



Single station MILP scheduling in discrete and continuous time

María-Luisa Muñoz-Díaz¹ · Alejandro Escudero-Santana¹ ·
Antonio Lorenzo-Espejo¹ · Roel Leus²

Accepted: 15 January 2024
© The Author(s) 2024

Abstract

This article focuses on production planning in the metallurgical sector. This study undertakes a detailed comparative study of mixed-integer linear programming models using different time representations: continuous and discrete. The analysis shows that the continuous model consistently outperforms its discrete counterpart in all evaluated scenarios. The key difference between the continuous and discrete models is the continuous model's ability to deliver better makespan results, achieving an improvement of up to 15% compared to the discrete model. This advantage holds even in complex environments with a high number of tasks and machines, where the continuous model consistently outperforms the discrete model by over 6% in the scenario with the highest number of tasks and machines. This preference extends beyond makespan considerations. The continuous model also maintains an edge in terms of runtime efficiency, achieving better times with a 99% improvement over the discrete model in all scenarios except one. These findings provide concrete evidence for the use of continuous models, which promise more effective production planning in analogous manufacturing domains.

Alejandro Escudero-Santana, Antonio Lorenzo-Espejo and Roel Leus have contributed equally to this work.

✉ María-Luisa Muñoz-Díaz
mmunozd@us.es

Alejandro Escudero-Santana
alejandroescudero@us.es

Antonio Lorenzo-Espejo
alorenzo@us.es

Roel Leus
roel.leus@kuleuven.be

¹ Department of Industrial Organisation and Business Management II, University of Seville, Camino de los Descubrimientos, s/n, 41092 Seville, Spain

² Research group ORSTAT, Faculty of Economics and Business, Naamsestraat 69, 3000 Leuven, Belgium

Keyword Scheduling, MILP, Single station, Continuous, Discrete

1 Introduction

The last few decades have seen significant advances in computing power, as well as increased pressure to optimise the efficiency of all aspects of production processes and reduce associated costs. This has naturally led to a significant increase in interest in production planning techniques, both in industry and academia.

In general terms, production planning is a decision-making process aimed at determining when, where, how, and what to produce from a given set of products that have defined characteristics and requirements. This process is carried out by managing a set of finite resources and is in most cases subject to time constraints from the outset, as indicated in the reference (Floudas and Lin 2004).

Mathematical programming, especially Mixed Integer Linear Programming (MILP), is widely used for production scheduling processes due to its rigour, versatility, and extensive modelling capabilities (Floudas and Lin 2005). Models assessing scheduling for a single station are commonly encountered in this field, mainly because they are highly applicable in three primary scenarios. The initial challenge lies with intricate multi-stage problems that can be deconstructed into simpler, self-reliant, or loosely connected single-stage problems (Pinto and Grossmann 1995); thus, handling production planning in a modular manner. The second application area is in systems where a station acts as a significant bottleneck. In such cases, production planning for the particular station is critical for the entire system. Consequently, this planning can be extended to the remaining stations (Díaz-Ramírez and Huertas 2018; Elekidis et al. 2019; Marchetti and Cerdá 2009a). Ultimately, it is relevant to issues with a solitary station (Aguirre and Papageorgiou 2018; Méndez and Cerdá 2002).

When developing mathematical models, one of the initial decisions to consider is how time should be represented (Floudas and Lin 2005; Sung and Maravelias 2009). Consequently, three primary classes of MILP models are identified: those formulated with a complete planning horizon in discrete time, those formulated in continuous time, and hybrid models. Despite the extensive literature dealing with such problems, there are no studies that analyse, for the same problem, the response of a continuous model and that of a discrete model.

Acknowledging the relevance of such problems and the existing gap in the literature to date, this study will introduce two MILP mathematical models to solve the same single station problem, one with continuous time representation and the other with discrete representation. Both models will be tested using a database containing actual time estimates from a factory with these characteristics. The results of this experiment will be presented and compared.

The structure of this paper is as follows. Section 2 defines the core problem and situates it within the existing literature. Section 3 presents the continuous and discrete models and is further divided into two subsections, one dedicated to each model. Section 4 presents a comprehensive analysis of the results obtained by

applying these models and draws important conclusions. Finally, the last section provides a list of all references cited throughout this document.

2 Problem definition

As noted previously, there exist three categories of models that are established on the basis of representative time: continuous, discrete, and hybrid models. In the continuous formulation, the planning horizon is partitioned as a component of the optimisation process, allowing the placement of tasks at any point in the timeline (Roslöf et al. 2001; Harjunkoski and Grossmann 2002; He et al. 2017). On the contrary, discrete models are founded on dividing the timeline into different intervals or time units of generally equal size (Liu et al. 2010b; Lee et al. 2002; Mouret et al. 2011). The model is constrained to schedule tasks at specific time points due to this discretization. Less frequently, there are mixed formulations in which discrete periods, such as weeks, are already predetermined within the temporal horizon (Aguirre and Papageorgiou 2018; Liu et al. 2010a; Chen et al. 2008). However, in each of these periods, there is a continuous representation in use. However, according to the review of the literature on MILP models of a single station conducted by Muñoz-Díaz et al. (2022), there are no articles that develop continuous and discrete representations to solve the same problem. This study presents and encodes both versions for future comparison.

Once the difference in time representation has been established, the remaining characteristics of the issue and models are specified. This will be identical for both the continuous and the discrete versions. It is essential to note that the problem described in this work is based on a real factory within the metallurgical sector. This factory and its processes have already been studied in the literature (Muñoz-Díaz et al. 2024; Lorenzo-Espejo et al. 2022). Therefore, this research contributes not only to the existing scientific literature, but also to the decision-making process for production planning in an actual facility.

Starting with the machinery, in the station under consideration there are parallel machines, all of which have to perform the same operation. However, these machines are not identical, and their production speeds are not proportional. In other words, the production speed depends not only on the machine on which a task is performed but also on the nature of the specific task being performed. This distinction is a key factor that distinguishes single-station models with parallel machines. In this scenario, we encounter a problem with Unrelated Parallel Machines (Brucker 2007), where all machines are capable of processing all jobs.

In terms of the objectives of the models, the ultimate goal is always to obtain the assignment of tasks to machines and the sequencing within each machine. However, there are many feasible solutions that fulfil this objective, so it is necessary to establish a criterion for optimisation among all the possible options. This criterion is the objective function of the model and is generally divided into two categories: economic functions and time-based functions (Merchan and Maravelias 2014). In this case, it is a time-based function, specifically minimising the maximum

completion time or makespan. In other words, the objective is to minimise the time that elapses between the start of processing the first task and the completion of the last.

At this point, it is important to emphasise that we are not dealing with batches but with individual units. This implies that there is no need to include the batch formation process in either the model or the objective function, contrary to the approach taken by other authors in similar problems (Berber et al. 2007; Méndez and Cerdá 2003). With the criterion for selecting a production schedule now established, we will proceed to explain the most relevant features in the context of the current problem.

The first characteristic to consider is the presence of setups. In production planning models, there are different types of setups: those that depend on the sequence (Liu et al. 2008; Chen et al. 2008), on the machine where the task is to be performed (Sun and Xue 2009; He et al. 2017), on the specific tasks to be processed (Mouret et al. 2011) or on certain resources specific to real plants (Méndez and Cerdá 2002). In this case, we will work with task-dependent setups, specifically based on the sequence of these tasks.

In addition, there will be specific mandatory precedence, i.e. there will be pairs of jobs that must be processed on the same machine and in immediate succession. Finally, there will be no consideration of delivery times for jobs, and it will be assumed that the machines are fully available.

With all this, we have an Unrelated Parallel Machine Scheduling Problem that can be classified according to Graham et al. (1979) as a $R|prec|C_{\max}$, where R means that the machines are Unrelated Parallel Machines, $prec$ refers to the fact that compulsory precedence relations are specified, and finally, C_{\max} indicates the Objective Function of the problem, minimising the maximum completion time.

3 Mixed integer linear programming models

In this section we present the mathematical formulation of the problem described above. First, for the continuous time representation (subsection 3.1), and second, for the discrete representation (subsection 3.2). Before that, the common parameters, indices and variables for both models are listed below (Table 1).

3.1 Continuous model

This section presents the Mixed Linear Programming model in its continuous time representation. As mentioned earlier, this formulation splits the time horizon as part of the optimisation process. In this representation, there are different approaches, typically classified as global-event based (Shaik et al. 2006), unit-specific-event based (Floudas and Lin 2004), slot based (Pan et al. 2009), and precedence based (Marchetti and Cerdá 2009b). In this case, we will focus on the last approach, which

Table 1 Common parameters, indices and variables in both MILP models

Parameters and indices	
i, j	Task indices; $i, j \in \{0, 1, 2, \dots, T\}$ in the continuous model and $i, j \in \{1, 2, \dots, T\}$ in the discrete model
T	Number of tasks
k	Machine index; $k \in \{1, 2, \dots, M\}$
M	Number of machines
$p_{i,k}$	Processing time of task i on machine k
$z_{i,k}$	Binary parameters to specify mandatory precedences;
$z_{i,j} = \begin{cases} 1 & \text{if task } i \text{ has to precede task } j \\ 0 & \text{otherwise} \end{cases}$	
E	A large number
Variables	
C_{max}	Maximum completion time: Makespan
C_i	Completion Time task i

Table 2 Specific variables of the continuous MILP model

Variables	
$y_{i,j,k}$	Binary variables to determine sequencing and assignment;
$y_{i,j,k} = \begin{cases} 1 & \text{if task } i \text{ is scheduled immediately before task } j \text{ on machine } k \\ 0 & \text{otherwise} \end{cases}$	

Table 3 Specific parameters, index and variables of the discrete MILP model

Parameters and index	
u	Period index, $u \in \{1, 2, \dots, W_{max}\}$
W_{max}	Parameter that delimits the length of the planning horizon
$W_{i,k}$	Parameters that defines the time window of task i on machine k . $W_{i,k} \in \{1, 2, \dots, W_{max} - p_{i,k} + 1\}$
Variables	
$y_{i,u,k}$	Binary variables to determine sequencing and assignment.;
$y_{i,u,k} = \begin{cases} 1 & \text{if task } i \text{ is scheduled to start in period } u \text{ on machine } k \\ 0 & \text{otherwise} \end{cases}$	

uses specific variables to represent the start or finish of tasks and binary variables to represent direct precedence relationships between different tasks.

The variables used exclusively for this formulation are presented below (Table 2). This is followed by the MILP model in its continuous formulation and, finally, an explanation of each of the constraints.

$$\text{Min} \quad C_{\max} \quad (1)$$

$$\text{s.a.} \quad C_{\max} \geq C_i \quad \forall i = 1, 2, \dots, T \quad (2)$$

$$\sum_{j=1}^T y_{0jk} = 1 \quad \forall k \quad (3)$$

$$\sum_{k=1}^M \sum_{i \in T \setminus \{j\}} y_{ijk} = 1 \quad \forall j = 1, 2, \dots, T \quad (4)$$

$$\sum_{i \in T \setminus \{j\}} y_{ijk} = \sum_{l \in T \setminus \{j\}} y_{jlk} \quad \forall j = 1, 2, \dots, T; \forall k \quad (5)$$

$$\sum_{k=1}^M y_{ijk} \geq z_{ij} \quad \forall i, j = 1, 2, \dots, T \quad (6)$$

$$C_j \geq C_i + p_{jk} - E(1 - y_{ijk}) \quad \forall j = 1, 2, \dots, T; \forall i; \forall k \quad (7)$$

$$C_0 = 0 \quad (8)$$

$$y_{ijk} \in \{0, 1\} \quad \forall i, j, k \quad (9)$$

$$C_i \geq 0 \quad \forall i \quad (10)$$

$$C_{\max} \geq 0 \quad (11)$$

The objective function to minimize is stated in Constraint (1) and the Makespan is obtained in Constraints (2). A dummy task is introduced in Constraints (3), which indicate that the dummy task 0 is placed at the beginning of each machine. Constraints (4) ensure that each task has only one immediately previous task, only one immediately subsequent task and is assigned to only one machine, except task 0, the dummy task. Balance equations are included through Constraints (5). Constraints (6) are enforced to guarantee that the mandatory sequences between certain pairs of tasks are fulfilled. Constraints (7) and (8) define the completion

time of all tasks. Finally, Constraints (9), (10) and (11) are the basic restrictions on the decision variables.

In assessing the scale of the mathematical model under consideration, the order of magnitude of the number of constraints and variables deserves attention. The number of constraints can be expressed as $T^2 \cdot M + T^2 + 4 \cdot T \cdot M + 3 \cdot T + 2 \cdot M + 3$, while the number of variables can be given by $T^2 \cdot M + 2 \cdot T \cdot M + T + M + 2$. This implies a polynomial growth pattern where the number of constraints in the model has a complexity of $\mathcal{O}(T^2 \cdot M)$ and the number of variables has a complexity of $\mathcal{O}(T^2 \cdot M)$. Such a characterisation illustrates the increase in constraints and variables concerning the dimensions T and M within the model.

3.2 Discrete model

In this section we present the Mixed Linear Programming model in its discrete time representation. As mentioned earlier, in these models the partitioning of the time horizon is a task that takes place before and outside of the optimisation process. This implies constraints on the times at which each job can start (Burkard and Hatzl 2005), as the points on the horizon where the start of each job can be planned are fixed and no intermediate points can be chosen.

In this case, for this discretisation, a practical upper bound (W_{\max}) is used to determine the length of the time horizon. In addition, a time period (U) is used to define the size of each time interval within the horizon, $[U - 1, U]$. In this research, U corresponds to a time interval of 30 min.

Next, we present the parameters, index and variables used exclusively for this discrete formulation (Table 3). We then present the MILP model in its discrete formulation, and finally, an explanation of each of the constraints.

$$\text{Min} \quad C_{\max} \tag{12}$$

$$\text{s.a.} \quad C_{\max} \geq C_i \quad \forall i \tag{13}$$

$$\sum_{k=1}^M \sum_{u \in W_{ik}} y_{iuk} = 1 \quad \forall i \tag{14}$$

$$\sum_{i=1}^T \sum_{v=u-p_{ik}+1}^u y_{ivk} \leq 1 \quad \forall k; \forall u \in \{1, \dots, W_{\max}\} \tag{15}$$

$$y_{(i+1)(u+p_{ik})k} \geq y_{iuk} - E(1 - z_{i(i+1)}) \quad \forall i \in \{1, \dots, T - 1\}, \forall k, \forall u; \tag{16}$$

$$C_i = \left[\sum_{k=1}^M \sum_{u \in W_{ik}} y_{iuk} \cdot (u + p_{ik}) \right] - 1 \quad \forall i \quad (17)$$

$$y_{ijk} \in \{0, 1\} \quad \forall i, j, k \quad (18)$$

$$C_i \geq 0 \quad \forall i \quad (19)$$

$$C_{\max} \geq 0 \quad (20)$$

The objective function to minimize is stated in Constraint (12) and the Makespan is obtained in Constraints (13). Constraints (14) require each task to be started exactly once and Constraints (15) ensure that at a given time period u , only one task can be executed on each machine. Constraints (16) are enforced to guarantee that the mandatory sequences between certain pairs of tasks are fulfilled. Constraints (17) define the completion time of all tasks and Constraints (18), (19) and (20) are the basic restrictions on the decision variables.

When assessing the scale of the mathematical model under consideration, the order of magnitude of the number of constraints and variables needs to be examined. The number of constraints is expressed as $T \cdot M \cdot W_{\max} + T^2 \cdot M + 4 \cdot T + 1$, while the number of variables is given by $T \cdot M \cdot W_{\max} + T + 1$. This implies a polynomial growth pattern, where the number of constraints has a complexity of $\mathcal{O}(T \cdot M \cdot W_{\max})$, since W_{\max} is necessarily greater than T , and the number of variables has a complexity of $\mathcal{O}(T \cdot M \cdot W_{\max})$. Such a characterisation illustrates the increasing constraints and variables concerning the dimensions T , M and W_{\max} within this model.

4 Results and conclusions

As mentioned above, the problem described is based on a real production planning problem in a metallurgical plant. This has allowed the use of time estimations derived from the actual historical data of the factory. In addition to the time estimations, the priority relationships are also based on a real project, and the different configurations represent situations of interest to this industry.

These configurations are characterised by the number of machines available for task processing, shown in the column M , and the number of tasks to be scheduled, shown in the column T . Configurations with 2, 4 and 6 machines were chosen, and for each of these 20, 40, 80 and 160 tasks were considered, giving a total of 12 configurations. All were run for both models, with the same processing times for each task and the same priority relationships for each configuration.

A time limit of 2 h was also implemented as a time constraint for model execution. When this limit was reached, the model stopped and returned the best solution achieved up to that point, together with the corresponding makespan. In cases where this measure was implemented, the column *Runtime* in Table 4 contains the

Table 4 Comparison of Makespans (hours) and runtimes (seconds) obtained for the 12 configurations in the continuous and discrete models. LR: limit reached (2 h/7200 s)

M	T	Continuous model		Discrete model	
		Makespan (h)	Runtime (s)	Makespan (h)	Runtime (s)
2	20	47.3897	0.1359	49.5000	3.8118
	40	98.6939	0.6057	104.5000	69.3254
	80	190.5366	26.2331	204.0000	LR
	160	375.4440	LR	443.0000	LR
4	20	40.6574	0.1659	42.0000	2.6155
	40	48.0537	0.6876	51.5000	52.6859
	80	87.5284	4.4235	91.0000	1686.0407
	160	173.8521	LR	202.0000	LR
6	20	31.7948	0.2019	34.5000	3.8138
	40	40.6575	0.8925	42.0000	54.7838
	80	62.2815	9.0968	69.0000	1070.8293
	160	99.4577	LR	106.0000	5148.3380

abbreviation *LR* (limit reached). In order to facilitate the interpretation of the results, for each of the twelve configurations, the best result in terms of makespan, i.e. the lowest maximum completion time between both models, has been highlighted in bold.

Both models have been implemented in Python and solved on an Intel® Core™ i7-4790 CPU at 3.60 GHz with 12 GB of RAM using Gurobi Optimizer v9.1.1.

These results are analysed in more detail in the following, with the aim of responding to as wide a range of situations as possible.

Strictly considering the makespan, which is the only objective in both objective functions (Constraints 1 and 12), it is evident that the continuous model gives better results in all cases. In fact, the results show that the makespan of the discrete model is higher than that of the continuous model, ranging from 3.2% ($M = 4, T = 20$ and $M = 6, T = 40$) to 18% ($M = 2, T = 160$). This result is consistent with the implications of the time discretisation itself, because even if both models were to choose exactly the same scheduling, it is highly likely that the discrete model would result in a higher makespan. This is due to the losses incurred in assigning tasks with real processing times, which may not be multiples of U , to time intervals of duration U . In most cases, this will result in idle times in the last interval in which each task is assigned in the discrete model. To illustrate this concept, consider a simple example

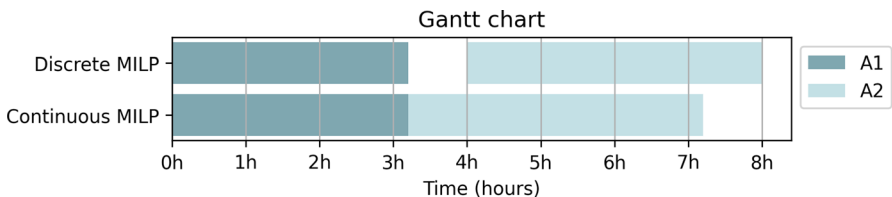


Fig. 1 Gantt chart of the continuous and discrete models for two identical tasks (*A1* and *A2*) following the same sequence

with only two tasks, $A1$ and $A2$. Assuming that the time discretization for the discrete model is one hour ($U = 1$ h), if tasks $A1$ and $A2$ have durations of 3.2 h and 4 h respectively, the discrete model would give a makespan of 8 h compared to the continuous model's 7.2 h (Fig. 1), despite returning the same actual schedule.

This phenomenon occurs to a minor extent if the size of the individual time intervals (U) is reduced, because the unallocated times generated will have a shorter maximum duration. However, using smaller U intervals will also result in longer execution times. Furthermore, these unallocated times will always be present, and with a greater number of tasks, the probability of generating unallocated times increases. This can be clearly seen in the results: as the number of tasks increases, the difference between the makespans of the continuous and discrete models increases significantly (Table 4).

In order to better study the influence of the number of tasks and machines on the difference between the results of the two models, Table 5 is presented. In this table, the first two columns again give the number of machines (M) and the number of tasks (T) for each configuration run. The third column (*Makespan difference*) shows the difference between the makespans, with the continuous model consistently giving smaller makespans. The fourth column (*Runtime difference*) shows the difference between the runtimes obtained, where *LR* indicates when one of the models reached the time limit. Finally, the fifth column (*Best Runtime Performance Model*) indicates which model delivered the better runtime performance. In this last column, a dash (-) appears when both models reached the time limit. In cases where only one model reached the time limit, *LR* is displayed in the *Runtime Difference* column, and the *Best Runtime Performance Model* column indicates the model that did not reach the time limit and therefore provided the best runtime.

With these results in mind, it is easy to see that the difference between the makespans of the two models does indeed grow as the number of tasks increases. Furthermore, when we increase the number of available machines, this growth

Table 5 Difference between the performance of the two models in terms of makespan and execution times

M	T	Makespan difference	Runtime difference	Best runtime performance model
2	20	2.11	3.68	Continuous
	40	5.81	68.72	Continuous
	80	13.46	LR	Continuous
	160	67.56	LR	-
4	20	1.34	2.45	Continuous
	40	3.45	52.00	Continuous
	80	3.47	1681.62	Continuous
	160	28.15	LR	-
6	20	2.71	3.61	Continuous
	40	1.34	53.89	Continuous
	80	6.72	1061.73	Continuous
	160	6.54	LR	Discrete

is significantly mitigated. This confirms that the problem of time discretization becomes more pronounced as more tasks are assigned to the same machine, leading to an accumulation of unallocated times.

It is important to note that the solution returned by the discrete model does not have a higher makespan solely due to unallocated times. Additionally, the assignment and sequencing of tasks does not always coincide with that of the continuous model. Therefore, if the term *net makespan* refers to the makespan resulting from removing unallocated times from the schedule returned by the discrete model, the net makespan of the discrete model is equal to or higher than that of the continuous model in all configurations studied. This is also due to the loss of accuracy caused by the discretisation of time.

Finally, regarding the makespan, it's worth noting that the continuous model reaches the time limit for the first and only time with 6 machines and 160 tasks. However, its makespan is still lower than that of the discrete model, which does not reach the time limit. This again emphasises the loss of efficiency in the discrete model due to the inherent discretisation of the time horizon.

On the other hand, if we now turn our attention to the execution times, the continuous model again delivers better results than the discrete model in all but one configuration. Furthermore, if we look at the runtimes in Table 4, we can see that the model does not take more than half a minute in any of the configurations with 20, 40 and 60 tasks. However, when the number of tasks increases to 160, the time limit of 2 h is reached in all cases. On the other hand, the discrete model reaches the time limit in problems with fewer machines and does not do so as the number of machines increases.

In conclusion, for environments with these characteristics, the use of continuous models is clearly more advantageous, as they provide solutions with better makespans and runtimes. However, an interesting line of future research would be to extend both mathematical models to other industries where the number of tasks is higher, as the performance of the discrete model suggests that it could give better results in such environments compared to the continuous model.

Acknowledgements This research was funded by the Agency for Innovation and Development of Andalusia (IDEA), by means of "Open, Singular and Strategic Innovation Leadership" Programme, through the joint innovation unit project OFFSHOREWIND (802C200003).

Funding Funding for open access publishing: Universidad de Sevilla/CBUA.

Declarations

Conflict of interest The authors declare no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission

directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aguirre AM, Papageorgiou LG (2018) Medium-term optimization-based approach for the integration of production planning, scheduling and maintenance. *Comput Chem Eng* 116:191–211
- Berber R, Yuceer M, Ozdemir Z (2007) Automatic generation of production scheduling models in single stage multi-product batch plants: some examples. *Math Comput Model* 46:69–79
- Brucker P (2007) *Scheduling algorithms*. Springer, Berlin
- Burkard RE, Hatzl J (2005) Review, extensions and computational comparison of MILP formulations for scheduling of batch processes. *Comput Chem Eng* 29:1752–1769
- Chen P, Papageorgiou LG, Pinto JM (2008) Medium-term planning of single-stage single-unit multiproduct plants using a hybrid discrete/continuous-time MILP model. *Ind Eng Chem Res* 47:1925–1934
- Díaz-Ramírez J, Huertas JI (2018) A continuous time model for a short-term multiproduct batch process scheduling. *Ing Investig* 38:96–104
- Elekidis AP, Corominas F, Georgiadis MC (2019) Production scheduling of consumer goods industries. *Ind Eng Chem Res* 58:23261–23275
- Floudas CA, Lin X (2004) Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng* 28(11):2109–2129
- Floudas CA, Lin X (2005) Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Ann Oper Res* 139:131–162
- Graham RL, Lawler EL, Lenstra JK, Kan AHGR (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann Discrete Math* 5:287–326
- Harjunkoski I, Grossmann IE (2002) Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Comput Chem Eng* 26:1533–1552
- He Y, Liang Y, Liu Z, Hui CW (2017) Improved exact and meta-heuristic methods for minimizing makespan of large-size SMSP. *Chem Eng Sci* 158:359–369
- Lee KH, Heo SK, Lee HK, Lee IB (2002) Scheduling of single-stage and continuous processes on parallel lines with intermediate due dates. *Ind Eng Chem Res* 41:58–66
- Liu S, Pinto JM, Papageorgiou LG (2008) A TSP-based MILP model for medium-term planning of single-stage continuous multiproduct plants. *Ind Eng Chem Res* 47:7733–7743
- Liu S, Pinto JM, Papageorgiou LG (2010) MILP-based approaches for medium-term planning of single-stage continuous multiproduct plants with parallel units. *CMS* 7:407–435
- Liu S, Pinto JM, Papageorgiou LG (2010) Single-stage scheduling of multiproduct batch plants: an edible-oil deodorizer case study. *Ind Eng Chem Res* 49:8657–8669
- Lorenzo-Espejo A, Escudero-Santana A, Muñoz-Díaz M-L, Robles-Velasco A (2022) Machine learning-based analysis of a wind turbine manufacturing operation: a case study. *Sustainability* 14:7779
- Marchetti PA, Cerdá J (2009a) An approximate mathematical framework for resource-constrained multistage batch scheduling. *Chem Eng Sci* 64:2733–2748
- Marchetti PA, Cerdá J (2009b) A continuous-time tightened formulation for single-stage batch scheduling with sequence-dependent changeovers. *Ind Eng Chem Res* 48:483–498
- Merchan AF, Maravelias CT (2014) Reformulations of mixed-integer programming continuous-time models for chemical production scheduling. *Ind Eng Chem Res* 53:10155–10165
- Méndez CA, Cerdá J (2002) An MILP framework for short-term scheduling of single-stage batch plants with limited discrete resources. *Comput Aided Chem Eng* 10:721–726
- Méndez CA, Cerdá J (2003) *Dynamic scheduling in multiproduct batch plants*, vol 27. Elsevier Ltd, Amsterdam, pp 1247–1259
- Mouret S, Grossmann IE, Pectiaux P (2011) Time representations and mathematical models for process scheduling problems. *Comput Chem Eng* 35:1038–1063
- Muñoz-Díaz M-L, Escudero-Santana A, Lorenzo-Espejo A (2024) Solving an Unrelated Parallel Machines Scheduling Problem with machine- and job-dependent setups and precedence constraints considering Support Machines. *Comput Oper Res* 163:106511

- Muñoz-Díaz M-L, Escudero-Santana A, Lorenzo-Espejo A, Robles-Velasco A (2022) Modelos lineales mixtos para la programación de la producción con una sola etapa: estado del arte. *Dir Org* 77:63–73
- Pan M, Li X, Qian Y (2009) Continuous-time approaches for short-term scheduling of network batch processes: small-scale and medium-scale problems. *Chem Eng Res Des* 87:1037–1058
- Pinto JM, Grossmann IE (1995) A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Ind Eng Chem Res* 34:3037–3051
- Roslöf J, Harjunkoski I, Björkqvist J, Karlsson S, Westerlund T (2001) An MILP-based reordering algorithm for complex industrial scheduling and rescheduling. *Comput Chem Eng* 25:821–828
- Shaik MA, Janak SL, Floudas CA (2006) Slot-based vs. global event-based vs. unit-specific event-based models in scheduling of batch plants. *Comput Aided Chem Eng* 21:1923–1928
- Sun HL, Xue YF (2009) An MILP formulation for optimal scheduling of multi-product batch plant with a heuristic approach. *Int J Adv Manuf Technol* 43:779–784
- Sung C, Maravelias CT (2009) A projection-based method for production planning of multiproduct facilities. *AIChE J* 55:2614–2630

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.