A contribution for generalization of scheduling problem classes

Goran D. Putnik
University of Minho
Department of Production Systems Enginnering
4800-058, Guimarães
Portugal
putnikgd@dps.uminho.pt

Cátia Alves
University of Minho
Department of Production Systems Enginnering
4800-058, Guimarães
Portugal
catia.alves@dps.uminho.pt

Leonilde Varela
University of Minho
Department of Production Systems Enginnering
4800-058, Guimarães
Portugal
leonilde@dps.uminho.pt

Paulo A. Ávila
ISEP – Instituto Politécnico do Porto
DEM
4200 Porto
Portugal
psa@isep.ipp.pt

Abstract

It is identified that different scheduling problem classes are just special cases of one "general scheduling problem" that is that a number of traditional scheduling problem classes (for the case of 64 classes obtained by combination of 6 parameters) could be reduced to the scheduling problem for a set of lots, with one part per lot. "General scheduling problem" definition helps to minimize the simulator's size/complexity concerning modeling and evaluation of three scheduling paradigms.
Further, for scheduling problems different notations were developed to represent different scheduling problems classes. In this paper we introduced a new field into the Graham´s notation, named "sub-problem solution classes", considering that the general solution (objective) is integrated over solutions for sub-problems.

Keywords

Scheduling problem, General scheduling problem, Graham's notation, Simulator complexity.

1. Introduction

Scheduling problems has been widely studied. Different notations were developed to represent different scheduling problems classes.

In this paper we present a contribution for generalization of scheduling problem classes. The underlying objective of this paper is to identity the different classes which represent special cases of a general scheduling problem. "General scheduling problem" definition helps to minimize the simulator's size/complexity concerning modelling and evaluation of three scheduling paradigms, which general architecture was presented in (Putnik et al., 2014).

The paper is further organized onward as follows. In Section 2 scheduling problems and solutions classes are briefly presented, and a notation to represent scheduling problems and solutions classes are proposed. Section 3 presents a scheduling problem classes generalization to generalize different problems into one problem. Finally, the major conclusions are presented in Section 4.

2. Scheduling problems and solutions classes

Stecca (2014) defines scheduling problem as "*a finite number n of jobs that need to be processed over a finite number m of machines*".

In the context of this paper job is defined as the work to produce one lot or a part, determined by the "order" for one lot or the "order" for one part, although in practice there could be different associations between orders and jobs. Additionally, it is assumed that there is a correspondence between a lot and the job.

2183-3060/2016/0401-009

Each job corresponds to one process plan and to one piece. On the opposite, to one process plan corresponds different jobs (with the same operations), and each piece corresponds to one job.

Task is the job element, and job is composed by sequences of tasks necessary to complete the job. The task corresponds to one operation of the process plan. On the opposite, to one operation in process plan corresponds different number of the tasks of the same contents.

In other words, process plan is definition of the job in terms of the number of tasks, on which machine tasks must be executed and with associated duration of each task. On other hand, the "operation" (within the process plan) is a detailed definition of the task in terms of how to execute the task, and with repeated information of the machine on which task should be executed and including duration of the task. However, in many companies this information is unfortunately not used in a canonical way and for practitioner many time there is a confusion on terminology use and meaning.

Other synonym for task is activity (Emmons, 1987).

Graham, Lawler, Lenstra, and Kan (1979) introduced the three-field notation α,β,γ, where α represents the machine environment, β represents the job characteristics, and γ represents the optimally criteria (objective).

Other notations where developed in the domain of scheduling such as the well-known nomenclatures such as Conway (Conway, Maxwell, & Miller, 2012), French (French, 1982), Brucker (Brucker, 1995), Blazewicz (Blazewicz, Ecker, Pesch, Schmidt, & Weglarz, 1997) , Pinedo (Pinedo, 2002) and Jordan (Jordan, 1996). In Varela (2007) a comparative analysis of these nomenclatures is presented.

Thus, following the Graham's notation α,β,γ, we propose a new field δ described in this paper as the required sub-problem solutions class. By the required "sub-problem solution class" we consider that the general solution should contain solutions for sub-problems such as dispatching rule, lot splitting rule, and a type of a heuristic algorithm to be used. This class was introduced for the purpose of guiding the simulation process. In other words this notation, with proposed field δ and corresponded parameters, serves to specify the parameters of the simulation process of the "general purpose" scheduling simulator (Putnik et al., 2014).

As our goal it is not to deepen through all notation's parameters but to identify the notation's parameters which can be simulated by scheduling simulator which framework is presented in Putnik et al. (2014), the considered notation's parameters of $\alpha,\beta,\gamma,\delta$ are presented in Table 1.

Table 1. – Notation's Parameters $\alpha, \beta, \gamma, \delta$

	Parameter	(Brucker, 1995)	Designation
α (system environment)	α_1	System Type	System Type
	α_2	System size (number of processors or machines)	System size (number of processors or machines)
β (characteristics, restrictions, conditions of: jobs (β_1), machines/resources (β_2) and combined β_1 and β_2 (β_3))	β_1	Job preemption	Number of lots; Number of parts per lot; Size of lots; Lot divisible (Lot splitting); Process plan for all lots; Process plan per part in one lot; Lot readiness/availability (ready time); Lot preference; Job preemption
	β_2	Precedence relations	Setup (Variable or fixed) Resource availability
	β_3	Release dates	Combined β_1 and β_2
	β_4	Restrictions on processing time or on the number of operations	n.d.

	β_5	Deadline	n.d.
	β_6	Batching problem	n.d.
γ (performance measures): γ_1 – single objective; γ_2 – multi-objective	γ_1	Optimality Criteria	Time-to-Market Cost CO2 emissions
	γ_2		Combined single objectives from γ_1
δ – Sub-problem solution classes		n.d.	Dispatching rule Lot Splitting Rule Algorithm Class (Neural, Dynamic, Genetic, …)

Because of the combinatorial explosion and limitation of space, in the context of this paper, and for the contribution for generalization of scheduling problem classes, we focuses only on parameter β_1, considering the number of lots, number of parts per lot, size of lots, lot divisible (lot splitting), process plan for all lots, and process plan per part in one lot. These variable parameters were combined following their values (other parameters will be considered as constant in this work), such as:

- Number of lots: $\{1, m\}$;
- Number of parts per lot: $\{1, n\}$;
- Size of lots: $\{Equal, Different\}$;
- Lot Divisible (Lot splitting): $\{Yes, No\}$;
- Process plan for all lots: $\{Equal, Different\}$;
- Process Plan per part in one lot: $\{Equal, Different\}$;

These 6 parameters makes 64 (2^6=64) possible combinations, instantiations of the scheduling problem.

Table 2 and Table 3 resume the unfeasible and feasible classes, respectively, from the 64 possible combinations of the scheduling parameter values. The naming of the Classes that is the order number of each class is given by the authors (in accordance with the combinations generation applied).

Classes 1-2, 4-8 and 33-40 (Table 2) are impossible to realize. These classes consider only one lot, with the size of 1 (one part) only. From this reason there can't be different size of lots, lot division, different process plan for lots and/or different process plans per part belonging to one lot (the lot consist of only one part).

Considering the combinations for the parameters: number of lots equal to 1 and number of parts per lot equal to n n, as we have only one lot, the classes that consider different size of lots or different process plans for all lots, are not feasible as there is only one lot. Therefore the Classes 10, 12-16, 42 and 44-48 (Table 2) are not feasible under this criteria.

Considering the combinations for the parameters: number of lot equal to m and number of parts per lot equal to 1, which contains the different size of lots, lots divisible and different process plan per part in one lot, are not feasible, namely the Classes 18, 21-24 and 49-56 (Table 2).

Finally, all the Classes for the combination with the parameters: number of lot equal to m and number of parts per lot equal to n are feasible, i.e., Classes 25-32 and 57-64 (Table 3).

Table 2. – Unfeasible classes under the 64 possible combinations/class problem

Class	Number of lots	Number of parts per lot	Size of lots	Lot Divisible	Process Plan for all lots	Process Plan per part in one lot
1	1	1	Equal	Yes	Equal	Equal
2	1	1	Equal	Yes	Different	Equal
4	1	1	Equal	No	Different	Equal
5	1	1	Different	Yes	Equal	Equal
6	1	1	Different	Yes	Different	Equal
7	1	1	Different	No	Equal	Equal
8	1	1	Different	No	Different	Equal
10	1	n	Equal	Yes	Different	Equal
12	1	n	Equal	No	Different	Equal
13	1	n	Different	Yes	Equal	Equal
14	1	n	Different	Yes	Different	Equal
15	1	n	Different	No	Equal	Equal
16	1	n	Different	No	Different	Equal
18	m	1	Equal	Yes	Different	Equal
21	m	1	Different	Yes	Equal	Equal
22	m	1	Different	Yes	Different	Equal
23	m	1	Different	No	Equal	Equal
24	m	1	Different	No	Different	Equal
33	1	1	Equal	Yes	Equal	Different
34	1	1	Equal	Yes	Different	Different
35	1	1	Equal	No	Equal	Different
36	1	1	Equal	No	Different	Different
37	1	1	Different	Yes	Equal	Different
38	1	1	Different	Yes	Different	Different
39	1	1	Different	No	Equal	Different
40	1	1	Different	No	Different	Different
42	1	n	Equal	Yes	Different	Different
44	1	n	Equal	No	Different	Different
45	1	n	Different	Yes	Equal	Different
46	1	n	Different	Yes	Different	Different
47	1	n	Different	No	Equal	Different
48	1	n	Different	No	Different	Different
49	m	1	Equal	Yes	Equal	Different
50	m	1	Equal	Yes	Different	Different
51	m	1	Equal	No	Equal	Different
52	m	1	Equal	No	Different	Different

53	m	1	Different	Yes	Equal	Different
54	m	1	Different	Yes	Different	Different
55	m	1	Different	No	Equal	Different
56	m	1	Different	No	Different	Different

Table 3. – Feasible classes under the 64 possible combinations/class problem

Class	Number of lots	Number of parts per lot	Size of lots	Lot Divisible	Process Plan for all lots	Process Plan per part in one lot
3	1	1	Equal	No	Equal	Equal
9	1	n	Equal	Yes	Equal	Equal
11	1	n	Equal	No	Equal	Equal
17	m	1	Equal	Yes	Equal	Equal
19	m	1	Equal	No	Equal	Equal
20	m	1	Equal	No	Different	Equal
25	m	n	Equal	Yes	Equal	Equal
26	m	n	Equal	Yes	Different	Equal
27	m	n	Equal	No	Equal	Equal
28	m	n	Equal	No	Different	Equal
29	m	n	Different	Yes	Equal	Equal
30	m	n	Different	Yes	Different	Equal
31	m	n	Different	No	Equal	Equal
32	m	n	Different	No	Different	Equal
41	1	n	Equal	Yes	Equal	Different
43	1	n	Equal	No	Equal	Different
57	m	n	Equal	Yes	Equal	Different
58	m	n	Equal	Yes	Different	Different
59	m	n	Equal	No	Equal	Different
60	m	n	Equal	No	Different	Different
61	m	n	Different	Yes	Equal	Different
62	m	n	Different	Yes	Different	Different
63	m	n	Different	No	Equal	Different
64	m	n	Different	No	Different	Different

3. Generalizing scheduling problem classes

Scheduling problems classes were reduced from 64 to 24 classes, as presented in the previous chapter (Table 3). These 24 classes can be also reduced into several scheduling problems considering the combinations of the different parameters.

Table 4 presents the description of the 16 problems, and the classes associated to them (in grey). As number of lots $m \in \mathbb{N}$ and number of parts per lot $n \in \mathbb{N}$, Classes 3-19, 9-17, 20, 43 and 41 are reduced into Classes 27, 25, 28, 59 and 57 respectively.

Table 4. – Generalizing particular scheduling problem

Problem	Class	Number of lots	Number of parts per lot	Size of lots	Lot Divisible	Process Plan for all lots	Process Plan per part in one lot
1 - Equal size of lots, lots indivisible, equal process plan for all lots and equal process plan per part in one lot	3	1	1	Equal	No	Equal	Equal
	11	1	n	Equal	No	Equal	Equal
	19	m	1	Equal	No	Equal	Equal
	27	m	n	Equal	No	Equal	Equal
2 - Equal size of lots, lots divisible, equal process plan for all lots and equal process plan per part in one lot	9	1	n	Equal	Yes	Equal	Equal
	17	m	1	Equal	Yes	Equal	Equal
	25	m	n	Equal	Yes	Equal	Equal
3 - Equal size of lots, lots indivisible, different process plan for each lot and equal process plan per part in one lot	20	m	1	Equal	No	Different	Equal
	28	m	n	Equal	No	Different	Equal
4 - Equal size of lots, lots divisible, different process plan for each lot and equal process plan per part in one lot	26	m	n	Equal	Yes	Different	Equal
5 - Different size of lots, lots indivisible, equal process plan for all lots and equal process plan per part in one lot	31	m	n	Different	No	Equal	Equal
6 - Different size of lots, lots divisible, equal process plan for all lots and equal process plan per part in one lot	29	m	n	Different	Yes	Equal	Equal
7 -Different size of lots, lots indivisible, different process plan for each lot and equal process plan per part in one lot	32	m	n	Different	No	Different	Equal
8 - Different size of lots, lots divisible, different process plan for each lot and equal process plan per part in one lot	30	m	n	Different	Yes	Different	Equal
9 - Equal size of lots, lots indivisible, equal process plan for all lots and different process plan per part in one lot	43	1	n	Equal	No	Equal	Different
	59	m	n	Equal	No	Equal	Different
10 - Equal size of lots, lots divisible, equal process plan for all lots and different process plan per part in one lot	41	1	n	Equal	Yes	Equal	Different
	57	m	n	Equal	Yes	Equal	Different
11 - Equal size of lots, lots indivisible, different process plan for each lot and different process plan per part in one lot	60	m	n	Equal	No	Different	Different
12 - Equal size of lots, lots divisible, different process plan for each lot and different process plan per part in one lot	58	m	n	Equal	Yes	Different	Different
13 - Different size of lots, lots indivisible, equal process plan for all lots and different process plan per part in one lot	63	m	n	Different	No	Equal	Different
14 - Different size of lots, lots divisible, equal process plan for all lots and different process plan per part in one lot	61	m	n	Different	Yes	Equal	Different

15 -Different size of lots, lots indivisible, different process plan for each lot and different process plan per part in one lot	64	m	n	Different	No	Different	Different
16 - Different size of lots, lots divisible, different process plan for each lot and different process plan per part in one lot	62	m	n	Different	Yes	Different	Different

These 16 problems represent the problems that could be resolved by one or more algorithms. However, it is virtually unrealistic to create problem classes for all different scheduling parameters combinations, once the number of combinations for *n* parameters is 2^n, or even more if considering the values in parameter description (in this paper we combine only 6 parameters giving 64 combinations – as we have seen in the previous section).

Thus, we breakdown these different problems in accordance with their input for the algorithm. In this paper, only the breakdown for the Problem 16 is presented (Table 5), as it encloses all other classes as it includes all the parameters. In this case, the lots are indivisible, the size of lots is considered different, as well as the process plan for each lot and the different process plan per part in one lot. So, 2 lots are presented with the size of 3 and 2 parts, respectively for the lot 1 and 2, and with 3 operations for both lots. Table 5 aims to help the understanding of the problem generalization.

Table 5. – Input considering the problem 16

Size of Lots	Lot Divisible	Process Plan for all lots	Process Plan per part in one lot
Lot 1 (3 Parts {A,A,A}	Lot (sublot) 1.1 A	Process Plan 1 = {op1, op2, op3}	Process Plan 1 (Lot 1, Part A) ={op1, op2, op3}
	Lot (sublot) 1.2 A	Process Plan 1 = {op1, op2, op3}	Process Plan 2 (Lot 1, Part A) ={op4, op5, op6}
	Lot (sublot) 1.3 A	Process Plan 1 = {op1, op2, op3}	Process Plan 3 (Lot 1, Part A) ={op7, op8, op9}
Lot 2 (2 Parts {A,A})	Lot (sublot) 2.1 A	Process Plan 2 = {op4, op5, op6}	Process Plan 4 (Lot 1, Part A) ={op10, op11, op12}
	Lot (sublot) 2.2 A	Process Plan 2 = {op4, op5, op6}	Process Plan 5 (Lot 1, Part A) ={op13, op14, op15}

Generalizing the intrinsic nature of the parameter's values, it is possible to reduce all the 16 problems into 1. This means that in the input will be composed by a matrix $i \times j$, where i is determined by the total number of parts of all lots, and j corresponds to the variable parameters. For example, in the problem 16 it is considered 5 lines of input and 3 columns.

Considering Table 4 and Table 5, it can be assumed that all problems can be built by primitive blocks. For example, problem 16 (considering the input values - Table 5) can be generalized into 5 lines composed by primitive blocks of problem 1, Class 3. So, all problems described can be represented through the elementary Building Block – Class 3 , , i.e. equal size of lots (size of lot equal to 1), number of parts per lot (lot is composed by 1 part), lots indivisible, equal process plan for all lots and equal process plan per part in one lot , represented in Table 6.

Table 6. – Generalized scheduling problem class

Problem	Class	Number of lots	Number of parts per lot	Size of lots	Lot Divisible	Process Plan for all lots	Process Plan per part in one lot
1 - Equal size of lots, lots indivisible, equal process plan for all lots and equal process plan per part in one lot	3	1	1	Equal	No	Equal	Equal

4. Conclusions

Simulators development for scheduling problem classes could be very complexity. Some authors dedicate their research to find and develop simulator for the resolution of the scheduling problem classes. However, in this

paper we didn't go to deepen through all notation's parameters but to identify the notation's parameters which can be simulated by scheduling simulator to be developed.

Proposed notation was introduced to specify the parameters of the simulation process of the "general purpose" scheduling simulator: $\beta_1 = \{1,1,Equal,No,Equal,Equal,\varnothing\}$ and $\delta = \{FIFO,\varnothing,\text{Selection\_algorithm}\}$. Other parameters values will be considered fixed, as applying the same methodology all values and parameters can be represented by the elementary building block.

A contribution for generalization of scheduling problem classes to minimize a simulator's size/complexity was given generalizing different scheduling problems into one problem, i.e., problems can be constituted by primitive "blocks" concerning Problem 1 – Class 3. Simulator open architecture already used in CIM (Computer Integrated Manufacturing) could be considered to represent any problem through the elementary Building Block, i.e. represent any problem through the Class 3 (Figure 1). Considering the terms used in CIM's jargon, Compiler 1 (Compiler Class n to Class 3) and Compiler 2 (Compiler Class 3 to Class n) presented in Figure 1 are the "pre-processor" and "post-processor of the simulator.

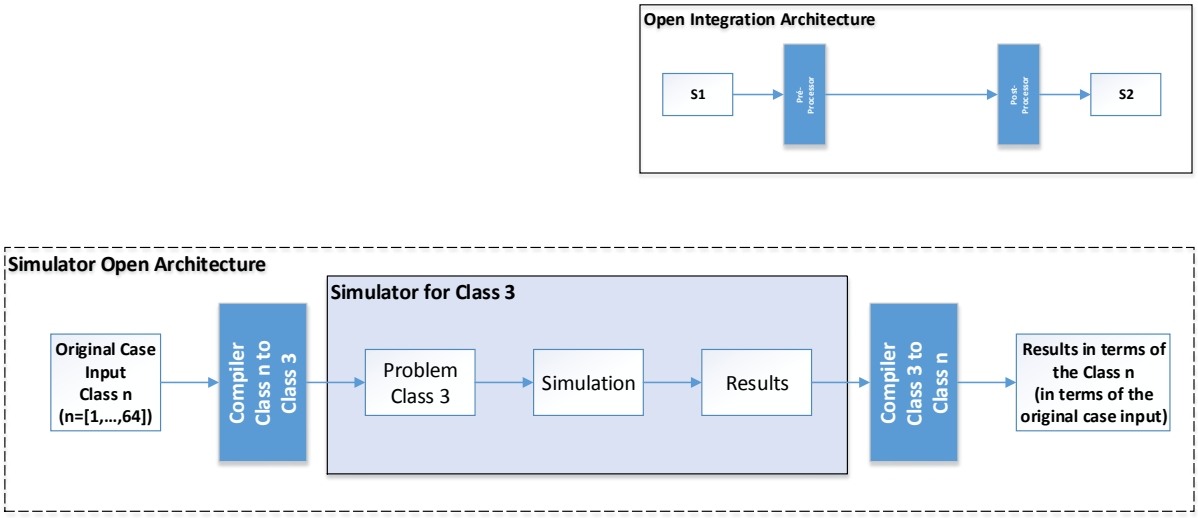

Figure 1. Simulator open architecture for representation of scheduling problem input/output transformation

Further research will focuses in the proof that all scheduling models can be represented in architectures based on Fixed Horizon paradigm as an elementary building block.

Acknowledgments

The authors wish to acknowledge the support of the Fundação para a Ciência e Tecnologia (FCT), Portugal, within the Project Reference UID/CEC/00319/2013 and the "Ph.D. Scholarship Grant" reference SFRH/BD/85672/2012.

References

Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., & Weglarz, J. (1997). Scheduling computer and manufacturing processes. *Journal of the Operational Research Society, 48*(6), 659-659.

Brucker, P. (1995). Scheduling Algorithms: Germany: Springer-Verlag.

Conway, R. W., Maxwell, W. L., & Miller, L. W. (2012). *Theory of scheduling*: Courier Corporation.

Emmons, H. (1987). Scheduling and sequencing algorithms. *Production Handbook, 4th ed., John Wiley & Sons, New York*.

French, S. (1982). *Sequencing and scheduling: an introduction to the mathematics of the job-shop* (Vol. 683): Ellis Horwood Chichester.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics, 5*, 287-326.

Jordan, C. (1996). Batching and scheduling. *Models and Methods for Several*.

Pinedo, M. (2002). Scheduling: theory, algorithms, and systems: Prentice Hall.

Putnik, G., Alves, C., Ávila, P., Ferreira, L., Castro, H., & Shah, V. (2014, November 5-7, 2014.). *A Framework for simulator development for Fixed Horizon, Rolling Horizon and Real Time Management Modelling and Evaluation* Paper presented at the Fourth International Conference on Business Sustainability 2014 - Management, Technology and Learning for Individuals, Organisations and Society in Turbulent Environments, Póvoa de Varzim.

Stecca, G. (2014). Scheduling *CIRP Encyclopedia of Production Engineering* (pp. 1092-1095): Springer.

Varela, M. L. R. (2007). *Uma contribuição para o escalonamento da produção baseado em métodos globalmente distribuídos.* (Doctoral Thesis), University of Minho. Retrieved from http://hdl.handle.net/1822/7234