y., Terekhov, D. (2018). Logic-Based Benders Decomposition for Two-Stage Flexible Flow Shop Scheduling with Unrelated Parallel Machines. In: Bagheri, E., Cheung, J. (eds) Advances in Artificial Intelligence. Canadian AI 2018. Lecture Notes in Computer Science(), vol 10832. Springer, Cham
33. Li, H., Womer, K. Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. J Sched 12, 281–298 (2009).
34. Emde, Simon, Lukas Polten, and Michel Gendreau. "Logic-based benders decomposition for scheduling a batching machine." Computers & Operations Research 113 (2020): 104777.
35. Gendron, Bernard & Scutellà, Maria & Garroppo, Rosario & Nencioni, Gianfranco & Tavanti, Luca. (2016). A Branch-and-Benders-Cut Method for Nonlinear Power Design in Green Wireless Local Area Networks. European Journal of Operational Research. 255. 10.1016/j.ejor.2016.04.058.
36. Codato, G., & Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. Operations Research, 54(4), 756-766.
37. M. Zhang, A. Matta, A. Alfieri and G. Pedrielli, 2017, A simulation-based benders' cuts generation for the joint workstation, workload and buffer allocation problem, 13th IEEE Conference on Automation Science and Engineering (CASE) , Xi'an, China, 2017, pp. 1067-1072, doi: 0.1109/COASE.2017.8256247
38. M.A. Forbes, M.G. Harris, H.M. Jansen, F.A. van der Schoot, T. Taimre, 2024, Combining optimisation and simulation using logic-based Benders decomposition, European Journal of Operational Research,Volume 312, Issue 3,2024,Pages 840-854, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2023.07.032.
39. Hooker, J., Ottosson, G., 2003, Logic-based Benders decomposition. Math. Program., Ser. A 96, 33–60
40. Luthra, S., Govindan, K., Kannan, D., Mangla, S. K., & Garg, C. P. (2017). An integrated framework for sustainable supplier selection and evaluation in supply chains. Journal of cleaner production, 140, 1686-1698.
41. García-Flores, R., & Wang, X. Z. (2002). A multi-agent system for chemical supply chain simulation and management support. Or Spectrum, 24, 343-370.
42. Marques, C. M., Moniz, S., de Sousa, J. P., Barbosa-Povoa, A. P., & Reklaitis, G. (2020). Decision-support challenges in the chemical-pharmaceutical industry: Findings and future research directions. Computers & Chemical Engineering, 134, 106672.