



HAL
open science

A generic decision support tool for lot-sizing and scheduling problems with setup and due dates

Cristovao Silva, Nathalie Klement, Olivier Gibaru

► To cite this version:

Cristovao Silva, Nathalie Klement, Olivier Gibaru. A generic decision support tool for lot-sizing and scheduling problems with setup and due dates. International Joint Conference - CIO-ICIEOM-IIE-AIM (IJC 2016), Jul 2016, San Sebastián, Spain. 10.1007/978-3-319-58409-6_15 . hal-03165141

HAL Id: hal-03165141

<https://hal.science/hal-03165141>

Submitted on 10 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A generic decision support tool for lot-sizing and scheduling problems with setup and due dates

Silva C¹, Klement N², Gibaru O²

Abstract: Decision support tools are essential to help in the management of industrial systems at different levels: strategic, to design the system; tactical to plan activities or assign resources; operational to schedule activities. In this paper, we present a generic and modular decision support tool to solve different planning, assignment, scheduling or lot-sizing problems. To the best of our knowledge, such generic tool does not exist. The methodology is illustrated by solving a real world lot-sizing and scheduling problem from a plastic injection company.

Keywords: Decision support tool; Scheduling; Lot-Sizing; Injection moulding

1 Introduction

Nowadays, industry is changing. We are in the fourth industrial revolution. Industrial system has to be more and more intelligent and flexible. It has to be more intelligent to anticipate a shortage of raw materials, a maintenance operation or an emergency command. It can be more intelligent thanks to the internet of objects. It has to be more flexible to adapt itself to the future commands. It could be more flexible thanks to a tool which would model different new configurations of the company and evaluate the consequences of the potential changes.

Our main purpose in this article is to present a decision support tool which can be used in many industrial cases at different levels, thus allowing increasing company flexibility. This decision support tool can be used to plan activities and assign them resources according to several constraints. In this paper, we illustrate

¹Cristovao Silva (✉e-mail: cristovao.silva@dem.uc.pt)
Dpto. de Engenharia Mecânica. CEMUC. Universidad de Coimbra. Rua Luís Reis Santos, 3030-788 Coimbra, Portugal.

²Nathalie Klement (e-mail: nathalie.klement@ensam.eu)
Olivier Gibaru (e-mail: olivier.gibaru@ensam.eu)
LSIS CNRS UMR 7296 (équipe INSM). Ecole Nationale des Arts et Métiers. 8, boulevard Louis XIV - 59046 Lille Cedex. France

how the proposed tool can be used at the tactical level: which resource can do each activity and when; and at the operational level: determine the schedule of activities assigned to each resource. At this level, it could also consider emergencies, like arriving jobs that have to be done as soon as possible. But it could also be used at the strategic level: according to activities which have to be done in a horizon planning, how many resources do we need?

The proposed tool is composed by a generic module, which can be used in several different problems, and three specific steps conducted according to the considered problem. Thus, the proposed tool can support the decision process for several different planning problems with a minimum development work. The method used by the tool is a hybridization of a metaheuristic and a list algorithm. The metaheuristic can be used without any changes and the list algorithm needs to be adapted according to the studied problem.

The general method is described in Section 2. Section 3 specifies the problem considered in this paper to test the tool. The list algorithm developed to solve the proposed problem is given in Section 4. The different results are presented in Section 5. Then, the paper ends with the main conclusions and proposed further work.

2 Method: hybridization of a metaheuristic and a list algorithm

The proposed tool uses a hybridization of a metaheuristic and a heuristic, more precisely a list algorithm, see Figure 1. A single solution based metaheuristic or a population based metaheuristic can be used. The encoding used by the metaheuristic is a list Y of jobs. In the single solution based metaheuristic case, the neighbourhood system is a permutation of jobs. The list algorithm considers the jobs according to their order in list Y to schedule and assign them to the required resource, considering the problem constraints. This builds the solution X . The objective function H evaluates the solution X . According to this evaluation, the solution is chosen or not by the metaheuristic. At the end of the running, the given solution by the hybridization is the best list of jobs: the one which optimizes the objective function by applying the list algorithm.

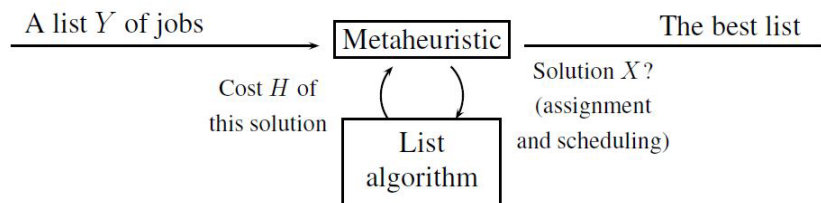


Fig 1. Hybridization metaheuristic - list algorithm

This hybridization can be used to solve many problems: the specificity of a given problem is only considered in the list algorithm. Three steps need to be specified according to the studied problem:

- Problem specification: definition of the constraints and objective function of the problem, presented in Section 3.
- List algorithm: developed according to the problem specification. The list algorithm proposed to solve the problem considered in this paper is presented in Section 4.
- Generation of instances: data of the problem are used to characterize the jobs and to implement the list algorithm, presented in Section 5.

3 Considered problem

The proposed tool was already used to solve a hospital activities planning and resource assignment problem (Klement, 2014). The current paper deals with the use of the same tool to solve a lot-sizing and scheduling problem found in a plastic injection company, previously described in (Silva and Ferreira, 2004), thus showing its ability to be adaptable to several different problems.

The problem presented in this paper has been motivated by a production planning and scheduling problem faced by a company that produces small plastic parts for the electric/electronic industry. The products are obtained by injection moulding, a process in which a polymer is heated to a plastic state and forced to flow into a mould cavity, where it solidifies. The moulded part is then removed from the cavity. The process produces components that are almost always net shape. Thus, injection moulding is typically a single-stage manufacturing process where a number of products are manufactured on shared machines with their respective tools. The mould is the main tool used in the injection moulding, being custom designed for a given part to be produced. In each cycle the injection machine close the mould, inject melted plastic into the cavity, open the mould and eject the plastic part.

The company considered in this paper manufactures a wide range of products, injected in more than 500 different moulds, owned by the clients, but managed by the company. To satisfy the costumers' orders, the company must plan the execution of more than 300 production orders per week. The company shop-floor is composed by 25 injection moulding machines. Once a week, the production planner considers all the orders to be produced (characterized by the quantity of items to be produced and the required due date) and define a production plan for the next week, allocating them to the available machines and defining the sequence by which they have to be produced.

The problem can be stated as follow: A set of n jobs has to be scheduled on shared machines with their respective mould. Each job has a given size, which determines its processing time, and an associated due date. A sequence dependent setup time is required when the production changes over from a job requiring a given mould to a job requiring a different one. A job is not allowed to be split but several jobs, requiring the same mould, may be grouped together to form one lot and, thus, saving setup costs. Due to compatibility factors, each mould can only be allocated to a subset of the available machines. Each mould is unique; thus the

same mould cannot be allocated to different machines during the same time period. The objective is to allocate jobs to each available machine and define the processing sequence in each machine in order to minimize the total tardiness.

The problem described previously can be seen as a lot-sizing and scheduling problem. A small literature review about this kind of problems is given below.

One definition of lot-sizing problem can be the following: given a number of jobs to process and a number of available machines, the objective is to determine the best size of lots of items, to assign them to the machines and to schedule them. The problem can be single-item, or multi-item. It can be single-level, if jobs need to be processed by one machine; or multi-level, if they need to be processed by several machines consecutively. The demand can be static or dynamic. The planning horizon is divided into periods, called time-bucket, which can be big or small. In big bucket problems, a period is long enough to produce multiple items. In short bucket problems, time period is so short that only one type of item can be produced in each time period (Gicquel *et al.*, 2008). (Drexel and Kimms, 1997) makes a chronology of existing lot-sizing problems. A recent review of lot-sizing problems is done in (Copil *et al.*, 2016).

The problem described in this paper is a particular case of the lot-sizing problem which considers big time-buckets, called Capacitated Lot-Sizing and Scheduling Problem (CLSP). The CLSP is defined by a planning horizon (finite or infinite), a number of levels (single-level or multi-level), a number of jobs, some capacity or resource constraints, the demand (static or dynamic), and the setup structure (simple or complex if setup costs or times are sequence dependent), (Karimi *et al.*, 2003).

Since in our problem we have parallel machines and a given job can only be allocated to a restricted number of machines, due to mould/machine compatibility constraints, two approaches can be considered.

The first approach consists in considering the shop-floor composed by a set of unrelated machines: the job processing time is either the one given if the machine can use the needed mould, or infinite otherwise. In their literature review about lot-sizing and scheduling problems with sequence-dependent setup, (Zhu and Wilhelm, 2006) distinguishes identical machines (processing time of a job on a machine depends only on the job), uniform machines (processing time depends only on the job and the speed of the machine to which it is assigned) and unrelated machines (processing time depends both on the job and the machine to which it is assigned). Around eighty articles are referenced in the literature review, but only seven of them considered unrelated machines.

Lot-sizing problems with unrelated machines are NP-Hard (Anderson *et al.*, 1997), thus most papers use approach methods to solve it. (Ozdamar and Birbil, 1998) developed hybrid heuristics involving search techniques such as simulated annealing, tabu search and genetic algorithms to deal with the capacitated lot-sizing and loading problem. (Kimms, 1999) developed genetic algorithms to solve the multi-machine lot-sizing and scheduling problem, considering some scarce resources, but the main assumption is that at most one setup can occur per period.

(Weng et al., 2001) proposed several heuristics to solve the problem of scheduling independent jobs on a set of unrelated parallel machines with sequence dependent setup times. (Kim et al., 2002) used simulated annealing to solve a similar problem but in which items of one job can be split among several machines. (Meyr, 2002) proposed some methods to solve simultaneously the lot-sizing and scheduling problem with non-identical machines. (Dastidar and Nagi, 2005) also solved the unrelated machines case in a plastic injection application, with resource constraints, using a two-phase strategy. (Toledo and Armentano, 2006) used a Lagrangian-based heuristic to solve capacitated lot-sizing problem involving the production of multiple items on unrelated parallel machines. Generally, the developed methods are quite sophisticated.

Another way to consider our problem consists in seeing the shop-floor as being composed by a set of identical machines with eligibility: a machine can be eligible to process a job if the mould used by the job can be used on this machine. Lot-sizing and scheduling problems with respective solving methods, with parallel machines and eligibility constraints, have been considered by (Ruiz and Maroto, 2006) and (Xiao *et al.*, 2015).

Our approach is to consider the plastic injection problem as a CLSP with finite planning horizon, big buckets, single-level, multi-items, capacitated resources, static demand, complex setup structure and parallel identical machines with eligibility.

4 The proposed list algorithm

List scheduling algorithms are one-pass heuristics that are widely used to prescribe schedules. The standard list scheduling algorithm constructs a schedule by assigning each job according to their order in the list to the first machine that becomes idle (Zhu and Wilhelm, 2006). Some list algorithms have been proposed in (Klement, 2014) to solve the problem of activities planning and resources assignment in the hospital case. We propose a new list algorithm to schedule jobs for the proposed injection moulding problem, see Figure 2.

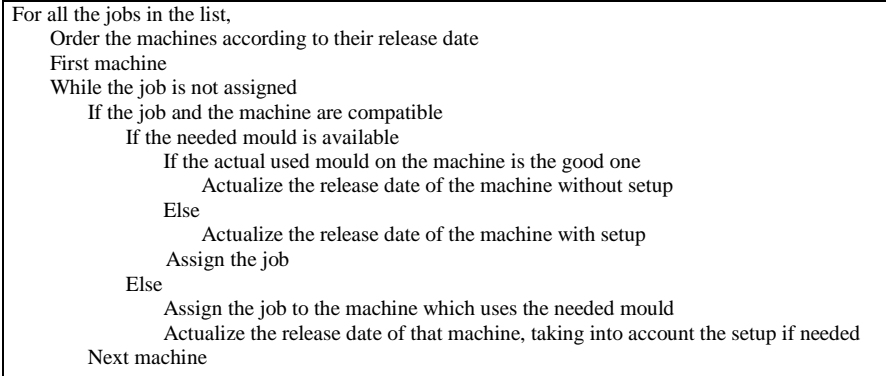


Fig 2. List algorithm for the injection problem

The proposed list algorithm is combined with a single solution based metaheuristic, the stochastic descent, to form the method described in Section 2. In the next section, the results obtained by the proposed method are compared with the results obtained with a two stage heuristic presented in (Silva and Ferreira, 2004).

5 Results

To test the proposed tool, company historical data were collected and used to randomly generate instances of several sizes: 3 instances with 3 machines (16, 18 and 20 moulds; 47, 53 and 57 jobs, respectively); 3 instances with 5 machines (25, 26 and 26 moulds; 81, 80 and 79 jobs) and one instance with 10 machines (59 moulds and 177 jobs). Jobs processing time follow an exponential distribution with an average of 10.75 hours and due dates were generated using a uniform distribution ranging between 24 and 312 hours. Setup times consider the time to dismount the current mould (ranging between 15 and 45 minutes) and to mount the next one (ranging between 20 and 60 minutes).

Table 1 presents the results obtained for each instance size with the heuristic proposed in (Silva and Ferreira, 2004) and with the proposed tool. Comparison considers the average tardiness and average number of setups among all instances of a given size.

Table 1. Results

Instance	(Silva and Ferreira, 2004)		Our proposal	
	Aver. Tardiness	Aver. Setup	Aver. Tardiness	Aver. Setup
3 machines	68,6	21,6	53,6	35,3
5 machines	212,9	32,7	98,3	45,7
10 machines	1020	82	964	114

The results show that the developed tool is effective. An average reduction of 25% of tardiness is achieved. For some instances the reduction of tardiness is up

to 50%. Nevertheless, the proposed method leads to an increase in the number of setups. It is important to note that, since the main objective of the company was to reduce the tardiness, the number of setups was not considered in the objective function of the proposed method. For other problems, an objective function considering tardiness and number of setups, with different weights, can be envisaged. (Silva and Ferreira, 2004) used a two-phase algorithm: first it assigned moulds to machines, and then it schedules jobs on each machine. So it was not possible to assign one mould on several machines at different times. In this way, the proposed method is less constraining which explains the better results.

The proposed method is easy to develop and gives good results. For small instances, 3 and 5 machines, results are achieved in a small computational time; a few minutes. For larger instances, the computational time required to attain a good solution increase and can reach a few hours. That is why for the instance with 10 machines, the proposed methodology doesn't conduct to an improvement as high as the ones found for smaller instances. To reduce the time required to reach good solutions for large instances more efficient metaheuristics, like simulated annealing, are being coded. We expect that, with these metaheuristics it will be possible to solve industrial size problems (20 machines and approximately 300 jobs) in a reasonable amount of time.

6 Conclusions

In this paper, we present a generic decision support tool which can be used to solve several different planning problems with a minimum development work. The application of this tool is illustrated with a real world lot-sizing and scheduling problem, from a plastic injection company, considering sequence dependent setup time, unrelated machines or identical machines and machine eligibility, for multi-items at single level. Results show that the proposed tool can be effectively used to solve the proposed problem, promoting better results, in terms of tardiness, than a two-stage heuristic developed for the same problem in the past. The tool is considered generic because it is easily adaptable to a large range of problems, simply by coding adequate list algorithms.

Several paths can be considered for future development for this work. We intend to code new metaheuristics for the proposed method: other single based solution as simulated annealing or iterated local search, or population based metaheuristics like particle swarm optimization. This will allow assessing the effectiveness of the proposed tool under different metaheuristic approaches. For the case of the problem described in this paper, we intend to develop a mathematical model and use optimization tools to solve it for small instances (since the problem is NP-hard), thus comparing the proposed tool results with optimum ones.

We also intend to develop new list algorithms for different industrial problems. Right now data are being collected in two companies with lot-sizing and scheduling problems. One consists of a Discrete Lot-Sizing and Scheduling problem and the other consists of a Continuous Lot-Sizing and Scheduling problem with multi-

stages. Then, we intend to develop new list algorithms for different scheduling, planning or assignment problems presented in the literature for which instances are known. Ultimately, we will be able to have a tool with a generic part composed by a metaheuristic and a library of list algorithms, which will allow a rapid implementation for different companies, facing different problems.

References

- Anderson, E. J., Glass, C. A., and Potts, C. N. (1997). Machine scheduling. Local search in combinatorial optimization, 11:361–414.
- Copil, K., Wöbelauer, M., Meyr, H., and Tempelmeier, H. (2016). Simultaneous lotsizing and scheduling problems: a classification and review of models. *OR Spectrum*, pages 1–64.
- Dastidar, S. G. and Nagi, R. (2005). Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. *Computers & Operations Research*, 32(11):2987–3005.
- Drexl, A. and Kimms, A. (1997). Lot sizing and scheduling—survey and extensions. *European Journal of Operational Research*, 99(2):221–235.
- Gicquel, C., Minoux, M., and Dallery, Y. (2008). Capacitated lot sizing models: a literature review.
- Karimi, B., Ghomi, S. F., and Wilson, J. (2003). The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31(5):365–378.
- Kim, D.-W., Kim, K.-H., Jang, W., and Chen, F. F. (2002). Unrelated parallel machine scheduling with setup times using simulated annealing. *Robotics and Computer-Integrated Manufacturing*, 18(3):223–231.
- Kimms, A. (1999). A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. *Computers & operations research*, 26(8):829–848.
- Klement, N. (2014). Planification et affectation de ressources dans les réseaux de soin: analogie avec le problème du bin packing, proposition de méthodes approchées. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II.
- Meyr, H. (2002). Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2):277–292.
- Ozdamar, L. and Birbil, S. I. (1998). Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research*, 110(3):525–547.
- Ruiz, R. and Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169(3):781–800.
- Silva, C. and Ferreira, L.M. (2004). Microplano—a scheduling support system for the plastic injection industry. In *E-Manufacturing: Business Paradigms and Supporting Technologies*, pages 81–89. Springer.
- Toledo, F. M. B. and Armentano, V. A. (2006). A lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines. *European Journal of Operational Research*, 175(2):1070–1083.
- Weng, M. X., Lu, J., and Ren, H. (2001). Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *International journal of production economics*, 70(3):215–226.
- Xiao, J., Yang, H., Zhang, C., Zheng, L., and Gupta, J. N. (2015). A hybrid lagrangian simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. *Computers & Operations Research*, 63:72–82.
- Zhu, X. and Wilhelm, W. E. (2006). Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE transactions*, 38(11):987–1007.