Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 237 (2024) 289–296

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry Sciences and Computer Science Innovation

A generalized disjunctive graph model for a complex production problem

Marta Flamini[a], Gaia Nicosia[b]

[a] *Università Telematica Internazionale UNINETTUNO, C.so Vittorio Emanuele II, 39, Rome 00186, Italy*
[b] *Dipartimento di Ingegneria Civile, Informatica e delle Tecnologie Aeronautiche, Università degli Studi "Roma Tre", Via della Vasca Navale 79, 00146 Rome, Italy*

Abstract

This work focuses on the modeling of the features of a real-world pharmaceutical production problem, where several limited resources must be gathered and scheduled in order to mix and process active substances. The problem can be modelled as a generalization of the job-shop scheduling problem, in which each job corresponds to an order. Besides the standard precedence and capacity constraints of a classical job-shop scheduling problem, release, no-wait and time interval production constraints are added. Furthermore, the processing of each job task must be assisted by an appropriate number of operators and performed by a specific machine, out of a set of feasible machines, that is temporarily installed in a compatible clean room. An innovative generalized disjunctive graph that models all the constraints and the resource conflicts is presented.

Peer-review under responsibility of the scientific committee of the iSCSi – International Conference on Industry Sciences and Computer Science Innovation

Keywords: Production Planning; Job-Shop Scheduling; Disjunctive Graph; Additional Resources.

1. Introduction

The problem addressed in this work concerns a resource-constrained job shop scheduling problem that arises in a real-world pharmaceutical plant. In resource-constrained scheduling problems the execution of the jobs requires, in addition to machines, the use of scarce extra resources [1]. The objective of this paper is to propose a theoretical model, namely, a generalized disjunctive graph model, to represent a production planning and scheduling problem in which, in addition to the standard constraints of a job-shop scheduling problem (precedence, capacity, no-preemption, blocking), some further complicating features have to be considered.

E-mail address: marta.flamini@uninettunouniversity.net

Peer-review under responsibility of the scientific committee of the iSCSi – International Conference on Industry Sciences and Computer Science Innovation
10.1016/j.procs.2024.05.107

The here considered problem arises in a production plant where micronization of active pharmaceutical ingredients and excipients is performed. Micronization is a technique used to decrease the particle size of the active ingredient in an excipient. In the pharmaceutical industry, powders are composed of a diverse group of solid particles and are often considered the most basic form of pharmaceuticals. They can be used as standalone formulations or serve as the basis for developing other dosage forms, such as tablets, capsules, or suspensions. In this context, a customer order specifying the quantity, delivery date, and release date of a particular component to be prepared corresponds to a job. Each job consists of a series of ordered operations that must be performed in the exact specified order. Each job (order) has a release date and a delivery date, and penalties are imposed for early or late delivery of the manufactured component. The more punctual the delivery of all jobs, the better the resulting production schedule will be.

One of the key features of the problem at hand is that the processing operations take place on machines located in specialized rooms. These rooms, usually called *clean rooms*, allow for the precise control and measurement of environmental factors such as temperature, humidity, pressure, air purity and composition, and the presence of contaminants or ionizing radiation, ensuring high repeatability of production conditions [2]. It is important to note that the configuration of machines within the clean room can change over time as needed. Each clean room can only accommodate one machine at a time, and not all machines or jobs can be accommodated in a given clean room. Moreover, each job is associated with a list of clean rooms that are capable of processing the job. Each job requires supervision by one or more human operators, who are identical and able to monitor each operation of each job. The operators are also capable of working in each clean room. Jobs can be executed during day or night shifts, but some jobs can only be processed during the day shift and cannot be processed at night. Additionally, some jobs have ”no-wait” operations, which are pairs of operations that must be executed consecutively without interruption. The output of the problem at hand is a job schedule that specifies the start and end times for all operations of all jobs. The objective is to minimize the total earliness and tardiness penalties with respect to the required delivery dates for each job.

The paper is organized as follows. First, we briefly report on some related results in the literature. Then, in Section 3, we rigorously describe the addressed scheduling problem, while Section 4 is focused on the generalized disjunctive graph model proposed. More precisely, different node and arc sets are defined, and a description of how a feasible solution can be derived from a specific selection of arcs is included. Finally, some conclusions are drawn.

2. Related Literature

The problem addressed in this paper is a job shop scheduling problem that involves two additional resources: operators and clean rooms. In resource-constrained scheduling problems, the execution of jobs requires not only machines, but also the use of scarce extra resources [3]. In the context of job shop problems with a single additional resource, the first problem considered in [1] was motivated by a manufacturing process in which human operators had to share the same set of tools. The authors proposed several exact and heuristic algorithms to solve the problem. Subsequently, this problem and other job shop scheduling problems with extra resources have been addressed by other researchers, who developed heuristic approaches, see, e.g. [4, 5].

The disjunctive graph is one of the most used models for describing instances of the job shop scheduling problem. This model was first proposed in [6] then, since the paper of [7], it has been widely used for solving several types of job shop scheduling problems. More recently, many authors have presented modified versions of the disjunctive graph to represent and help to address various types of job-shop problems (see for instance [1, 8, 9, 10]). In [1] the disjunctive graph is extended by considering extra sets of arcs to represent the assignment of operations to operators. In [8] is considered a generalized flexible job-shop scheduling problem arising from a real-world scenario observed in a seamless rolled ring manufacturing setting, which include constraints on machine capacity, lag times, holding times, and sequence-dependent setup times. The authors propose a metaheuristic algorithm based on a generalization of the classical disjunctive graph. In [9] the authors incorporate the scheduling of a single robot into the existing model, introduce transport operations as additional vertices in the disjunctive graph, representing the necessary transports, and specify that these operations must be carried out by the robot. In [10] a disjunctive graph model is modified to describe the relationship between transport and processing tasks in a a job shop problem in which mobile robots are utilized for the transport of work-in-progress.

3. Problem Statement

In this section we provide a detailed and formal description of the problem addressed in this paper.

We are given a set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$ each one composed by an ordered set of n_j operations $O(j) = \langle o_1^j, o_2^j, \ldots, o_{n_j}^j \rangle$. The processing time of the generic i-th operation of job J_j, o_i^j, is denoted by p_{ij}. The execution of each operation requests at least two resources, namely a machine and a clean room.

We are given μ machine types $MT = \{M_1, M_2, \ldots M_\mu\}$ and m_ℓ copies of machines of the same type $M_\ell \in MT$, with $m_\ell \geq 1$. Each operation o_i^j must be processed by a machine of a certain type $M(o_i^j) \in MT$.

Each operation must be executed within a designated clean room, requiring the installation of an appropriate machine prior to the processing of that operation. We are given r different clean rooms, namely $R = \{r_1, r_2, \ldots, r_r\}$. A machine can only be placed in a specific subset of R, in this case we say that the machine and the clean room are compatible. Consequently, based on the types of machines capable of processing the operations of a job, we also establish a compatibility relationship between a job and a specific subset of clean rooms. In addition to the machines and the clean rooms we are given another set of resources: that is a set of p identical human operators $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_p\}$. Each operation o_i^j requires, for all the processing time, the assistance of a given number of operators $\omega_i^j \in \{0, 1, \ldots p\}$. The time horizon of the problem is formed by T days, where for each day two shifts are considered: a day shift from (hour) a to (hour) b and the complementary night shift.

Each job J_j has associated a release date, which denotes the earliest possible start time for the job's processing. Additionally, each job has a due date, which represents the preferred completion time for the job's processing. Meeting this due date is desirable for optimal scheduling. Two penalty parameters, namely α_j and β_j, weigh an early or a tardy completion of job J_j whit respect to its due date d_j.

The objective function we consider is the minimization of the total weighted earliness and tardiness, namely:

$$\min \sum_{j=1}^{n} \alpha_j \cdot E_j + \beta_j \cdot T_j \tag{1}$$

where E_j and T_j are the earliness and tardiness of job J_j, respectively.

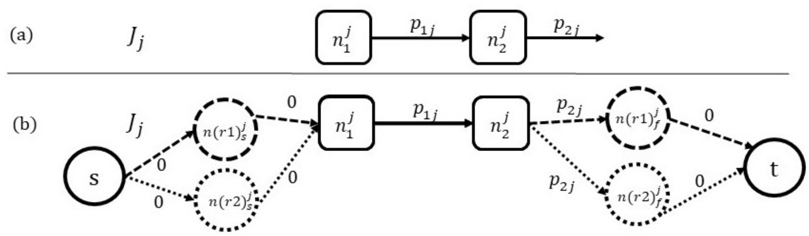

Fig. 1. A graphical representation of job operations precedence arcs and clean room assignment arcs (as in Example 1).

The problem has some standard constraints: the processing of a job cannot start before its release date; each machine can process only one operation at a time; each operation cannot be processed if the previous operation has not been completed; no preemption is allowed; each operation must be processed by exactly one feasible machine.

Moreover, the following constraints are considered: (1) each job must be assigned to exactly one compatible clean room; (2) each clean room can host at most one compatible machine at a time; (3) for each operation, the exact number of human operators specified in the input must be assigned to assist throughout the entire processing time; (4) each human operator can assist at most one operation at a time; (5) no-wait constraints apply between some pairs of consecutive operations of the same job; (6) some operations cannot be processed during the night, therefor must be processed during the day shifts.

The problem described above is modeled as a flexible job-shop scheduling problem with additional resource sets and peculiar constraints due to the safety and quality requirements arising in pharmaceutical production systems.

4. Generalized disjunctive graph model

In the classical disjunctive graph model the graph is denoted as $G = (N, A \cup D)$, where the set of nodes N contains the set of job operations plus, two nodes, s and t, that represent the start and end times. Each node (or outgoing arc) has an associated weight, representing the processing time of the operation it represents. The arc set A includes a set of conjunctive arcs connecting the i-th and $(i+1)$-th operations of a job, while D represents a set of disjunctive arcs connecting operations that are processed on the same machine. We extend the above model to include all the elements and constraints arising in the real world application. The more complex model here proposed is described below.

The generalized disjunctive graph associated to an instance of the problem here considered can be denoted as $G = (N \cup NR, A \cup W \cup R \cup DM \cup DO \cup DS)$, where nodes represent operations or dummy operations and arcs represent several features of a feasible solution that must be decided either by selecting some arcs or by deleting them.

4.1. Node sets

There are two node sets, denoted as N and NR, defined hereafter.

Operation node set N. N is the set of all jobs operations plus an initial node s, a final node t; the node associated to the generic operation o_i^j (that is the $i-th$ operation of job J_j) is denoted as n_i^j.

Clean room assignment node set NR. NR is the set of nodes representing possible assignments of the clean rooms to the jobs. More precisely, for each compatible pair of job J_j and clean room r, we generate two nodes, namely the set $NR_r^j = \{n(r)_s^j, n(r)_f^j)\}$, where $n(r)_s^j \in NR$ and $n(r)_f^j$ represent the starting and the end of the possible execution of J_j in clean room r, respectively. Then $NR = \cup_{(j,r)} NR_r^j$ is the union of all the NR_r^j. Nodes $n(r)_s^j$ for each compatible pair of job J_j and clean room r, are successors of node s and predecessors of the node representing the first operation of job J_j. Similarly, nodes $n(r)_f^j$ precede node t and succeed the last operation of job J_j.

4.2. Arc sets

There are six arc sets, used to represent the different constraints of the problem, denoted as A, W, R, DM, DO, and DS and defined hereafter.

Job operations precedence arc set A. The set A, as in the standard disjunctive graph, is the set of arcs modelling the precedence between each pair of consecutive operations of the same job, so $A \subset N \times N$. Each arc in A that originates from a node $i \in N$ is assigned a weight equal to the processing time of the operation represented by node i.

No-wait constraints arc set W. The set of arcs W model the no-wait constraints between two consecutive operations of the same job. For each pair of such operations an arc is present in W. More precisely, let n_i^j and n_{i+1}^j be two operations of job J_j that must be scheduled consecutively with no idle time, then the no-wait constraint is modelled by a directed arc (n_{i+1}^j, n_i^j) with weight $-p_{ij}$ (where p_{ij} is the processing time of the first operation of the pair, i.e. o_i^j).

Below, several sets of arcs that extend the notion of disjunctive arcs are described. We denote them as *option arcs*, since they usually model a number of possibilities or options, which are greater than two as in the disjunctive arc case. Each option is modelled by a group of at least two arcs, whose meaning is illustrated below. With a slight abuse of notation we call disjunctive these types of arcs.

Job-clean room assignment arc set R. The set R contains the arcs modelling the possibility for a job to be processed in different clean rooms. Let CR_j be the set of clean rooms compatible with job J_j, then for each clean room $r \in CR_j$ a set of four arcs is in the graph: $R_r^j = \{(s, n(r)_s^j), (n(r)_s^j, n_1^j), (n_{n_j}^j, n(r)_f^j), (n(r)_f^j, t)\}$, being n_1^j and $n_{n_j}^j$ the nodes associated to the first and the last operation of job J_j, respectively. For each job a set $R^j = \cup_{r \in CR_j} R_r^j$ of type-R of arcs is defined. Hence, the set R is given by $R = \cup_j R^j$ The arcs in R have different weights: those outgoing node s have weight equal to the release date of the corresponding job; those outgoing the last operation of each job have weight equal to the processing time of the corresponding operation (i.e. for job J_j is equal to $p_{n_j j}$); the remaining ones have null weight.

Example 1. Let us consider a job J_j with two operations, namely o_1^j and o_2^j, and null release time. The nodes and arcs associated to J_j form a chain as depicted in Figure 1(a) where each squared node represents an operation. If job J_j is compatible with clean rooms r_1 and r_2, then arc set R includes nodes and arcs allowing the assignment of job J_j to exactly one of these two clean rooms.

So, besides considering the two nodes s and t, we have two nodes associated to clean room r_1, namely $n(r_1)^j_s$ and $n(r_1)^j_f$ and four arcs, (see the dashed ones in Figure 1). Analogously, for clean room r_2, we have two nodes $n(r_2)^j_s$ and $n(r_2)^j_f$ and four arcs.

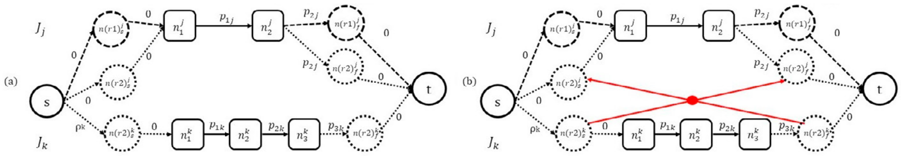

Fig. 2. A graphical representation of clean room disjunctive arcs as in Example 2.

Clean room disjunctive arc set *DR*. The arc set DR consists of clean room-disjunctive arc pairs that model the conflict arising between two jobs that can be processed in the same clean room. Let J_j and J_k be two conflicting jobs and the two sets of four arcs in R that model the assignment to clean room r be $R^j_r = \{(s, n(r)^j_s), (n(r)^j_s, n^j_1), (n^j_{n_h}, n(r)^j_f)(n(r)^j_f, t)\}$ and $R^k_r = \{(s, n(r)^k_s), (n(r)^k_s, n^k_1), (n^K_{n_k}, n(r)^k_f)(n(r)^k_f, t)\}$, respectively. Then, the disjunctive pair is formed by the directed arcs of null weight $(n(r)^j_f, n(r)^k_s)$ and $(n(r)^k_f, n(r)^j_s)$ representing the precedence of J_j over J_k in the processing in clean room r and the precedence of J_k over J_j, respectively.

Example 2. Besides job J_j of Example 1, we also consider job J_k with release ρ_k, formed by three operations and compatible with clean room r_2. In Figure 2 (a) we illustrate the square node chains related to the two jobs J_j and J_k, the nodes related to the clean rooms r_1 (only for job J_j) and r_2 (for both jobs) and the dotted clean room disjunctive arcs. In Figure 2 (b) we also have the disjunctive arc pair for clean room r_2, in red.

Machine disjunctive arc set *DM*. The set DM includes the machine-disjunctive arcs triplets between two operations belonging to two different jobs competing for the use of the same machine. Each triplet is formed by (i) two directed and opposite arcs linking the two nodes associated to the two competing operations, representing the two possible priorities between the operations and (ii) an undirected arc between the same two node. This undirected arc is used for modelling purposes only, since it represents the possibility of overlapping the processing of the two operations and hence, this is equivalent to removing it from the graph. Note that the undirected arc can be considered for a feasible solution only if the system is equipped with at least two copies of the machine that can process the two operations. Each directed arc outgoing a node $i \in N$ has a weight equal to the processing time of the operation corresponding to node i; the undirected arc has null weight.

Example 3. Let us consider the two jobs of Example 2 and assume that the first operation of both jobs must be processed by a machine of type M_1, while the second one (again, for both jobs) by M_2. So, as depicted in Figure 3a, for each pair of conflicting operations a triplet of arcs is present in the graph. The triplet for operation pair o^j_1 and o^k_1 (see the blue colored arcs) is formed by two directed arcs, namely (n^j_1, n^k_1) with weight p_{1j}, and (n^k_1, n^j_1) with weight p_{1k} and one undirected arc $[n^j_1, n^k_1]$ of null weight. Similarly, the triplet of arcs for operation pair o^j_2 and o^k_2 are depicted in green in Figure 3a.

Operator disjunctive arc set *DO*. The set of operator-disjunctive arcs DO contains triplets of arcs between each pair of nodes representing two operations (belonging to two different jobs), both requesting the assistance of some human operators. The triplet is formed by (i) two directed and opposite arcs linking the two nodes associated to the two operations, representing the two possible precedences between the operations and (ii) one undirected arc between the same two nodes, representing the possibility of overlapping the processing of the two operations (this may happen if

(a) Machine disjunctive arc triplets. (b) Operator disjunctive-arcs triplets.

Fig. 3. Graphical representations of machine and operator disjunctive arcs.

there are enough available operators). Note that operations that do not require any operator are not connected by these operator-disjunctive arc triplets. Each directed arc in DO outgoing a node $i \in N$ has a weight equal to the processing time of the operation corresponding to node i, while the undirected arc has null weight.

Example 4. Let us consider the two jobs in the previous examples and assume that the second operation of job J_j, o_2^j, and the third operation of job J_k, o_3^k, must be assisted by ω_2^j and ω_3^k operators, respectively. The triplet of arcs (coloured in orange in Figure 3b) in DO between the two nodes has two oriented arcs, namely (n_2^j, n_3^k) with weight p_{2j}, and (n_3^k, n_2^j) with weight p_{3k} and one undirected arc $[n_2^j, n_3^k]$ of null weight.

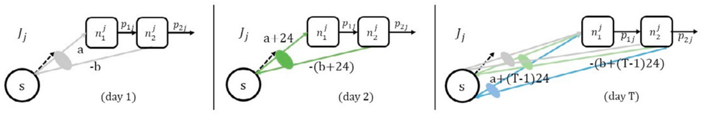

Fig. 4. Graphical representation of day-shift arc pairs.

Day-shift constraints arc set DS. The set of arcs DS models the day-shift constraints. The day shift of each day starts at time a and ends at time b and therefore lasts $b - a$ time units. DS is composed by a subset of arcs associated to each operation i of job J_j subject to the daily-shift constraint: this subset is denoted as DS_i^j. Hence, $DS = \bigcup_{(i,j)} DS_i^j$. Each set DS_i^j is formed by a bundle of T pairs of arcs (being T the number of days in the considered time horizon). Each pair is formed by two directed arcs, namely (s, n_i^j) and (n_{i+1}^j, s). For each day t, the weight of arcs (s, n_i^j) are equal to $a + 24(t-1)$, while the weight of arcs (n_{i+1}^j, s) are equal to $-(b + 24(t-1))$.

Example 5. Let us suppose that the first operation of job J_j is subject to the day-shift constraint. In Figure 4 (day 1) the grey pair of arcs is related to the first day, the arcs weights are a and $-b$ respectively; the green pair is related to day 2, its arcs are weighted $a + 24$ and $-(b + 24)$ (see Figure 4 (day 2)); finally the pair associated to day T is depicted in blue, the two arcs are weighted $a + (T-1)24$ and $-(b + (T-1)24)$ respectively.

4.3. Selections and solutions

Once the disjunctive graph associated to an instance of our problem is built, in order to determine a feasible solution a number of choices (selections) on disjunctive arcs must be made. More in detail, a *complete selection* S is associated to all the following choices:

- for each subset $R^j \subseteq R$ exactly one set of four arcs must be selected, the others are removed;
- for each pair of arcs in DR exactly one arc is selected, the other is removed;
- for each triplet of arcs in DM exactly one arc is to be selected, the others are removed;
- for each triplet of arcs in DO exactly one arc is selected, the others are removed;
- for each set $DS_i^j \subseteq DS$ a pair is selected, the others are removed.

The meaning of this set of choices is briefly explained hereafter. The selection of a group of four arcs in R^j corresponds to the assignment of a job to a specific clean room, where the job will be processed. Recalling Example 1, the selection of the set of four arcs that assigns job J_j to clean room r_2 is illustrated in Figure 5 (note that, the other set of four arcs related to clean room r_1 is removed). A selection of an arc of a pair in DR fixes a precedence between the two corresponding jobs conflicting on a same clean room. Regarding the triplets of arcs in DM and DO, if a directed arc of a triplet is selected a precedence between two operations is fixed; otherwise, if the undirected arc is selected, it is assumed that the processing of the two operations can overlap. The arc pair selected in DS_i^j implies that the operation i is processed and completes in the day shift associated to the day represented by the arc pair.

A feasible solution of our scheduling problem is associated to a partial or complete selection S. In a *partial* selection, the choice for the above arc sets is restricted only to a proper subset of options.

We denote by $R(S)$, $DR(S)$, $DM(S)$, $DO(S)$, and $DS(S)$ the arcs corresponding to a selection S. A feasible solution, associated to a (complete or partial) selection S is represented by a graph $G(S)$ such that the following conditions hold:

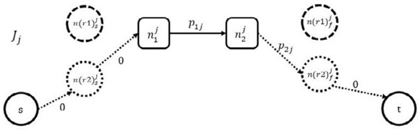

Fig. 5. An example of the selection corresponding to the assignment of job J_j to clean room r_2

- $G(S)$ does not contain any directed cycle of positive length (otherwise, it would mean that an operation is preceded by itself);
- Let $TG(S)$ be a directed graph corresponding to the transitive closure of graph $G(S)$, in which only the directed arcs of selection S are considered. For each machine type M_ℓ, with two or more machine copies (i.e. with $m_\ell \geq 2$), let Γ^ℓ be the sub-graph of $TG(S)$ induced by nodes associated to the operations that are to be processed by a machine of type M_ℓ. Let $\tilde{\Gamma^\ell}$ be the undirected graph complement of Γ^ℓ. Then, the cardinality of the maximum clique in $\tilde{\Gamma^\ell}$ must not exceed m_ℓ, the number of copies of machines of type M_ℓ.
- Let Ω denote the sub-graph of the transitive closure $TG(S)$ induced by the nodes incident to the (directed) selected arcs in $DO(S)$. Let $\tilde{\Omega}$ be the (undirected) complement graph of Ω. For each clique in $\tilde{\Omega}$ the sum of the operators requested for the operations of the clique must not exceed the number of total available operators p.

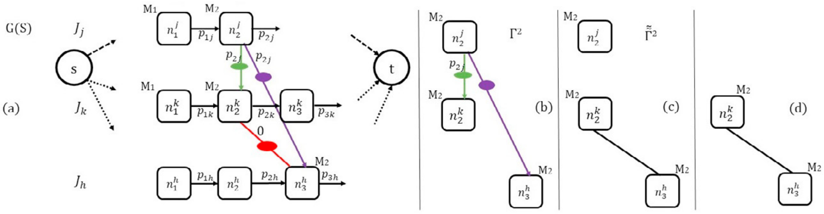

Fig. 6. A representation of the procedure for determining the feasibility of a solution

Example 6. Let us expand Example 3 by adding job J_h to the previously considered jobs J_j and J_k and focus on machines of type M_2. Let operations o_2^j, o_2^k and o_3^h request a machine type M_2 (Figure 6(a)). The (partial) selection S, depicted in Figure 6(a), contains two directed arcs, (n_2^j, n_2^k) and (n_2^j, n_3^h), and an undirected arc, $[n_2^k, n_3^h]$. In Figure 6(b) the sub-graph Γ^2 of the transitive closure $TG(S)$, induced by nodes associated to the operations that must be processed by a machine of type M_2 is reported. In Figure 6(c) the complement graph of Γ^2, denoted as $\tilde{\Gamma}^2$, is depicted. The maximum clique of such graph is selected in Figure 6(d) and has cardinality equal to two. Hence, the two operations of the clique could overlap their processing. Therefore, the solution associated to this (partial) selection S is feasible if the number of machines of type M_2 is at least two, otherwise S is associated to an unfeasible solution.

4.4. Preliminary Computational Tests

In our very preliminary tests we generated several instances and their corresponding disjunctive graphs based on 15 jobs, representing a one-month plan in a real-world scenario of the pharmaceutical company under consideration. These instances were created as a preliminary step to further analyze and evaluate the problem. The production plant at hand is specialized in the micronization of active pharmaceutical ingredients and has $r = 6$ clean rooms, $\mu = 20$ machine types (and a total of 28 machines), and $p = 8$ human operators. Each job has a number of operations n_j ranging from a minimum of 7 to a maximum of 8. Starting from the available real data, we generated four different classes of instances and the corresponding disjunctive graphs. All the tests were implemented by developing a Java procedure, using a computer a equipped with a 64 bit Windows operating system, with CPU 3 GHz and 8GB of RAM. To give an idea of the size of the disjunctive graphs associated to the generated instances, in Table 1 is reported the average number of nodes (N) and arcs (A) for four instance classes, each characterized by the number of jobs n. The average computation time needed to built a generalized disjunctive graph is less than one second in all classes. So far,

Table 1. Disjunctive graph size in the generated instances.

n	N	A
2	20	120
5	54	1058
10	108	4445
15	296	10122

we have not tested any procedure for verifying the feasibility of a selection, but we plan to design and develop ad-hoc heuristics using approaches similar to those presented in [1].

5. Conclusions

In this study, we have introduced a novel extension of the disjunctive graph model to incorporate various characteristics of the real-world scheduling problem considered. While the primary focus of the proposed model is on theoretical representation, preliminary computational tests indicate that it can be effectively utilized heuristically with reasonable performance. As a result, future research directions will involve streamlining the model to enhance its usability and practicality. The aim is to simplify the described model for easier implementation and broader applicability.

Also heuristic approaches based on mathematical models [11, 12], or genetic algorithms, as in [5], are among the techniques being explored. Various other methodologies, including an ad-hoc Branch and Bound based on different lower bounds [13, 14, 15], are also being taken into consideration for tackling the problem.

References

[1] A. Agnetis, M. Flamini, G. Nicosia, A. Pacifici, A job-shop problem with one additional resource type, Journal of Scheduling 14 (3) (2011) 225 – 237.

[2] A. Sharma, N. Anup, R. K. Tekade, Achieving sterility in biomedical and pharmaceutical products (part-I): thermal, chemical, and filtration sterilization, in: R. K. Tekade (Ed.), The Future of Pharmaceutical Product Development and Research, Advances in Pharmaceutical Product Development and Research, Academic Press, 2020, pp. 695–788.

[3] J. Błażewicz, J. Lenstra, A. Rinnooy Kan, Scheduling subject to resource constraints: classification and complexity, Discrete Applied Mathematics 5 (1) (1983) 11–24.

[4] R. L. Burdett, P. Corry, C. Eustace, S. Smith, A flexible job shop scheduling approach with operators for coal export terminals – a mature approach, Computers & Operations Research 115 (2020).

[5] R. Mencía, M. R. Sierra, C. Mencía, R. Varela, Memetic algorithms for the job shop scheduling problem with operators, Applied Soft Computing 34 (2015) 94–105.

[6] B. Roy, B. Sussmann, Les problemes d' ordonnancement avec constraints disjonctives, SEMA, Note D.S., Paris (1964).

[7] E. Balas, Machine sequencing via disjunctive graphs: An implicit enumeration algorithm, Operations Research 17 (6) (1969) 941–957.

[8] V. Boyer, J. Vallikavungal, X. Cantu Rodriguez, M. A. Salazar-Aguilar, The generalized flexible job shop scheduling problem, Computers & Industrial Engineering 160 (2021) 107542.

[9] J. Hurink, S. Knust, Tabu search algorithms for job-shop problems with a single transport robot, European Journal of Operational Research 162 (1) (2005) 99–111.

[10] Y.-J. Yao, Q.-H. Liu, X.-Y. Li, L. Gao, A novel MILP model for job shop scheduling problem with mobile robots, Robotics and Computer-Integrated Manufacturing 81 (2023) 102506.

[11] E. Pastore, A. Alfieri, C. Castiglione, G. Nicosia, F. Salassa, A matheuristic approach to the open shop scheduling problem with sequence-dependent setup times, IFAC-PapersOnLine 55 (10) (2022) 2167 – 2172.

[12] M. Cosmi, G. Nicosia, A. Pacifici, Scheduling for last-mile meal-delivery processes, IFAC-PapersOnLine 52 (13) (2019) 511 – 516.

[13] P. Baptiste, M. Flamini, F. Sourd, Lagrangian bounds for just-in-time job-shop scheduling, Computers & Operations Research 35 (3) (2008) 906 – 915.

[14] F. Daneshamooz, P. Fattahi, S. M. H. Hosseini, Mathematical modeling and two efficient branch and bound algorithms for job shop scheduling problem followed by an assembly stage, Kybernetes 50 (12) (2021) 3222–3245.

[15] S. Tanaka, R. Sadykov, B. Detienne, A new lagrangian bound for the min-sum job-shop scheduling, in: International Symposium on Scheduling ISS'2015, 2015.