

An integrated personnel allocation and machine scheduling problem for industrial size multipurpose plants

Fernando Santos* Ricardo Fukasawa*
Luis Ricardez-Sandoval*

* University of Waterloo, Waterloo, Canada
e-mails: {fasantos, rfukasawa, laricardezsandoval}@uwaterloo.ca

Abstract: This paper describes the development and implementation of an optimization model to solve the integrated problem of personnel allocation and machine scheduling for industrial size multipurpose plants. Although each of these problems has been extensively studied separately, works that study an integrated approach are very limited, particularly for large-scale industrial applications. We present a mathematical formulation for the integrated problem and show the results obtained from solving large size instances from an analytical services facility. The integrated formulation can improve the results up to 22.1% compared to the case where the personnel allocation and the machine scheduling problems are solved sequentially.

Keywords: Scheduling, Personnel Allocation, Analytical Services Sector, Optimization

1. INTRODUCTION

The machine scheduling problem, also known as scheduling operations problem or short-term scheduling, is one of the most relevant optimization problems and arises in several industrial and engineering fields such as chemical production, mining, computers and manufacturing plants (Abedinnia et al., 2017; Blazewicz et al., 1994; Schulze and Zimmermann, 2017; Patil et al., 2015). In general, this problem involves finding the assignment of resources to execute tasks in a given time horizon aiming to optimize a goal, such as minimizing costs, turnaround time (makespan) or maximizing throughput.

The complexity of machine scheduling problems varies according to the class associated to the problem (Blazewicz et al., 2007). Although some scheduling problems can be solved in polynomial time, the most realistic problems consider operational constraints that increase their complexity and fall in the NP-Hard class. The machine scheduling problem that is considered in this study belongs to the NP-Hard class since jobs are non-preemptive and have precedence constraints (Lenstra and Kan, 1978).

The scheduling problem associated to this work arises from an analytical services facility that receives thousands of samples from different clients every week. The large scale feature associated to this problem increases its complexity and requires efficient methods to solve the problem within a reasonable amount of time. This problem was first introduced by Patil et al. (2015) who proposed a solution using multicommodity flow and integer linear programming based on a discrete time representation framework.

Follow-up studies conducted by Lagzi et al. (2017a,b) focused on different formulations for the machine scheduling

problem, in particular different time representation frameworks. Those studies show that, for the problem under consideration non-uniform discrete time representation is the better choice. One of the drawbacks of their approach, and other traditional machine scheduling problems, is the assumption that processing time of machines is constant for identical tasks, which is not necessarily the case for many applications. For instance, this happens when machines depend on the operator skills to execute their tasks. This is typical in the analytical services sector, where tasks involve preparation and setup of machines and samples. Therefore, a more skilled operator may perform this activity in shorter processing times and better overall solutions may be achieved if the machine scheduling and the employee allocation decisions are performed simultaneously.

The personnel allocation problem, also known as employee timetabling or staff scheduling, has also received attention from the literature (Ernst et al., 2004). The problem consists of finding work timetables for an organization staff so that it can satisfy the demand for goods or services. Different types of objectives and constraints can be considered, including employees satisfaction, regulations and costs. The complexity of the problem depends on the constraints to be met; however, in most real world applications, problems are NP-Hard (Brucker et al., 2011).

These two important optimization problems - Machine Scheduling and Personnel Allocation - come together when one needs to assign employees to machines in order to process tasks of a given scheduling problem. In particular, this problem becomes more challenging when machines' processing times are given in terms of the employees' skills. A naive solution approach is to tackle the problem sequentially: first to solve the personnel allocation problem and then proceed with the solution for the machine scheduling problem. However, this approach may lead to suboptimal solutions, since optimal solutions for the first problem do not necessarily provide the best setting to solve the second.

* The financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Ontario Centers for Excellence (OCE), and the industrial partner in the analytical services sector are gratefully acknowledged.

The integrated personnel allocation and machine scheduling problem has also received attention from the literature. One of the first attempts to solve the problem was proposed by Daniels and Mazzola (1994) followed by Daniels et al. (2004), where they model employees as resources required by the machines and solve the machine scheduling problem with this additional constraint. The approach proposed by Artigues et al. (2009) combine methods based on integer linear programming and constraint programming to solve an integrated problem that considers regulations in the employees' schedule. More recently Guyon et al. (2014) introduce a cut generation framework based on Bender's decomposition to evaluate optimal solutions for the problem.

Despite the number of works related to the integrated problem, there is still a lack of works addressing large scale industrial applications. One of the few works solving the problem for industrial size problems is from Schulze and Zimmermann (2017), which proposes a mathematical formulation to solve small size instances and a heuristic algorithm to obtain feasible solutions for large size instances. Similarly, Ahmadi-Javid and Hooshangi-Tabrizi (2017) solved an industrial application of the integrated personnel allocation and machine scheduling problem using a heuristic algorithm, which is tuned using a mathematical formulation that solves only small size instances.

The above works differ from this study from two perspectives: (i) they solve different scheduling problems from the one considered here; (ii) both implement heuristic algorithms to solve large size instances. No works were found in the literature focusing on efficient exact methods for the integrated problem applied to large-scale industrial applications. In this context, the proposed work improves on the literature of the integrated machine scheduling and personnel allocation introducing an efficient mathematical formulation to solve a real application from an analytical service industry.

2. PROBLEM DEFINITION

Let P be the set of processing units available in the analytical services facility. Each processing unit $i \in P$ consists of a set of identical machines P_i that perform a specific process. The set of all machines in the plant is given by $M = \cup_{i \in P} P_i$. The facility receives a set of jobs J to be processed within a scheduling horizon $H > 0$. Each job $j \in J$ must be processed along a sequence of n_j processing units $K_j = (P_1^j, P_2^j, \dots, P_{n_j}^j)$, also called a path.

Values a_{jk} define how many samples from job j should start processing at the k^{th} processing unit of path K_j . Once a sample starts processing at the k^{th} processing unit, it needs to visit all of the subsequent processing units of path K_j . These values may represent, for instance, samples for a particular job that have already gone through previous processing units at the beginning of the current scheduling horizon.

Sync is not required between samples of the same job, which means that given x and y distinct samples from job j , it is possible that different processing units P_i and P_k process respectively x and y at the same time, as long as

they visit processing units from path K_j in the sequence specified.

Each processing unit P_i has a specific capacity β_i , which means that each machine $m \in P_i$ can be loaded with at most β_i samples from potentially different jobs. Although machines from the same processing unit are assumed to be identical, they can have different processing times depending on which employee is operating the machine. Let S be a discrete number of skills employees can operate machines and $s = 0, 1, \dots, S$, where 0 represents no operation and S represents the maximum skill level. We define τ_{ms} as the processing time of machine m under skill s . Also, E is the number of employees available for scheduling in the facility such that $e = 1, \dots, E$ represent each employee.

Non-preemption is assumed, which means that once machine m operated by employee e is assigned to process samples, it will run without interruption for τ_{ms} units of time before releasing the samples. Also, there is no minimum working capacity for any machine, i.e, machines can be turned on with any number of samples between 0 and β_i and no transportation time or storage capacity between processing units is considered. It is assumed that all machines in the facility are available to start processing materials at the beginning of the scheduling horizon.

Feasible solutions for this problem consist of an integrated schedule in the analytical services facility such that each employee is associated to exactly one machine and no more than one employee operates a machine. Also, samples of each job $j \in J$ must visit processing units in the order given by path K_j such that each sample is processed by at most one machine at a time and the total sum of samples in a given machine from processing unit P_i does not exceed β_i . The most common objective function for scheduling problems is to minimize makespan, which is the time spent processing all jobs in the facility. However, this objective cannot be explicitly implemented in this study because it considers that some jobs may not be able to finish processing within the horizon. Accordingly, we set our objective function to maximize a weighted samples throughput. The weight gives higher priority to samples in the last stage of their paths, which also increases the jobs throughput within the horizon H .

3. MATHEMATICAL FORMULATION

In this section we first introduce a mathematical formulation for the personnel allocation problem, then we show how the employees allocation decisions are considered together with the machine scheduling problem in an integrated formulation. In the present case, we assume that horizon H is no longer than the employees' shifts which allows us to assign employees to machines for the entire horizon.

3.1 The Personnel Allocation Problem

The problem here consists of assigning employees to machines respecting the employees' skills for each machine in order to minimize the total processing time. The formulation requires as input two matrices, i.e.:

$$\mathcal{S} = \begin{pmatrix} s_{11} & \dots & s_{1E} \\ \vdots & \ddots & \vdots \\ s_{|P|1} & \dots & s_{|P|E} \end{pmatrix} \quad \tau = \begin{pmatrix} \tau_{11} & \dots & \tau_{1S} \\ \vdots & \ddots & \vdots \\ \tau_{|P|1} & \dots & \tau_{|P|S} \end{pmatrix}$$

\mathcal{S} is the skills matrix and stores skills values s associated to each processing unit p and employee e , while matrix τ stores the processing times of each processing unit p for the skills classes $s > 0$. Note that entries in matrices \mathcal{S} and τ are in terms of processing units $p \in P$ but they can be easily mapped to machines $m \in M$ since in a given processing unit all machines are assumed to be identical.

For the present formulation we introduce a parameter to control whether employee e is able or not to operate machine m . For a given processing unit $i \in P$ we consider all machines $m \in P_i$ and assign values in terms of entries of matrix \mathcal{S} as follows:

$$\sigma_{me} = \begin{cases} 0; & \text{if } s_{ie} = 0 \\ 1; & \text{otherwise} \end{cases}$$

Also, to increase readability of the formulation we use τ_{me} to represent the processing time of machine m according to the skills employee e operates that machine, instead of $\tau_{mS_{me}}$ as defined in section 2. The mathematical formulation makes use of binary decision variables x_{me} to represent whether or not employee e is assigned to work on machine m during the scheduling horizon. The formulation is described in (1)-(5)

$$\min \sum_{m \in M} \sum_{e=1}^E x_{me} \tau_{me} \quad (1)$$

$$\sum_{e=1}^E x_{me} \leq 1 \quad \forall m \in M \quad (2)$$

$$\sum_{m \in M} \sigma_{me} x_{me} = 1 \quad \forall e = 1, \dots, E \quad (3)$$

$$\sum_{e=1}^E \sum_{m \in P_i} \sigma_{me} x_{me} \geq 1 \quad \forall i \in P \quad (4)$$

$$x_{me} \in \{0, 1\} \quad \forall m \in M, e = 1, \dots, E \quad (5)$$

Objective function (1) minimizes the overall processing time of machines assigned to employees. Constraints (2) enforce that no more than one employee is assigned to a machine whereas constraints (3) guarantee that each employee is assigned to exactly one machine. Furthermore, constraints (4) ensure that at least one machine of each processing unit is assigned to an employee and constraints (5) define the decision variables space.

Although formulation (1)-(5) can be efficiently solved, optimal solutions for the personnel allocation can lead to suboptimal allocations for the integrated problem with machine scheduling. For this reason, we introduce a mathematical formulation for the integrated personnel allocation and machine scheduling problem.

3.2 The Integrated Machine Scheduling and Personnel Allocation Problem

One of the main questions on solving machine scheduling problems is regarding time representation in the scheduling formulation. The methods and formulations rely on either continuous or discrete time representation of events. In the continuous time, events are allowed to happen any time in the system, which provides more flexibility but has higher computational cost. As an alternative, discrete time representation consists of using a pre-determined number of time points to control the events, which leads to a trade-off between solution quality and computational cost: the more time points the better the quality of solutions but the higher the computational costs. Discrete time representation can be uniform or non-uniform. The formulation proposed here extends the non-uniform discrete time formulation proposed by (Lagzi et al., 2017b) to include personnel allocation decisions.

The time representation makes use of steps $\Delta_p > 0$ to represent the time elapsed between two consecutive time points for processing unit $p \in P$. Let $\theta_p = (0, \epsilon_{p1}, \epsilon_{p2}, \dots, H)$ be the increasing sequence of time points and $t_p = 0, 1, \dots, \mathcal{T}_p$ be the time points indices associated to θ_p , such that $\Delta(p) = \epsilon_{pt} - \epsilon_{p(t-1)} : 1 \leq t \leq \mathcal{T}_p$. Note that, Δ_p is known a priori as a parameter of the problem. Also note that, the formulation can be readily adapted to deal with more general discretizations where consecutive time intervals can have different lengths.

The formulation comprises four groups of decision variables. Variables $x_{jkst} \in \mathbb{Z}^+$ control the number of samples from job $j \in J$ that start processing in the k^{th} processing unit of the path which is operated by an employee with skills s at time index t . We recall that in path K_j , P_1^j is not necessarily the processing unit P_1 but the first processing unit that needs to be visited by job j , P_2^j is the second and so on. It is also noteworthy that, processing unit $i \in P$ has a set of P_i identical machines performing the same process but having possibly different processing times due to different employees skills assigned to these machines.

Decision variables $y_{pst} \in \mathbb{Z}^+$ represent the number of machines from processing unit $p \in P$ operated by employees with skills s to process samples at time index t . w_{jkt} are the variables modelling the number of samples from job j that are available to start processing in the k^{th} processing unit at time index t . These variables are introduced to control the samples' flow balance in the system. In addition, decision variables $z_{pe} \in \{0, 1\}$ represent whether employee e is assigned or not to operate processing unit p . These variables are responsible for the personnel allocation control in the machine scheduling model.

Some additional notation is required before introducing the formulation. For the k^{th} processing unit of job j at time index t operated using skills set s , we define

$$\gamma(P_k^j, t, s) = \{t' : \epsilon_{P_k^j(t-1)} < \epsilon_{P_{k-1}^j t'} + \tau_{P_{k-1}^j s} \leq \epsilon_{P_k^j t}\}$$

as the set of time point indexes associated to the preceding processing unit of P_k^j able to finish processing samples between time points $\epsilon_{P_k^j(t-1)}$ and $\epsilon_{P_k^j t}$. Similarly, the set

$$\phi(P_k^j, t, s) = \{t' : \epsilon_{P_k^j t} < \epsilon_{P_k^j t'} \leq \epsilon_{P_k^j t} + \tau_{P_k^j s}\}$$

comprises all indices associated to time points of processing unit P_k^j able to finish processing between $\epsilon_{P_k^j t}$ and $\epsilon_{P_k^j t} + \tau_{P_k^j s}$. Parameter σ_{pe} from formulation (1)-(5) needs an extra index to determine whether employee e operates processing unit p with skills s or not and is redefined as

$$\sigma_{pes} = \begin{cases} 1; & \text{if } s_{pe} = s \\ 0; & \text{otherwise.} \end{cases}$$

The formulation for the integrated personnel allocation and machine scheduling is given in (6)-(15).

$$\max \sum_{j \in J} \sum_{k=1}^{n_j} \frac{k}{n_j} \sum_{s=1}^S \sum_{t=1}^{t_{P_k^j}} x_{jkst} \quad (6)$$

$$\sum_{j \in J} \sum_{k=1}^{n_j} \sum_{s=1}^S x_{jkst} = 0 \quad (7)$$

$$w_{jk0} = a_{jk}; \quad \forall j \in J, \forall k = 1 \dots n_j \quad (8)$$

$$\sum_{s=1}^S x_{j1st} + w_{j1t} = w_{j1(t-1)} \quad \forall j \in J, t = 1, \dots, t_{P_1^j} \quad (9)$$

$$\sum_{s=1}^S x_{jkst} + w_{jkt} = w_{jk(t-1)} + \sum_{s=1}^S \sum_{t' \in \gamma(P_k^j, t, s)} x_{j(k-1)st'} \quad (10)$$

$\forall j \in J, k = 2, \dots, n_j, t = 1, \dots, t_{P_k^j}$

$$\sum_{j, k: p=P_k^j} x_{jkst} \leq \beta_p y_{pst} \quad \forall p \in P, s = 1, \dots, S, t = 1, \dots, t_p \quad (11)$$

$$\sum_{s=1}^S \sum_{t' \in \phi(P_k^j, t, s)} y_{pst'} \leq \sum_{s=1}^S \sum_{e=1}^E s_{pes} z_{pe} \quad \forall p \in P, t = 1, \dots, t_p \quad (12)$$

$$y_{pst} \leq \sum_{e=1}^E s_{pes} z_{pe} \quad \forall p \in P, s = 1, \dots, S, t = 1, \dots, t_p \quad (13)$$

$$\sum_{p \in P} z_{pe} \leq 1 \quad \forall e = 1, \dots, E \quad (14)$$

$$\sum_{e=1}^E z_{pe} \leq |P_p| \quad \forall p \in P \quad (15)$$

Objective function (6) aims to maximize samples throughput. It includes a weight $\frac{k}{n_j}$ to increase samples priority according to their order in the jobs' path, which gives higher weights to samples in the last processing units of the path and helps improving jobs throughput. Constraints (7) and (8) are for initialization purposes and complement each

other defining decision variables values at the beginning of the time horizon. More specifically, constraints (7) ensure that no samples start processing at time point 0, while constraints (8) state that all samples become available in the facility at that time.

The samples' flow balance is enforced by constraints (9) and (10). They guarantee that, at each time point, the sum of samples waiting, processing and already processed is the same for each job and processing unit. Constraints (9) are in charge of the flow balance of the first machine of the path for each job, while constraints (10) ensure the flow balance for the remaining processing units of the path. Constraints (11) are the knapsack-style constraints to enforce capacity feasibility in the machines of each processing unit.

Coupling constraints (12) and (13) are included to integrate the machine scheduling and the personnel allocation problems. On the one hand, constraints (12) enforce that the number of active machines in a given processing unit p in between two consecutive time points $\epsilon_{p(t-1)}$ and ϵ_{pt} is no higher than the number of employees assigned to it. On the other hand, constraints (13) guarantee that machines operate under skills s only if there are proper employees assigned to the associated processing unit.

Moreover, constraints (14) and (15) are related to the personnel allocation problem and ensure respectively that one employee is assigned to at most one processing unit and the number of employees assigned to a given processing unit is no more than the number of machines it comprises.

4. COMPUTATIONAL EXPERIMENTS

This section presents results of formulations (1)-(5) and (6)-(15) solving the integrated personnel allocation and machine scheduling problem for an analytical services facility. We first provide details about the facility, then we show the results for instances built based on it. The identity of the industrial partner is not disclosed as per our non-disclosure agreement. However, the proposed computational study retains the characteristics of the majority of the operations at this facility.

We recall that, the following experiments cannot be compared to related works in the literature since they study applications with different sets of operational constraints. In particular, Schulze and Zimmermann (2017) solve an integrated scheduling problem that arises in the operation of a potash mine, while Ahmadi-Javid and Hooshangi-Tabrizi (2017) consider a ternary-integration scheduling problem that incorporates personnel allocation into the scheduling of machines and transporters.

4.1 The Analytical Services Facility Operations

The facility is composed by $|P| = 25$ processing units with a total of $|M| = 102$ machines. Each processing unit i has $1 \leq |P_i| \leq 10$ machines, which are assumed to be identical with capacity β_i . The performance of each machine of a given processing unit is determined by the skills' set in which they are operated. For this study, we define $S = 3$ which discretizes the skills' set into four groups $s = \{0, 1, 2, 3\}$, representing whether the employee

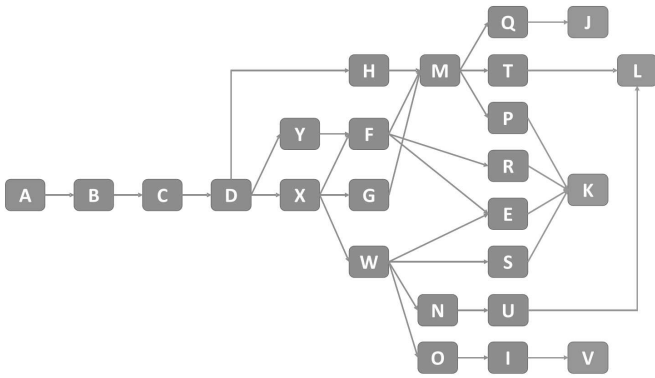


Fig. 1. Analytical Service Facility Processing Units and Paths (Lagzi et al., 2017b)

associated to skill s is not able to operate machines (0) or operates with junior (1), regular (2) or senior (3) skills. Since all machines of a processing unit are assumed to be identical, two different machines from the same processing unit $m_1, m_2 \in P_i$ have the same performance if operated under the same skill s , $\tau_{m_1 s} = \tau_{m_2 s}$.

The number of employees available to operate the machines may vary seasonally between 25 (the minimum to ensure the facility operation) and 102 (the total number of machines) in terms of the facility demands. We consider here 51 employees for each time horizon, since this number represents half of the facility capacity and is a good estimate between the lower and higher demand seasons.

Jobs to be processed in the facility have different paths according to the analysis they require. Figure 1 shows how the processing units are connected generating up to 20 different paths. Processing unit A is the starting point of all jobs and must be followed by B, C and D in this order. These are preprocessing activities for all jobs in the facility. J, K, L, V are terminal processing units while the remaining are intermediate processing units. We recall that, processing units are composed by up to 10 parallel machines that work in a multitasking environment.

Although the facility receives a different number of jobs for each scheduling horizon, we use here instances with 100, 150 and 200 jobs, which represent classes for low, medium and high demands. The number of samples a_{jk} is an integer value between 1 and 500 and is assigned based on real demands values. In the case job j has $a_{jk} > 0 : k > 1$ it means that a_{jk} samples of job j started processing in a previous horizon but did not finish visiting all processing units of path K_j . Therefore, these samples need to visit the remaining processing units in the subsequent horizons. This feature allows us to perform rolling horizon in the scheduling problem.

4.2 Results

We compare solutions for the integrated personnel allocation and machine scheduling using three different approaches. The first considers formulation (6)-(15), which evaluates optimal solutions for the integrated problem. The second, solves the problem separately: it first solves the personnel allocation using formulation (1)-(5) and maps the optimal decision variable values x_{me} into z_{pe}

allowing us to solve formulation (6)-(15) considering z_{pe} as constant values. Then, the resulting machine scheduling problem is solved to optimality. Hereinafter we name this approach as *sequential optimization* and the integrated approach as *simultaneous optimization*. The third approach is similar to the second but assign employees to machines randomly instead of solving formulation (1)-(5) and is named *random personnel allocation* approach. The models were implemented using *Julia* programming language and solved using *CPLEX 12.1* package with default parameters running on a machine with 264GB of RAM memory and 2.3GHz of speed using up to 4 simultaneous threads.

The results obtained by the three approaches are reported in Table 1. The first column shows the instance id, while columns 2 and 3 contain the number of samples processed during the scheduling horizon and the computational time (in seconds) spent to evaluate the optimal solution, labelled respectively ‘#smp’ and ‘t(s)’. The number of samples here refers to the sum of the amount of samples processed by all machines during the scheduling horizon. For example, assume job j with 4 processing units in its path ($n_j = 4$) and 100 units of samples are fully processed, its contribution to column ‘#smp’ is 400, since 100 samples were processed by 4 machines. We chose to report the number of samples instead of the objective function value here due to the noise the term $\frac{k}{n_j}$ introduces to interpret solution. Similar entries are given in columns 4 and 5 for the *sequential optimization* and columns 6 and 7 for the *random personnel allocation* approach. We report the results in blocks of 100, 150 and 200 jobs. For each block the average number of samples and computational time is reported for each approach.

As expected, the *simultaneous* and the *sequential* approaches outperform the *random personnel allocation* on solving the problem. We recall that, although the *random personnel allocation* approach solves the machine scheduling problem to optimality, it assigns employees to machines without any optimization criteria. This approach was included here as baseline to compare with the *sequential* and the *simultaneous* approaches.

When comparing the *simultaneous* and the *sequential* optimization approaches, the former clearly outperforms the latter from the solutions’ quality point of view but it also requires more computational efforts. Solutions for the *simultaneous* approach are able to process on average 21.2% more samples compared to the *sequential optimization* approach for those instances with 100 jobs. Similar improvements are obtained for instances with 150 and 200 jobs, with respectively 20.1% and 19.0%. However, the computational time also increases considerably when solving the *simultaneous* approach. From all instances, the maximum computational time spent by the *sequential optimization* approach to evaluate optimal solutions is 466 seconds, while the *simultaneous* requires up to 1739 seconds. This difference increases if more jobs are considered in the facility. On average, solving the *simultaneous* approach takes 672.9 seconds against 258.1 seconds from the *sequential*.

It is clear the trade-off involving solution quality and computational time when comparing the *simultaneous* and *sequential* approaches. The decision on the best approach

Table 1. Results of three different approaches for solving the integrated scheduling problem

instance	Simultaneous		Sequential		Random	
	#smp	t(s)	#smp	t(s)	#smp	t(s)
100J-1	34565	234.3	28442	105.6	21041	101.3
100J-2	33934	196.3	28345	94.9	16908	117.0
100J-3	34073	159.6	28475	98.2	17125	99.4
100J-4	34420	113.4	28496	73.9	22562	192.2
100J-5	33352	258.0	27586	117.0	21294	292.1
100J-6	29278	89.2	24263	80.8	16477	117.7
100J-7	35089	187.8	28854	107.9	21360	171.8
100J-8	35915	134.0	29787	105.7	23295	133.0
100J-9	30717	127.8	24961	93.1	18374	179.1
100J-10	32346	167.9	26700	89.9	20921	203.0
100J-11	31319	153.2	25199	107.5	19067	505.2
100J-12	32228	125.6	26439	85.1	16708	122.7
avg-100	33103	162.2	27295	96.6	19594	186.2
150J-1	40563	664.4	33627	269.6	25798	362.7
150J-2	36891	487.2	30942	230.9	19493	590.9
150J-3	39119	1101	33116	243.4	21895	599.0
150J-4	38671	702.2	31626	306.6	17050	417.3
150J-5	41431	1739	34098	287.4	20788	468.6
150J-6	43861	1049	36480	275.7	26046	499.8
150J-7	36886	383.7	30908	223.7	21300	371.4
150J-8	37834	558.7	31451	229.5	25788	475.4
150J-9	39050	468.3	32891	231.2	24412	350.3
150J-10	40710	525.8	34150	242.4	26507	420.1
150J-11	39823	716.2	32898	285.1	23699	339.7
150J-12	38331	519.8	31870	273.0	25111	261.4
avg-150	39430	748.0	32838	258.2	23157	429.7
200J-1	43590	797.2	37028	444.3	26062	592.6
200J-2	48533	1632	41225	462.6	26027	585.6
200J-3	42104	800.0	35834	465.4	24004	531.6
200J-4	42688	957.1	35111	426.0	23270	466.6
200J-5	44080	785.6	37317	333.5	27591	468.7
200J-6	46223	933.9	38941	353.3	28056	355.3
200J-7	45852	1076	38409	456.7	26712	1534
200J-8	47691	1275	40335	430.5	24617	364.8
200J-9	40840	978.0	33464	399.8	20379	460.7
200J-10	44557	1173	37187	415.9	28330	763.0
200J-11	45548	1471	38188	410.8	33083	1768
200J-12	45543	1419	38196	436.7	27618	785.0
avg-200	44770	1108	37602	419.6	26312	725.7

must be made according to the facility's requirements but in the case solution time is not critical, the *simultaneous* approach should be chosen since it improves upon the *sequential* solutions quality in more than 19% in average for the instances considered in this study.

5. CONCLUDING REMARKS

This study introduces a new mathematical formulation for the integrated personnel allocation and machine scheduling problem. Although these problems separately are widely studied in the literature, the number of works for the integrated version is still modest.

The formulation proposed here extends the non-uniform discrete formulation introduced by Lagzi et al. (2017b) and considers personnel allocation decisions integrated to the scheduling problem. In our approach, machines processing times vary according to operator skills, which makes the problem more challenging compared to the approach with constant processing times.

We evaluate our formulation using instances based on real world data from an analysis services facility and compare

the results with a sequential optimization approach, which consists of solving the personnel allocation problem first and using its solution as input to the machine scheduling problem. The results show that the integrated approach provides better overall solutions but is also more complex and requires more computational time to evaluate optimal solutions though its computational costs are not prohibitive to be actually used depending on the application.

REFERENCES

Abedinnia, H., Glock, C., Schneider, M., and Grosse, E. (2017). Machine scheduling problems in production: A tertiary study. *Comp. & Ind. Eng.*, 111, 403–416.

Ahmadi-Javid, A. and Hooshangi-Tabrizi, P. (2017). Integrating employee timetabling with scheduling of machines and transporters in a job-shop environment: A mathematical formulation and an anarchic society optimization algorithm. *Comp. Oper. Res.*, 84, 73–91.

Artigues, C., Gendreau, M., Rousseau, L., and Vergnaud, A. (2009). Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. *Comp. Oper. Res.*, 36, 2330–2340.

Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., and Weglarz, J. (2007). *Definition, Analysis and Classification of Scheduling Problems*, 57–72.

Blazewicz, J., Ecker, K., Schmidt, G., and Weglarz, J. (1994). *Scheduling in Computer and Manufacturing Systems*. Springer.

Brucker, P., Qu, R., and Burke, E. (2011). Personnel scheduling: Models and complexity. *Eur. J. Oper. Res.*, 210, 467–473.

Daniels, R., Mazzola, J., and Shi, D. (2004). Flow shop scheduling with partial resource flexibility. *Manag. Sci.*, 50, 658–669.

Daniels, R. and Mazzola, J. (1994). Flow shop scheduling with resource flexibility. *Oper. Res.*, 42, 504–522.

Ernst, A., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *Eur. J. Oper. Res.*, 153, 3–27.

Guyon, O., Lemaire, P., Pinson, E., and Rivreau, D. (2014). Solving an integrated job-shop problem with human resource constraints. *Annals Oper. Res.*, 23, 147–171.

Lagzi, S., Fukasawa, R., and Ricardez-Sandoval, L. (2017a). A multitasking continuous time formulation for short-term scheduling of operations in multipurpose plants. *Com. & Chem. Eng.*, 97, 135–146.

Lagzi, S., Lee, D.Y., Fukasawa, R., and Ricardez-Sandoval, L. (2017b). A computational study of continuous and discrete time formulations for a class of short-term scheduling problems for multipurpose plants. *Ind. Eng. Chem. Res.*, 56, 8940–8953.

Lenstra, J. and Kan, A. (1978). Complexity of scheduling under precedence constraints. *Oper. Res.*, 26, 22–35.

Patil, B., Fukasawa, R., and Ricardez-Sandoval, L. (2015). Scheduling of operations in a large-scale scientific services facility via multicommodity flow and an optimization-based algorithm. *Ind. Eng. Chem. Res.*, 54, 1628–1639.

Schulze, M. and Zimmermann, J. (2017). Staff and machine shift scheduling in a german potash mine. *Journal of Scheduling*, 1–22.