Minimizing the costs induced by perishable resource waste in a chemotherapy production unit

Alexis ROBBES[1], Yannick KERGOSIEN[1], Virginie ANDRÉ[2] and Jean-Charles BILLAUT[1]

[1] Université de Tours, LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002, 64 avenue Jean Portalis, 37200 Tours
{alexis.robbes, yannick.kergosien, jean-charles.billaut}@univ-tours.fr
[2] CHRU de Tours, Hôpital Bretonneau, 2 boulevard Tonnellé, 37044 Tours Cedex 9
v.andre@chu-tours.fr

Keywords: Scheduling, Bin Packing, Healthcare, Stock management, Perishable resource.

1 Introduction

The UBCO (Bio pharmaceutical Unit of Oncology Clinic of the hospital of Tours) produces around 150 chemotherapy drugs per day for three hospital units of Tours (France). Perishable resources (molecules) are needed to prepare the drugs, precisely one molecule type per Chemotherapy drug. Each molecule type has its own stability time (shelf life after opening) and cost. The UBCO aims to respect a production lead-time so that the patient waiting time is reduced and to minimize the production cost. Around 40 molecules types are used for the production. The cost per vial of each molecule type varies between few euros and more than 1200€. The production is a three-step process: sterilization of the resources, preparation of the drug inside an isolator by an operator and quality control of a sample by an automatic analyzer. In (Robbes *et. al.* 2019), a chemotherapy drug production is modelized as a Hybrid Flow-shop scheduling problem and minimizing the production and delivery delay is solved by a heuristic. However the heuristic presented does not take account of the costs induced by the waste of perishable resources. In this paper we propose a parallel machine modelization of the chemotherapy production scheduling problem (simplified model) and a matheuristic algorithm. The goal is to find a schedule for the chemotherapy production which minimizes the total cost of the perishable resources.

2 Problem definition

A chemotherapy drug production plant is composed of a set I of parallel identical isolators, each one is composed by m work stations (machines). This production can be modelized as a parallel machines scheduling problem where machines are gathered into groups corresponding to the isolators. A job can be processed on any isolator and on any machine of the isolator. The chemotherapy drugs are the jobs and the molecules used for the preparation are perishable resources used by the jobs. The preparation of one chemotherapy drug j uses only one type of molecule. The same vial can be used for several jobs. When one vial is assigned to two jobs on two different isolators, the sterilization time σ has to be taken into account. Each job j is also characterized by a preparation processing time p_j and a required resource quantity q_j of molecule type μ_j. Each perishable resource $r \in R$ is stored in vials of volume V_r with a price of $cost_r$ and has a stability time γ_r. We consider a planning horizon of $|D|$ consecutive days, knowing that no drug can be produced during the week-end. The scheduling variables are the starting time s_j (day and hour) and completion time C_j of each job j. The vial index assigned to the job j is noted f_j. We denote by z_r the cost induced by the perishable resource r and LB_r is the lower bound of z_r. The problem

is to assign the jobs to the machines, to set their starting times and to assign a vial of perishable resource to each job, in order to minimize the total cost.

3 Matheuristic method

We propose a matheuristic method to solve this problem. We first generate an initial solution where jobs are assigned to the machines. Then, the algorithm is based on a Gradient Descent algorithm with a Tabu list, which explores neighborhoods for the assignment of jobs, and for the sequencing of jobs on the machines. The assignment of the vials to the jobs is done, molecule type per molecule type, by solving a Bin Packing problem with Conflicts. To compute the initial solution, we sort the molecule types by decreasing $cost_r$. Then, for each molecule type in this order, we assign all the jobs using the molecule type to the minimum possible number of machines. This behaviour tends to allow more possibilities of vial assignment by reducing the number of jobs using the same molecule type at the same time. Grouping the jobs using a same molecule type on few days reduces the number of vial assignment incompatibilities due to the molecule stability time.

3.1 Assignment to vials: Bin Packing problem with Conflicts

The Bin Packing problem with Conflicts (BPC) is a variant of the Bin Packing problem, where a conflict graph defines which pair of jobs cannot be packed together in the same bin. The one-dimensional version was introduced in (Jansen and Öhring 1997).

A conflict graph is generated from conflict rules. An edge corresponds to an incompatibility between two jobs to use the same vial. There are 3 conflicts cases for two jobs j and j' using the same type of perishable resource ($\mu_j = \mu_{j'}$):

1. **Machine overlap conflict**: the jobs are performed by different machines at the same time, i.e. $\max(s_j, s_{j'}) < \min(C_j, C_{j'})$
2. **Isolator overlap conflict**: the jobs are performed in different isolators and the time gap is smaller than the sterilization time, i.e. $\max(s_j, s_{j'}) - \sigma < \min(C_j, C_{j'})$
3. **Stability conflict**: the jobs are performed with a time gap greater than the stability time of the molecule type, i.e. $\max(C_j, C_{j'}) - \min(s_j, s_{j'}) > \gamma_{\mu_j}$

The BPC instance built from this conflict graph is solved using an Integer Linear Programming (ILP) modelization based on the new formulation of the Bin Packing Problem proposed in (Hadj Salem and Kieffer 2019).

3.2 Local permutation: Non Destructive Permutation

The Gradient Descent algorithm with a Tabu list executes as much as possible pairwise permutations on the schedule. Evaluating the schedule after each permutation is time consuming. To avoid a costly neighbourhood exploration, we choose to focus the on permutations which are consistent with the vials assignments (BPCs solutions) and do not need to solve the BPCs again.

We define a Non Destructive Permutation (NDP) as a job pairwise permutation respecting some rules. Let consider two jobs j and j' and a vial f containing the molecule type μ_j that is assigned to other jobs. The goal to assign vial f ($f \neq f_j$) to j in order to reduce the number of vials of μ_j molecule type. The remaining volume of vial f should be greater than q_j.

The idea is to swap job j with j' in the schedule, in order to modify the conflict graph of the molecule type μ_j. A permutation between j and j' is Non Destructive if the swap between j and j' does not modify the vial assigned to j' (the assignment of f'_j to j' remains consistent). The following constraints have to be satisfied.

- $z_{\mu_j} \neq LB_{\mu_j}$ i.e. it is possible to reduce the number of vials.
- the vial f_j used for j is not totally used.
- $p_j = p_{j'}$ i.e. the permutation does not impact the rest of the schedule.
- $\mu_j \neq \mu_{j'}$ i.e. the two jobs are not using the same molecule type.
- for all j" as $f_{j"} = f_{j'}$, swapping j and j' does not generate conflict between j' and j".

To avoid any cyclic behaviour, all jobs permuted are added in a Tabu list. Any new NDP that implies a job in the Tabu list is forbiden.

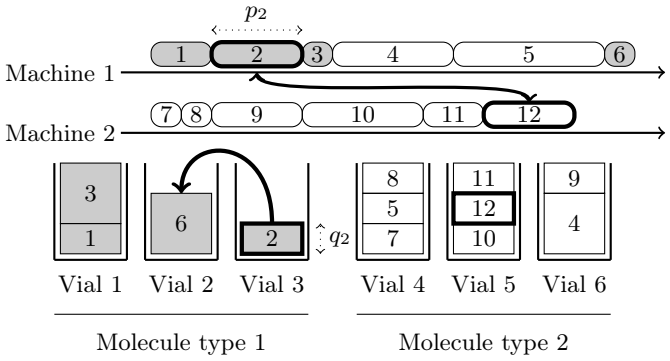

Fig. 1. Illustration of a Non Destructive Permutation

Fig. 1 shows a possible NDP between the jobs 2 and 12 by changing the assigned Vial $f_2 = 3$ to $f_2 = 2$. This NDP reduces the number of opened vial of Molecule type 1. Note that even if the job 12 is swapped with 2, no conflict will be created between jobs assigned to Vial 5.

4 Computational Experiments

Instances have been generated based on the real-life application at UBCO. The study have been limited to 6 types of perishable resources. For each perishable resource r, the vial volume is set to $V_r = 10$. For each job j, q_j is a random value in $\{1, \ldots, 10\}$ and p_j is a random value between 5, 10 and 15 minutes. The sterilization time σ is set to 15 minutes. A production day is 8 hours of work and a week has 5 production days. We consider a time horizon $|D|$ of 2 and 3 and 4 weeks. The number of isolators $|I|$ belongs to 2,3 and the number of machines per isolator is equal to $m = 2$.

Table 1. Perishable resources characteristics

r	1	2	3	4	5	6
Stability time γ_r	1 day	7 days	9 days	30 days	30 days	30 days
Cost per vial (€)	380	35	1195	25	405	740
% of the jobs requiring r	9	13	2	56	9	11

Table 1 presents the 3 characteristics of the 6 perishable resources: the stability times, the cost per vial and the percentages of job using the resource. We fix the computation times limit to solve an ILP to 120 seconds, with Gurobi Optimizer. The experimentation was performed on 10 instances for each instance size. The size of an instance is defined by

the schedule horizon (number of days or weeks to schedule), the number of isolators $|I|$ and m the number of work stations per isolator. The number of jobs depends on the jobs processing times, and we have $\sum_{j=1}^{n} p_j = 0.9 \times |D| \times 8 \times 60 \times |I| \times m$.

Table 2. Average Gap and average number of NDP depending of the initialization

Instance size		Gap%				Nb NDP	
Horizon $\|D\|$	$\|I\|$	Grouping		Random		Grouping	Random
		before GD	after GD	before GD	after GD		
2 weeks	2 isolators	0.21	**0.17**	0.37	0.36	0.7	0.7
2 weeks	3 isolators	0.97	0.81	0.77	**0.62**	4.2	2.4
3 weeks	2 isolators	0.95	**0.78**	1.39	1.02	2.4	7.3
3 weeks	3 isolators	2.12	**1.53**	2.93	1.85	10.3	15.3
4 weeks	2 isolators	2.04	**1.57**	3.58	1.99	6.4	21.3
4 weeks	3 isolators	2.29	1.65	2.92	**1.62**	13.3	24.8

In Table 2, the gap is defined by $100 * \frac{\sum z_r - \sum LB_r}{\sum LB_r}$, two initialisation methods are compared, the proposed Grouping scheduling and a Random one. The column "before GD" represents the average gap of the initial solution and the column "after GD" represents the average gap at the end of the Grandient Descent. "Nb NDP" is the average number of NDP performed by the Gradient Descent algorithm. The average gap between our matheuristic and the lower bound is very low. It seems that the initialisation method does not have an important impact on the results after the GD but the number of NDP is greater for a random initialisation than the Grouping heuristic. The gap of a random initialisation is quite small even before the GD, which means that for our instances, the most important part to reduce the total cost is not the scheduling but the vial assignments. Further experimentations have to be done to confirm this claim.

5 Conclusion and perspectives

In this paper, we propose a matheuristic algorithm to minimize the costs induced by perishable resources waste in a chemotherapy drugs production unit. We modelize the production as a parallel machine scheduling problem and present a Gradient Descent algorithm with a Tabu list where the perishable resources assignment is done by solving several Bin Packing problem with conflicts. The computational results shows that with a short production time horizon, we obtain solutions with a production cost near to its lower bound. However, a large planning horizon needs multiple Gradient Descent steps before finding a local minimum. The perspectives are to use the branch and price proposed in (Sadykov and Vanderbeck 2012) to decrease the computation time to solve each Bin Packing problem.

References

Hadj Salem K., Y. Kieffer, 2019, “Nouvelle formulation en PLNE pour le problème classique du Bin Packing", *ROADEF 2019*.

Jansen K., S. Öhring, 2012, “Approximation Algorithms for Time Constrained Scheduling", *Information and Computation*, Vol. 132, pp. 85-108.

Robbes A., Y. Kergosien and J-C. Billaut, 2019, “Multi-level heuristic to optimize the chemotherapy production and delivery", *Health Care Systems Engineering: HCSE 2019*.

Sadykov R., F. Vanderbeck, 2012, “Bin Packing with Conflicts: A Generic Branch-and-Price Algorithm", *INFORMS Journal on Computing*, Vol. 25.