



Use of excel worksheets with user-friendly interface to minimize the makespan in a permutation flow shop production system

Rony Peterson da Rocha^{1*}, Mauro Antônio Silva Sá Ravagnani², Cid Marcos Gonçalves Andrade² and Paulo Roberto Paraíso²

¹Departamento de Engenharia de Produção, Universidade Estadual do Paraná, Av. Comendador Norberto Marcondes, 733, 87302-060, Campo Mourão, Paraná, Brazil. ²Departamento de Engenharia Química, Universidade Estadual de Maringá, Maringá, Paraná, Brazil. *Author for correspondence. E-mail: petersonccbpr@hotmail.com

ABSTRACT. Current research addresses the Permutation Flow Shop scheduling problem, characterized as Production Scheduling in Batch Process (PSBP) to minimize total time to complete the schedule (Makespan). A method to approach the problem of production scheduling is to turn it into Mixed Integer Linear Programming (MILP) and to solve it by using commercial mathematical programming packages. An electronic spreadsheet with user-friendly interface (ESUFI) was developed in Microsoft Excel. Results showed that it is possible to use ESUFI to solve small problems.

Keywords: excel, linear program, permutation flow shop.

Utilização de planilhas do excel com interface amigável para minimização do makespan em um sistema de produção flow shop permutacional

RESUMO. Este trabalho dedica-se a um problema de programação em *Flow Shop* Permutacional, caracterizado como Programação da Produção em Processo Batelada (PPPB), com o objetivo de minimizar a duração total da programação (*Makespan*). Um método para a abordagem do problema de programação da produção é tornar o problema como de Programação Linear Inteira Mista (*Mixed Integer Linear Program – MILP*) e resolvê-lo usando pacotes de programação matemática. Neste estudo foi criada uma planilha eletrônica do *Microsoft Excel* com interface amigável (PEIA). Os resultados mostraram que é possível utilizar a PEIA em problemas de pequeno porte.

Palavras-chave: excel, programação linear, *flow shop* permutacional.

Introduction

Nowadays the planning of industrial activities is a necessity for their survival. Intense competition requires diversified products and delivery according to consumers' requirements. These activities require quick decision-making and lowest possible cost.

The need for scheduling in the chemical industry is becoming more mandatory especially in the batch production mode. According to Reklaitis (1985), batch production may be described as income-oriented production, with equipment network connectivity and resource limitations. Mendéz et al. (2006) classifies this type of operation as single stage, subdivided into single or parallel resources, and multiple stages, subdivided into multipurpose (Job Shop) and multiproduct (Flow Shop).

The diversity and complexity of a batch production industry recommend the application of production planning techniques to meet consumer demands. Thus, scientific development in the area of Production Scheduling is increasing, especially

with regard to batch processes. Scheduling is defined, according to Baker (1974), as the allocation of resources over time to perform a collection of tasks.

The scheduling problem addressed in current research is of the batch type - multiproduct - Permutation Flow Shop. According to MacCarthy and Liu (1993) and Baker (1974), Flow Shop is a type of process where all tasks have a similar flow pattern, that is, they have the same processing schedule on all resources and the number of resources in each stage of production equals one.

The environment of the permutation Flow Shop production scheduling problem is described by MacCarthy and Liu (1993), Baker (1974), Taillard (1993), and Moccellini (1995) as a type of Flow Shop in which the processing order of tasks is the same in all resources.

Batch-Multiproduct plants are in general employed for a set of products whose income structure is the same and production lines are also referred to as Flow Shop. States many chemical processing industries, such

as oil and paint, pharmaceutical and fine chemistry industries, fit into this category.

Sundaramoorthy and Karimi (2005) reported on the production scheduling problem in the short term for plants in multipurpose batches. The above study provided one formulation based on mixed integer linear programming (MILP), using a continuous representation of synchronous time slots and a new idea of multiple balances (time, mass, resources, etc.). The model works without big-M constraints and is equally effective to maximize profits and minimize makespan.

Burkard and Hatzl (2006) researched the problem of production scheduling in the chemical industry. The objective of the study was to implement an objective function to minimize the makespan by the employment of heuristics. Their study proposed an iterative algorithm construction that alternated between phases of construction and deconstruction. Strategies for diversification and intensification were also suggested to obtain optimal solutions in good moderate running times. Computational results showed the power of this algorithm.

Shakeri and Logendran (2007) studied the problem of production scheduling model as a mixed integer linear programming (binary) developed for scheduling tasks in multitasking environments, for which the number of completed tasks was not a good measure. The authors fixed issues for small, medium and large, with the aid of a tabu search algorithm. The solution obtained from the algorithm was compared with that of the optimal solution or the upper bound found by using Lagrangian relaxation.

Pan et al. (2008) describe a study for short-term scheduling of multipurpose batch plants using Mixed Integer Linear Programming. The research presented the network states and tasks (STN) to eliminate the inconvenience of precedence-based formulations which typically include a large number of batches. Rules have been proposed in the study heuristics for solving the problem. The results were effective to find the best solution to problems.

Lin and Liao (2012) presented a paper about a scheduling problem for a two-stage assembly shop in a machinery factory. Its aim was to minimize the weighted sum of makespan, total completion time and total tardiness. A Mixed Integer Programming (MIP) model was developed for solving small-size problems and three heuristics were proposed for solving medium- and large-size problems.

Koné et al. (2013) have studied scheduling problem that takes into account storage resources which may be produced or consumed by activities. The role model was elaborated in the Mixed Integer Linear Programming.

Fumero et al. (2013) also conducted a study on Mixed Integer Linear Programming. A model was performed for both design and scheduling of flow shop batch plants taking into account mixed product campaign and parallel unit duplication. Results showed that a realistic formulation was attained and industrial and commercial aspects were jointly taken into account.

Current research provides the development of a spreadsheet by the employment of Visual Basic (VB), with an interface with the user, whose goal is to minimize the task completion time (Makespan) of a Permutation Flow Shop Scheduling environment. Investigation is embedded in the area of Synthesis and Optimization of Processes of the Chemical Engineering Graduate Program at UEM – State University of Maringá, in partnership with GEPPGO - Group for Study and Research on Process and Operations Management from the Production Engineering Department of the State University of Paraná - Campus Campo Mourão.

Material and methods

For the development of a user-friendly interface to solve a production scheduling model for the permutation Flow Shop system, the paper was divided into five parts: Model development, Model formulation, Model resolution in Excel spreadsheets of Microsoft, Creation of interface between user and Worksheet in Visual Basic Applications language (VBA) of Microsoft's Visual Basic incorporated in all Microsoft Office programs and Evaluation of the consistency of the interface.

The problem proposed with regard to a multiproduct batch plant, characterized as a permutation Flow Shop environment, was analyzed for the model development. At this stage, the objective function was defined whilst considering the case of production scheduling as a mixed integer linear programming problem (MILP).

In this permutation Flow Shop programming environment, three research scenarios were modelled, all of them to minimize total task completion time (Makespan).

The model constraints are: a) the storage of intermediate products should not be available between the processing resources, that is, if a product is processed at resource j and resource $j+1$ is not available at the time of completion, the finished product should be kept at resource j , until resource $j+1$ is ready; b) after finishing the processing of a product at the last resource (equipment), this product is immediately sent to the stock of finished products; c) all resources are initially empty at time zero and the manufacture of any

product may be delayed at an arbitrary amount of time to keep it in the previous resource; d) the ordering of tasks on each resource is the same.

After finishing the modelling, the mathematical equations of all three scenarios were manually transcribed into Excel spreadsheets and solved by the Add-in Solver.

VB language was used to create an interface between the user and the Excel spreadsheet. Consequently, all mathematical equations modeled on the three scenarios, along with the Add-in Solver, were developed to create a generalized programming model for this environment, with the restrictions imposed by the proposed model.

The evaluation of the consistency of the spreadsheet interface with the user was done by comparing the results manually obtained with Excel spreadsheets with those obtained VBA language.

User-friendly interface for the resolution of a model of production scheduling in a permutation flow shop system with makespan reduction

Model development

Table 1 shows a production scenario with four tasks (T1, T2, T3, T4) processed in three features in series.

Table 1. Processing time (h).

Resources	Tasks			
	t1	t2	t3	t4
1	3.5	4.0	3.5	12
2	4.3	5.5	7.5	3.5
3	8.0	3.5	6.0	8.0

Table 1 shows that the first task is first processed in machine (Resources) 1 in 3.5h; machine (Resources) 2 in 4.3h; machine (Resources) 3 in 8h. The processing order of tasks 1, 2, 3 and 4 may be any combination of the four tasks but it must comply with the script on the machine first set 1, 2 and 3.

Model formulation

The mathematical modeling of the scenario in Table 1 is shown by the equations provided in this section where Equation 1 represents the main goal of the problem to be solved.

$$\text{Minimize } C_{nm} \quad (1)$$

Variable C in Equation 1 represents the Makespan, while N and M represent respectively the number of tasks and the number of resources. Thus, the objective function expressed by Equation 1 comprises finding, among the various combinations

of tasks and resources, the NM sequence that provides the lowest Makespan. For scenario 1, $N = 4$ and $M = 3$ in the objective function. This objective function is subject to a number of constraints, as described below. As Equations 2 and 3 show, the first constraints are binary.

$$\sum_i X_{ik} = 1 \quad \forall k \quad (2)$$

$$\sum_k X_{ik} = 1 \quad \forall i \quad (3)$$

Variable i represents the task to be processed at the resource and k represents the position of this task in the order of sequencing. Equations 2 and 3 thus represent a discrete optimization problem involving the decision between two alternatives, that is, if a task is processed at a given resource, the other tasks cannot be processed concurrently at the same resource. Therefore, X_{ik} is a binary variable defined as: $X_{ik} = 1$, if task i is in position k , and $X_{ik} = 0$, otherwise. Each task is sequentially processed by resources 1 (machine - 1), 2 (machine - 2) and 3 (machine - 3), respecting the permutation Flow Shop programming environment.

Equation 4 represents another problem constraint which also involves Makespan. In this equation, C_{kj} is the time of end of processing, by resource j , of the task occupying position k in the sequence; N is the number of tasks; $X_{s,k}$ is the binary variable, and $TP_{s,j}$ is the time for processing task i at resource j .

$$C_{k,j} \geq C_{k-1,j} + \sum_{s=1}^N X_{s,k} TP_{s,j} \quad \forall j; k = 2, \dots, N \quad (4)$$

Generally, the solution of the various alternatives of Equation 4 shows that, for a scheduled task to be processed in resource j , from the second order of sequencing, it must present a Makespan greater than or equal to the Makespan of a task placed earlier on the same resource j , plus the processing time of the same task on resource j .

Another important constraint is Equation 5 which also involves the Makespan of the tasks in the resources. Equation 5 shows that the Makespan of the task of order k on resource j is greater than or equal to the Makespan of the same task on resource $j-1$ plus the sum of the products of the binary variable $X_{s,k}$ and the processing times $TP_{s,j}$, starting with resource 2 (machine - 2) and finishing with resource 3 (machine - 3).

$$C_{k,j} \geq C_{k,j-1} + \sum_{s=1}^N X_{s,k} TP_{s,j} \quad \forall j; j = 2, \dots, M \quad (5)$$

The Makespan of the task placed in sequence 1 of the production scheduling, at resource j , is expressed in Equation 6. Equation 6 provides the constraint that the Makespan of such task is greater than or equal to the Makespan of the task placed in sequence 1 of the production scheduling, at resource $j-1$, plus the sum of the products of the binary variable $X_{s,1}$ and the processing times $TP_{s,j}$. In this equation, resource j must start with resource 2 (machine – 2) and finish with resource 3 (machine – 3), according to the model initially proposed.

$$C_{1,j} \geq C_{1,j-1} + \sum_{s=1}^N X_{s,1} TP_{s,j} \quad \forall j; j = 2, \dots, M \quad (6)$$

Another important constraint is the Makespan of the task sequenced with order 1 on resource 1 (machine – 1), represented by Equation 7 which shows that the Makespan of this task must be greater than or equal to the sum of the products of the binary variable ($X_{1,1}$, $X_{2,1}$, $X_{3,1}$, $X_{4,1}$) and the processing times ($TP_{1,1}$, $TP_{2,1}$, $TP_{3,1}$, $TP_{4,1}$).

$$C_{1,1} \geq \sum_{s=1}^N X_{s,1} TP_{s,j} \quad (7)$$

Constraint in Equation 8 shows that the Makespan sequenced in the order k on resource j must be greater than or equal to the Makespan sequenced in the order $k-1$ on resource $j+1$, starting from the order of sequence 1 on resource 1 (machine – 1).

$$C_{k,j} \geq C_{k-1,j+1} \quad k = 1, \dots, N \quad j = 1, \dots, M-1 \quad (8)$$

Finally, the last constraint of the problem concerns the question of non-negativity of the model, which is represented by the general Equation 9.

$$C_{k,j} \geq 0 \quad \forall k \quad \forall j \quad (9)$$

Model resolution in excel spreadsheets

Equations (1) through (9) of the four tasks and three resource models were inserted in the worksheet, in different cells. Further, the Add-in Solver, available within the Microsoft Excel electronic spreadsheet, was used to optimize the scheduling of the tasks on the resources.

In general, so that the manual resolution of the processing orders scheduling model proposed in this study could be developed, all available data were first organized on an Excel spreadsheet,

separating the cells representing the decision variables and the objectivefunction. For each problem constraint, a formula was developed in a separate cell of the spreadsheet, corresponding to the left-hand side (LHS) and the right-hand side (RHS) of the restriction.

LHS and RHS correspond to the transformation of the set of constraints into a set of equivalent equations, by introducing variables that represent the gap between the left (LHS) and right (RHS) sides of the inequalities, as shown in Figure 1.

As observed in Figure 1, the binary equations: B18 + B22 + B26 + B30; B19 + B23 + B27 + B31; B20 + B24 + B28 + B32; B21 + B25 + B29 + B33; B18 + B19 + B20 + B21; B22 + B23 + B24 + B25; B26 + B27 + B28 + B29, and B30 + B31 + B32 + B33 were inserted in cells F6 to F14, respectively. Cells H6 to H14 were assigned the value of 1. These equations represented the resolution of the general Equations 1 and 2.

The cells: B9, B10, B11, B12, B13, B14, B15, B16, and B17 were inserted in the respective cells F16 to F24. These cells are larger than or equal to the cells H16, H17, H18, H19, H20, H21, H22, H23, and H24, and correspond to the respective equations:

B6 + (B19 * D6 + B23 * D9 + B27 * D12 + B31 * D15); B7 + (B19 * D7 + B23 * D10 + B27 * D13 + B31 * D16); B8 + (B19 * D8 + B23 * D11 + B27 * D14 + B31 * D17); B9 + (B20 * D6 + B24 * D9 + B28 * D12 + B32 * D15); B10 + (B20 * D7 + B24 * D10 + B28 * D13 + B32 * D16); B11 + (B20 * D8 + B24 * D11 + B28 * D14 + B32 * D17); B12 + (B21 * D6 + B25 * D9 + B29 * D12 + B33 * D15); B13 + (B21 * D7 + B25 * D10 + B29 * D13 + B33 * D16), and B14 + (B21 * D8 + B25 * D11 + B29 * D14 + B33 * D17).

The cells: B7, B8, B10, B11, B13, B14, B16, and B17 were inserted in the respective cells J6 to J13. These cells are larger than or equal to the cells L6, L7, L8, L9, L10, L11, L12, and L13, and correspond to the respective equations: B6 + (B18 * D7 + B22 * D10 + B26 * D13 + B30 * D16); B7 + (B18 * D8 + B22 * D11 + B26 * D14 + B30 * D17); B9 + (B19 * D7 + B23 * D10 + B27 * D13 + B31 * D16); B10 + (B19 * D8 + B23 * D11 + B27 * D14 + B31 * D17); B12 + (B20 * D7 + B24 * D10 + B28 * D13 + B32 * D16); B13 + (B20 * D8 + B24 * D11 + B28 * D14 + B32 * D17); B15 + (B21 * D7 + B25 * D10 + B29 * D13 + B33 * D16), and B16 + (B21 * D8 + B25 * D11 + B29 * D14 + B33 * D17).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
3																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																

Figure 1. Excel electronic spreadsheet for the PSBP model.

Cells J15 and J16 are equal to cells B7 and B8 and are larger than or equal to cells L15 and L16, which correspond to equations $B6 + (B18 * D7 + B22 * D10 + B26 * D13 + B30 * D16)$ and $B7 + (B18 * D8 + B22 * D11 + B26 * D14 + B30 * D17)$, respectively.

Cell J18 is equal to cell L18 and corresponds to the equation $(B18 * D6 + B22 * D9 + B26 * D12 + B30 * D15)$.

Cells N6 to N11 are equal to cells B9, B10, B12, B13, B15, and B16, and are larger than or equal to cells P6 to P11, which in turn are equal to cells B7, B8, B10, B11, B13, and B14. Cells N13 to N24 are equal to cells B6 to B17, and are equal to or larger than cells P13 to P24, which in turn are equal to zero.

After inserting all the equations of the PSBP model, representing the objective function, decision variables and restrictions, in the Microsoft Excel electronic spreadsheet shown in Figure 1, the Add-in Solver, available in the tool bar of this spreadsheet, was used to find the optimal solution for the problem.

The steps required to use Solver are: Click the Office button, Excel Options, which opens a dialog box, where the user must click on 'Add-ins', which displays the options of the selected menu, with various add-ins. The user should then select 'Solver'. In the selection box 'Manage', select 'Excel Add-ins' and click the 'Go' button to enable the Microsoft Excel Solver. After that, it is necessary to choose the add-in Solver in the new dialog box that is opened. After enabling Solver, the user must select cell D3 of the Microsoft Excel spreadsheet, shown in Figure 1. The user must then click on 'Data' and 'Solver', as provided in the toolbar. This will open a data window, called 'Solver Parameters'.

It is necessary to indicate in the Solver Parameters box which cells represent the objective function, the decision variables and the problem's constraints. Thus, in the location 'Set Target Cell', the user must choose the cell that represents the objective function and in 'Equal to' choose minimization, since the objective function of the problem is to minimize the total time of task completion. 'By changing cells' represent the cells of

the decision variables of the problem, and 'Subject to the constraints' represent the LHS and RHS of the constraints.

After all variables of the model have been inserted in the Solver dialog box, the model is asked to solve the objective function. Prior to that, one should choose options and adjust some parameters. In a specific case, a maximum time of 100 seconds was set, with a number of iterations 1000, precision 0.000001, tolerance 5%, and convergence 0.001, assuming a non-negative linear model, tangent estimates, forward derivatives, and Newton search. These parameters presented were not modified, or rather, they are available and set by the Excel Solver.

After the parameters in the 'Solver Options' box have been defined, one must choose the option 'OK', which will open a new dialog box with an option to choose the Solver reports. In this case, the report 'Answer' is chosen.

After performing all the steps above, the add-in Solver, chosen to perform the optimization of the Makespan, that is, the minimization of the objective function of the problem, and therefore the reduction of the total time for the completion of tasks, will solve all the iterations until the optimal solution is reached, shown in cell D3 in Figure 1 in the Excel spreadsheet, at the value of 35 hours.

Creation of the friendly interface between the user and the excel spreadsheet in the language Visual Basic For Applications (Vba)

For the creation of the user-friendly interface with the Excel spreadsheet (ESUFI) used to solve the model to reduce the Makespan following the

constraints set forth in the methodology, it was necessary to first define a standard spreadsheet. The standard spreadsheet refers to a Microsoft Excel spreadsheet, composed of all the equations described in the study, organized in a standard model. This means that each equation should be placed in an array of rows and columns, available to perform the generalization of the model. The generalization of this model was performed by the VBA programming language.

The implementation of VBA code in the spreadsheet made possible the generalization of the model of four tasks and three resources for the configuration of (n) tasks and (m) resources. A friendly interface between the user and the Microsoft Excel spreadsheet was thus created and the user should only report the number of tasks (n) and resources (m) and the processing time of each task at the respective resource. The user may then click on the button 'Optimize Scheduling Order' to obtain the best production sequence that minimizes the Makespan. Figure 2 exemplifies the model of the Microsoft Excel spreadsheet with the user-friendly interface.

To build the user-friendly interface created for ESUFI provided in Figure 2, it was necessary to carry out some validation tests for each of the model equations. These tests serve to validate the model, or rather, to show the veracity of each of the equations from the expansion of the four-task and three-resource model for different situations of (N) tasks and (M) resources. The tests also served to find a number of possible combinations of (N) and (M) to be applied by the Solver tool.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y																					
1	PRODUCTION SCHEDULING IN PERMUTATION FLOW SHOP ENVIRONMENT																																													
2	Operator name		Rony Peterson da Rocha																																											
3																																														
4																																														
5	Number of tasks		2		Generate processing time										Optimize scheduling of orders																															
6	Number of resources		120																																											
7	Optimized sequence																																													
8	Total time of task completion		0																																											
9																																														
10	3,12																																													
11																																														
12	Equations																																													
13	Variables										Equations																																			
14											EQ2 =			EQ3 =			EQ4 >=			EQ5 >=			EQ6 >=			EQ7 =			EQ8 >=			EQ9 >=														
15											LHS			RHS			LHS			RHS			LHS			RHS			LHR			RHS			LHS			RHS			LHS			RHS		
16	Produto1	Processador1	C.1,1	0	TP 1,1						p1	s1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
17	Produto1	Processador2	C.1,2	0	TP 1,2						p1	s2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
18	Produto1	Processador3	C.1,3	0	TP 1,3						p2	s1	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
19	Produto1	Processador4	C.1,4	0	TP 1,4						p2	s2	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
20	Produto1	Processador5	C.1,5	0	TP 1,5													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
21	Produto1	Processador6	C.1,6	0	TP 1,6													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
22	Produto1	Processador7	C.1,7	0	TP 1,7													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
23	Produto1	Processador8	C.1,8	0	TP 1,8													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
24	Produto1	Processador9	C.1,9	0	TP 1,9													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										

Figure 2. Microsoft Excel spreadsheet model with user-friendly interface.

Evaluation of the interface's consistency

The assessment of the consistency of the generalized equations in the Excel spreadsheet with user-friendly interface was performed by comparing the results manually obtained with Excel spreadsheets with those obtained with the user-friendly interface, using VBA language.

So that the automated spreadsheet could be validated with regard to the accuracy of the equations generated by the expansion of the model, two scenarios were tested in each one of the classes $N \in \{2,3,4,5,6,7,8,9,10,11,12,13\}$ and $M \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60\}$, with time intervals of $[1,99]$, totalling 360 problems tested (2 scenarios x 12 classes of tasks x 15 classes of resources).

The classes of problems $n \in \{2,3,4,5,6,7\}$ and $m \in \{2,3,4,5,6,7,8,9,10,15,20,25\}$ represent the small-sized production scheduling problems. However, the same author reports that classes $N \in \{8,9,10,11,12,13\}$ and $M \in \{30,40,60\}$ correspond to medium-sized production scheduling problems.

For the assessment of the consistency of generalized equations in the Excel worksheet with user-friendly interface, tests were performed in cases of production scheduling in small and medium businesses. Besides verifying the veracity of equations (1) through (9), expanded on the Excel worksheet with user-friendly interface, the tests sought to find the limits of adjustable cells on the spreadsheet. Adjustable cells mean the largest number of tasks (N) and resources (M) of the subclasses that have the possibility of solving the objective function (minimizing the Makespan) in the Excel worksheet with user-friendly interface, regardless of the final solution.

The correlation coefficient (r) given by Equation 10 (MOREIRA, 2000) was used to assess the relationship between the number of tasks (N) and resources (M) of the limiting subclasses of the Excel spreadsheet model with user-friendly interface.

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{n(\sum X^2) - (\sum X)^2} \sqrt{n(\sum Y^2) - (\sum Y)^2}} \quad (10)$$

where:

- r = correlation coefficient;
- n = number of limiting subclasses;
- X = value of the number of the task subclass (n);
- Y = value of the number of the resource subclass (m).

Results and discussion

Tests of each problem of class {2} verified that all equations were consistent with the generic model equations. However, scope to increase the number of resources was still extant, since the expansion of the equations was not at the limit allowed in the Excel spreadsheet. Therefore, a new set of problems with $N \in \{2\}$ and $M > 60$ was tested, and the limit {2,98} was found.

The same behavior was observed with the class {3}. Consequently, a new set of problems with $N \in \{3\}$ and $M > 60$ was tested and the limit {3,62} was found. All equations of problems of class {4} were consistent with the generic model. However, subclass {4,60} could not be verified, since it went beyond the limit of Solver adjustable cells present in ESUFI. Thus, the limit {4,46} was found after the performance of further tests.

The same behavior was observed for the class {5}. It was found that subclasses {5,40} and {5,60} went beyond the limit. New tests determined the limit {5,35}. Further tests with the class {6} pointed to the limit {6,27}, after it was found that subclasses {6,30}, {6,40} and {6,60} went beyond the limit of Solver adjustable cells present in ESUFI.

Similarly, in the problems of class {7}, all equations were consistent with the generic model, although subclasses {7,25}, {7,30}, {7,40} and {7,60} exceeded the limit of adjustable cells in the ESUFI. Limit {7,21} was determined after the performance of new tests. For the same reason, subclasses {8,20}, {8,25}, {8,30}, {8,40} and {8,60} could not be verified when analyzing subclasses {8,2} to {8,10}, {8,15}, {8,20}, {8,25}, {8,30}, {8,40} and {8,60}, although all equations were consistent with the model up to limit {8,17}.

During the tests with the problems of subclasses {9,2} to {9,10}, {9,15}, {9,20}, {9,25}, {9,30}, {9,40} and {9,60}, all equations were consistent with the generic model, but the last six subclasses could not be verified since they were beyond the limit of Solver adjustable cells present in the ESUFI. New tests were performed and the limit {9,13} was found. Analogously, the last six subclasses of the set {10,2} to {10,10}, {10,15}, {10,20}, {10,25}, {10,30}, {10,40} and {10,60} were not verified, and the limit {10,10} was found after new tests.

Following the same pattern, the limit observed for the subclasses {11,2} to {11,10}, {11,15}, {11,20}, {11,25}, {11,30}, {11,40}, and {11,60} was {11,7}, while for the subclasses {12,2} to {12,10}, {12,15}, {12,20}, {12,25}, {12,30}, {12,40}, and {12,60} the limit was {12,4}. As for the results of

the tests with the problems of subclasses {13,2} to {13,10}, {13,15}, {13,20}, {13,25}, {13,30}, {13,40}, and {13,60}, all equations were found to be consistent with the generic model, but only the subclass {13,2} could be generated on the spreadsheet, as the others went beyond the limit of Solver adjustable cells present in the ESUFI.

Therefore, after performing all the tests in the classes of problems $N \in \{2,3,4,5,6,7,8,9,10,11,12,13\}$ and $M \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60\}$, with time intervals of [1,99], for the validation of the automated spreadsheet regarding the accuracy of the equations generated by the model expansion and establishment of the limits of classes $N \times M$ of the PSBP model in the ESUFI, it could be concluded that equations (1) through (9) were correctly expanded in all 360 analyzed problems. The above ensured the accuracy of the results obtained with the ESUFI. Table 4 summarizes the evaluated classes and subclasses of problems; the last subclass of each problem corresponds to the ESUFI limit for the application of the Solver tool to optimize the Makespan.

Table 4. Classes of problems analyzed for ESUFI validation.

Class	Subclass
2	{2,2};{2,3};{2,4};{2,5};{2,6};{2,7};{2,8};{2,9};{2,10};{2,15};{2,20};{2,25};{2,30};{2,40};{2,60};{2,98}
3	{3,2};{3,3};{3,4};{3,5};{3,6};{3,7};{3,8};{3,9};{3,10};{3,15};{3,20};{3,25};{3,30};{3,40};{3,62};
4	{4,2};{4,3};{4,4};{4,5};{4,6};{4,7};{4,8};{4,9};{4,10};{4,15};{4,20};{4,25};{4,30};{4,40};{4,46};
5	{5,2};{5,3};{5,4};{5,5};{5,6};{5,7};{5,8};{5,9};{5,10};{5,15};{5,20};{5,25};{5,30};{5,35};
6	{6,2};{6,3};{6,4};{6,5};{6,6};{6,7};{6,8};{6,9};{6,10};{6,15};{6,20};{6,25};{6,27};
7	{7,2};{7,3};{7,4};{7,5};{7,6};{7,7};{7,8};{7,9};{7,10};{7,15};{7,20};{7,21};
8	{8,2};{8,3};{8,4};{8,5};{8,6};{8,7};{8,8};{8,9};{8,10};{8,15};{8,17};
9	{9,2};{9,3};{9,4};{9,5};{9,6};{9,7};{9,8};{9,9};{9,10};{9,13};
10	{10,2};{10,3};{10,4};{10,5};{10,6};{10,7};{10,8};{10,9};{10,10};
11	{11,2};{11,3};{11,4};{11,5};{11,6};{11,7};
12	{12,2};{12,3};{12,4};
13	{13,2};

Observing the limits found for the classes of problems in Table 4, there is a clear reduction in the number of resources (M) as the number of tasks (t) increases. This reduction in the number of resources, concerning the limitation of adjustable cells in the ESUFI for the use of the Solver tool to solve the objective function of the PSBP problem, may be assessed by the application of the correlation coefficient (r), given by Equation 10. The value of -0.89 shows a very high inverse correlation ship between the number of tasks (n) and the number of resources (m), that is, for the adjustable cells of ESUFI. As the number of tasks increases, the number of resources decreases, resulting in a reduction in the number of resources to be programmed in ESUFI.

When carrying out the validation tests of the macros employed to automate the use of the Microsoft Excel using Visual Basic Application (VBA), a limitation of the spreadsheet concerning the amount of adjustable cells was detected, that is, as the number of tasks to be programmed in the spreadsheet increases, the number of resources decreases. Since production scheduling is limited to 13 tasks, the above shows that, within standard Excel, ESUFI may be employed for small-scale problems.

It may be suggested that it is important to develop studies applied to real cases and to use a professional Excel version to increase the number of adjustable cells, which would extend the limit of variables in the production scheduling and therefore permit working with larger problems. Further development of algorithms for solving the problem in question may be recommended, coupled to the comparison of results obtained with these algorithms and those obtained in current research. It is also highly important to investigate the effect of varying the correlation coefficient (R) in the number of resources (M) and the number of tasks (t).

However, it is worth noting that current study may be applied to several real cases, since it presented a diverse mix of products and tasks that may be sequenced according to the performance goal of production characterized as decreased total time finish tasks (MAKESPAN).

Conclusion

The present study shows that the ESUFI played a very important role in the production scheduling, since spreadsheets are known by most computer users and are available in almost all Office packages.

Due to the easy manipulation of ESUFI and VBA language, a user-friendly interface could be created between the user and the spreadsheet itself.

So that the production scheduling that minimizes the total time of completion of tasks may be generated, the number of tasks and resources and the processing time of the tasks in their respective resources are the only information required in the worksheet.

References

- BAKER, K. R. **Introduction to sequencing and scheduling**. New York: John Wiley and Sons Inc., 1974.
- BURKARD, Rainer. E.; HATZL, Johannes. A complex time based construction heuristic for batch scheduling problems in the chemical industry. **European Journal of Operational Research**, v. 174, n. 2, p. 1162-1183, 2006.
- FUMERO, Y.; CORSANO, G.; MONTAGNA, J. M. A mixed integer linear programming model for simultaneous design and scheduling of flow shop plant. **Applied Mathematical Modelling**, v. 4, n. 37, p. 1652-1664, 2013.

- KONÉ, O.; ARTIGUES, C.; LOPEZ, P.; MONGEAU, M. Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. **Flexible Services Manufacturing Journal**. v. 25, n. 1-2, p. 25-47, 2013.
- LIN, R.; LIAO, C.-J. A case study of batch scheduling for an assembly shop. **International Journal of Production Economics**, v. 139, n. 2, p. 473-483, 2012.
- MACCARTHY, B. L.; LIU, J. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. **International Journal of Production Research**, v. 31, n. 1, p. 59-79, 1993.
- MENDÉZ, C. A.; CERDÁ, J.; GROSSMANN, I. E.; HARJUNKOSKY, L.; FAHL, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. **Computers and Chemical Engineering**, v. 30, n. 1, p. 913-946, 2006.
- MOCCELLIN, J. V. A new heuristic method for the permutation flow shop scheduling problem. **Journal of the Operational research Society**, v. 46, n. 7, p. 883-886, 1995.
- MOREIRA, D. A. **Production management and operations**. 5. ed. São Paulo: Pioneira, 2000.
- PAN, M.; QIAN, Y.; LI, X. A novel precedence-based and heuristic approach for short-term scheduling of multipurpose batch plants. **Chemical Engineering Science**, v. 63, n. 17, p. 4313-4332, 2008.
- REKLAITIS, G. V. Scheduling approaches for the batch process industries. **ISA Transactions**, v. 34, n. 4, p. 349-358, 1985.
- SHAKERI, S.; LOGENDRAN, R. A mathematical programming-based scheduling framework for multitasking environments. **European Journal of Operational Research**. n. 176, p. 193-209, 2007.
- SUNDARAMOORTHY, A.; KARIMI, I. A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. **Chemical Engineering Science**. v. 60, p. 2679-2702, 2005.
- TAILLARD, E. Benchmarks for basic scheduling problems. **European Journal of Operational Research**, v. 64, n. 2, p. 278-285, 1993.

Received on September 30, 2011.

Accepted on January 7, 2014.

License information: This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.