Theory and Applications of
Mathematics & Computer Science

(ISSN 2067-2764)
http://www.uav.ro/applications/se/journal/index.php/tamcs

Theory and Applications of Mathematics & Computer Science 1 (2011) 45–62

The design and scheduling of chemical batch processes: building heuristics and probabilistic analysis

João Miranda[a,*]

[a]*College of Technology and Management, Portalegre Polytechnics Institute, Lugar da Abadessa, Apt. 148; 7301-999 Portalegre, Portugal. Centre for Chemical Processes, Instituto Superior Técnico, Technical University of Lisbon.*

Abstract

The number of industrial cases published in the design and scheduling of batch multiproduct plants is short, and the difficulties to solve large models of this kind are well known, since their modeling usually consider variables integrality and data uncertainty. One way to address such difficulties is to use analytical studies to obtain significant improvements in algorithms and problem structures. Several MILP models from the open literature are selected focusing the successive generalizations on the options set, namely: from single machine to multiple parallel machines (identified by ***S*** or ***M***) in each stage; and from single product campaigns to multiple products campaigns (***S*** or ***M*** too). Four models (hereby ***SS, MS, SM, MM***) that consider zero wait operations are thus analyzed and compared, and several heuristics are developed in order to produce good approximations to the objective function's value and to the binary solution. Then, the probabilistic analysis of the heuristics was performed: the deviations on the objective function values, the deviations on the binary solutions, and the computational times are evaluated. The analysis both allowed the certification of the modeling and the numerical implementation. The model MS, addressing multiple machines and SPC, is found to be the most promising model to further developments that aim the design and scheduling of batch processes in stochastic and robust frameworks.

Keywords: Design and scheduling, Batch processes, Heuristics, Probabilistic analysis.
2000 MSC: 93C42, 93C10, 37M05.

1. Introduction

The problem of scheduling batch processes presents a large number of industry applications, such as in manufacturing systems, mechanics, electronics, and in chemical industry. In addition, the batch processes in biotechnology, food industry, fine chemicals, and pharmaceutics usually apply restrictive conditions of temperature and pressure, quality, safety, and quite expensive equipments are thus required. Considering their cost, the maximization of the equipment utilization is important to the economic efficiency of the chemical batch plants, and this subject is commonly related to the minimization of makespan.

In the design and scheduling of batch chemical processes, it is widely accept that the design problem must include the scheduling subjects. It is well known that the production policies, the storage policies, and the sequencing of the production are quite important to the economic efficiency of the batch plants (Barbosa-Póvoa, 2007). Then, the design problem simultaneously considers the solution of detailed scheduling sub-problems, anticipating the operation modes early at the design stage.

It must be well defined if the model is focusing to support decisions of economic type, concerning the treatment of external elements (purchases, sales, investments), or if it is aiming the internal decisions related to the batch operations (setup and hold-on times, maintenance, control), in close relation with the given time horizon (Romero

*Corresponding author
Email address: `jlmiranda@estgp.pt` (João Miranda)

et. al., 2003; Moreno et. al., 2007). Usually, external decisions consider uncertainty and risk treatments through stochastic framework, while internal decisions are addressed by logical relations and applying integer and binary variables.

The simultaneous approach, considering both the external and stochastic subjects and the internal and integrality requirements, is very difficult to solve (Pekny, 2002). By other side, the implementation of MILP or MINLP deterministic models is common in the design of flowshop batch plants (Barbosa-Póvoa, 2007), with the purposes of selecting and sizing equipments. One way to address the difficulties is to use analytical studies and computational complexity to gain insight (Liu and Sahinidis, 1997; Ahmed(Ahmed & Sahinidis, 2000) and (Ahmed & Sahinidis, 2003), and to obtain significant improvements in algorithms and problem structures. Also, the application of heuristics, tabu search (Cavin *et al.*, 2004), ant colony procedures (Jayaraman *et al.*, 2000), or evolutionary algorithms (Tan & Mah, 1998), is spreading in the design and scheduling of batch chemical processes.

Consequently, several deterministic MILP models from the open literature are selected (Voudouris & Grossmann, 1992), they are analyzed and compared, and heuristics that can be useful in stochastic framework are then developed. The selected models of the design and scheduling of batch chemical processes are focusing successive generalizations in the options set, but all consider zero wait policy (ZW) and flowshop environment. The options set is considering variations in the number of processes implanted in each production stage (single machine vs. multiple machine, ***S*** or ***M***), and in the mode of production campaign, single versus multiple product campaign (SPC *vs*. MPC, ***S*** or ***M***). In a combinatory way, the four alternative models **(SS, MS, SM, MM)** are analyzed, (Miranda, 2007) showed that are occurring equivalence relations between specific instances of them, and these models thus belong to the same class of computational complexity.

Several heuristic procedures are developed, in order to produce good quality approximations to the objective function value, or to build the optimal or a near optimal binary solution. The probabilistic analysis of the heuristics is performed, evaluating the deviations to the objective value and to the binary solution, and also comparing the computational times. Numerical experiences are described, related not only to the heuristics analysis, but also to certify the modeling and the numerical implementation of the several models. This is an important point, due to some modeling inconsistencies that drives solutions incoherence, as described later.

The structure of the paper considers: in Section 2, the mathematical models at hand are presented; in Section 3, the heuristic procedures focusing model ***SS*** (single machine and SPC) are developed, and the results are analyzed; in Section 4, the heuristic procedures onto model MS (multiple machine and SPC) are developed and analyzed; in Section 5, the models ***SM***(single machine and MPC) and ***MM*** (multiple machine and MPC) are addressed; in Section 6, the main conclusions are presented.

2. The models for design and scheduling of batch chemical processes

The issue of scheduling batch processes is embedded in the process design, in a way to conjugate the short term decisions from the operational planning within the long term investment planning. The models are collected from the open literature (Voudouris & Grossmann, 1992), and they are selected based in their character of successive generalizations in the options set: *i)* from single machine to multiple machines working in parallel at each stage; *ii)* from SPC to MPC. The search space is thus enlarged, from one single option to many options in each of the subjects, and the solutions are expected to be qualitatively and quantitatively improved.

The four models in analysis are considering the flowshop and ZW contexts, in the following sense:

- The flowshop problem (Garey *et al.*, 1976) considers that N jobs are to be processed on M stages sequentially and it assumes: one machine is available at each stage; one job is processed on one machine at a time without preemption; and one machine processes no more than one job at a time.
- The zero wait (ZW) storage mode considers that the materials produced in one stage are directly entering in the next stage without any storage (unlimited intermediate storage, UIS), or even without waiting inside the equipment until the next stage is available to process them (no intermediate storage, NIS).

The models are described successively in this section, using a generalization approach that shows the successive enlargement of the options set. The MILP model ***SS*** considers one single machine in the several production stages, SPC and ZW operations, and the discrete enumeration of equipments.

Model SS

1-a $$[\min] \quad z = \sum_{j=1}^{M} \sum_{s=1}^{NS(j)} c_{js} y_{js}$$

subject to

1-b $$S_{ij} Q_i CT_i \sum_{s=1}^{NS(j)} \frac{y_{js}}{dv_{js}} - t_i \leq 0, \ \forall i, j$$

1-c $$\sum_{s=1}^{NS(j)} y_{js} = 1, \forall j$$

1-d $$\sum_{i=1}^{N} t_i \leq H$$

1-e, f $$t_i \geq 0, \ \forall i; y_{js} \in \{0; \ 1\}, \ \forall j, s$$

The MILP model ***MS*** is enlarging model ***SS*** by addressing multiple machines *NP(j)* in parallel in stages *j*, in a way to reduce the cycle time CT_i of each product *i*. In SPC and ZW modes, the cycle time corresponds to the maximum of the processing times of each product considering all stages *j* (Biegler *et al.*, 1997):

2 $$CT_i = \max_{\substack{j=1..M \\ i=1..N}} \left\{ \tau_{ij} / NP(j) \right\}$$

Thus, when a second machine is introduced on a specific stage, the processing times in this stage are reduced to half, and so on. The discrete enumeration of equipments allows that the normalized sizes of processing equipments are included.

Model *MS*

3-a $$[\min] \quad z = \sum_{j=1}^{M} \sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} c_{jsp} y_{jsp}$$

subject to

3-b $$S_{ij} Q_i \sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} \frac{y_{jsp}}{dv_{js}} - n_i \leq 0, \ \forall i, j$$

3-c $$\sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} y_{jsp} = 1, \ \forall j$$

3-d $$\sum_{i=1}^{N} t_i \leq H$$

3-e $$\sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} \left(\frac{\tau_{ij}}{p} nc_{ijsp} \right) - t_i \ \leq 0, \forall i, j$$

3-f $$nc_{ijsp} - nc_{ijsp}^{Upp} y_{jsp} \leq 0, \ \forall i, j, s, p$$

3-g $$\sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} nc_{ijsp} - n_i = 0, \ \forall i, j$$

3-h,i $$nc_{ijsp}, \ n_i, \ t_i \geq 0, \ \forall i, j, s, p; y_{jsp} \in \{0; \ 1\}, \ \forall j, s, p$$

When transition times (for clean-up, setup, or tuning) between different products are short, it is useful to consider a mix of products running in the same production campaign. The adoption of MPC considers the sequencing of different products with the purpose to reduce idle times, when the equipments are not in use. This way, the global cycle time can be minimized and the equipment utilization can be improved. With the same characteristics of flowshop and ZW, (Voudouris & Grossmann, 1992) presented the model considering one single machine at each stage and MPC production policy, hereby referred by ***SM***.

Model *SM*

4-a $$[\min] \quad z \ = \sum_{j=1}^{M} \sum_{s=1}^{NS(j)} c_{js} y_{js}$$

subject to

4-b $$S_{ij} Q_i \sum_{s=1}^{NS(j)} \frac{y_{js}}{dv_{js}} - n_i \leq 0, \ \forall i, j$$

4-c $$\sum_{s=1}^{NS(j)} y_{js} = 1, \forall j$$

4-d $$\sum_{k=1}^{N} Nch_{ik} = n_i, \ \forall i$$

4-e $$\sum_{i=1}^{N} Nch_{ik} = n_k, \ \forall k$$

4-f $$Nch_{ii} \leq n_i - 1, \ \forall i$$

4-g $$\sum_{i=1}^{N} \left[n_i \tau_{ij} + \sum_{k=1}^{N} SL_{ikj} Nch_{ik} \right] \leq H, \ \forall j$$

4-h,i $$n_i, \ Nch_{ik} \geq 0, \ \forall i, \ k; y_{js} \in \{0; 1\}, \ \forall j, s$$

The model ***SM*** is then directly enlarged by considering the implementation of multiple machines in parallel at each stage and neglecting the transition times. Thus, one more dimension or group of variables is incorporated in the problem: the number of processes p(j) considered in each stage j. In substitution of the restrictions **4-g**, related to the satisfaction of the horizon time in each stage j, the following restrictions are introduced:

5 $$\sum_{i=1}^{N} n_i \tau_{ij} \leq H. \sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} p(j) y_{jsp}, \ \forall j$$

This substitution is developed in flowshop and ZW contexts, and the model ***MM*** (Voudouris & Grossmann, 1992) focused multiple machines and MPC.

Model *MM*

6-a
$$[\min] \quad z = \sum_{j=1}^{M} \sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} c_{jsn} y_{jsp}$$

subject to

6-b
$$S_{ij} Q_i \sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} \frac{y_{jsp}}{dv_{js}} - n_i \leq 0, \ \forall i, j$$

6-c
$$\sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} y_{jsp} = 1, \forall j$$

6-d
$$\sum_{i=1}^{N} n_i \tau_{ij} - H. \sum_{s=1}^{NS(j)} \sum_{p=1}^{NP(j)} p(j) y_{jsp} \leq 0, \ \forall j$$

6-e
$$\sum_{k=1}^{N} Nch_{ik} = n_i, \ \forall i$$

6-f
$$\sum_{i=1}^{N} Nch_{ik} = n_k, \ \forall k$$

6-g
$$Nch_{ii} \leq n_i - 1, \ \forall i$$

6-h
$$\sum_{i=1}^{N} \left[n_i \tau_{ij} + \sum_{k=1}^{N} SL_{ikj} Nch_{ik} \right] \leq H, \ \forall j$$

6-i,j
$$n_i, \ Nch_{ik} \geq 0, \ \forall \ i, \ k; y_{jsp} \in \{0; \ 1\}, \ \forall j, s, p$$

3. Model SS analysis

The model **SS** (single machine and SPC) searches how to choose the smallest discrete size (the cheapest) equipment to satisfy products demand Q_i, on a given time horizon H, knowing that the cycle time CT_i of each product campaign (SPC) is size-dependent.

In Table 1, six numerical examples (A series) are specified in terms of the parameters of the model ***SS***, numbers of binary variables, continuous variables, and restrictions, and the mean of the solving times for the100 instances of each example. It was used the OSL solver provided within GAMS, running on an ASUS-F3JC (Intel Core2 T5500, 1.55GHz). In the example EX3A, the solver time (in *italic*) is related to a large number of unfeasible instances: when infeasibility occurs, the corresponding execution is stopping before the instance's optimum is reached.

In model ***SS***, some simplifications were important in the formulation of the horizon time *H*, the production times t_i, and in the cycle times CT_i. However, when comparing it with other models it must be noted that:

- The time slacks SL_{ikj} that represent transitions between different products were neglected;
- In each production campaign, the head and tail are neglected;
- The overlapping between campaigns of different products is avoided, assuming that one product *k* begins its SPC production only after the former product *i* was finished.

The approximation procedures for the heuristic aiming the optimal solution of model ***SS*** are described in the following steps:

Table 1. Numbers of variables and restrictions in the numerical examples of model ***SS*** (A series).

Examples	**Parameters** (N, M, NS)	**Binary variables**	**Continuous variables**	**Restrictions**	**Time *solver*** (s)
EX1A	2, 3, 4	12	2	9	0.23
EX2A	3, 4, 5	20	3	17	0.36
EX3A	5, 4, 6	24	5	25	0.28
EX4A	6, 4, 10	40	6	29	0.76
EX5A	8, 5, 12	60	8	46	1.03
EX6A	10, 5, 15	75	10	56	2.17

1. Feasibility's previous step;
2. Initial guess;
3. Building feasible solutions:
   a) Using proportionality to demands; and
   b) Reducing production times, through *iterated local search (ILS)*

Step 1) Feasibility's previous step

In order to ensure solutions feasibility, a previous procedure is developed, *PREVIOUS_SS*. It assumes that the time constraints must be satisfied when the largest size *NS(j)* is selected in all the stages *j*, by assigning:

7
$$s'(j) = NS(j), \ \forall j$$

The assumption is checked: if positive, the execution continues; else the single machine instance is unfeasible.

Step 2) Initial guess

The procedure *APROX0_SS* assumes that the time horizon *H* must be satisfied by the sum of production times ti in all the stages *j*. Then, the minimum volume *V_h0* that is required in each stage *j* is estimated, and the enumeration of sizes is avoided:

8
$$H = \sum_{i=1}^{N} \frac{S_{ij} Q_i CT_i}{V\_h0(j)}, \ \forall j \ \Rightarrow \ V\_h0(j) = \sum_{i=1}^{N} \frac{S_{ij} Q_i CT_i}{H}, \ \forall j$$

The estimated values of volumes are adjusted to the discrete size immediately above, and the corresponding binary variables are assigned with the value 1:

9
$$y\_h0(j, s'(j)) = 1$$

The other binary variables are assigned to 0, and the solution is checked using the procedure *VERIFY_SOLUTION_SS* to evaluate the costs, variables, and restrictions satisfaction. If the solution is unfeasible, then go to Step 3b.

Step 3) Building feasible solutions

The procedure *APROX1A_SS* builds feasible solutions by allocation of the production times t_i in proportionality with the demand quantities, Q_i, for each product *i*. In plus, facing an unfeasible solution (e.g., obtained from *APROX0* in *Step 2*), the procedure *APROX1B_SS* performs a local search (ILS) to reduce the production times t_i in close relation with the equipment costs c_{js}.

Step 3a) Procedure *APROX1A_SS*

For each one of the products *i*, the production time t_i is made proportional to its demand Q_i, and the time horizon *H* is partitioned among the various products in the same rate as the demand rate $\left(Q_i/Q_{total}\right)$ occurs:

10
$$\begin{cases} Q_{total} = \sum_{i=1}^{N} Q_i \\ t_{total} = \sum_{i=1}^{N} t_i \leq H \end{cases}$$

If no preference of products is specified in terms of the prices, profits, availabilities of raw materials, or other attributes, then,

11
$$\frac{t_i}{t_{total}} = \frac{Q_i}{Q_{total}} \Rightarrow \frac{t_i}{Q_i} = \frac{t_{total}}{Q_{total}} \leq \frac{H}{Q_{total}}.$$

The proportional times approach is driving feasible solutions, given that:

12
$$t_i \leq Q_i \frac{H}{Q_{total}} \Rightarrow \sum_{i=1}^{N} t_i \leq \sum_{i=1}^{N} \left(Q_i \frac{H}{Q_{total}}\right) \leq H.$$

The procedure checks all the products i, using the proportionality condition,

13
$$\frac{t_i}{Q_i} = \max_j \left\{\frac{S_{ij}\, CT_i}{dv_{j,s'(j)}}\right\} \leq \frac{H}{Q_{total}} , \forall i$$

and, in each stage *j*, the shorter discrete volume $dv_{j,s(j)}$ that verifies the inequality is then selected.

However, unfeasibility can occur during the selection of the discrete sizes. If a very large ratio arises from the problem parameters, the time horizon H may be overpassed even selecting the largest size *NS(j)* of the equipment, namely, in case of:

14
$$\max_j \left\{\frac{S_{ij}\, CT_i}{dv_{j,NS(j)}}\right\} > \frac{H}{Q_{total}} , \exists (i, j).$$

Step 3b) Procedure *APROX1B_SS* (Local Search)

Initiating with an unfeasible solution, the procedure *APROX1B_SS* searches to reduce the production times ti and then to satisfy the time horizon *H*. The reduction is performed in conformity with the equipment costs c_{js}, since the selection of the equipment that increases in size is based in best ratio between the corresponding variations in production times and variations in costs.

The evaluation of the costs increase is direct, by direct subtraction between the corresponding costs.

To evaluate the impact of selecting larger equipment in production times, it must be noted that the cycle time CT_i for each product *i* corresponds to the maximum of the processing times considering the stages *j* (the bottleneck stage). The stage where this maximum value occurs is the stage corresponding to the minimum value for the batches dimension B_i. The bottleneck stage is referred hereby by the *critical stage* $jcrit1_i$, and when larger equipment is selected on it, the bottleneck will then occur in other stage, $jcrit2_i$. The latter can be picked up in the descending order of the production times, from $tcrit1_i$ to $Bcrit2_i$, or in the ascending order of batches dimension, respectively, from $Bcrit1_i$ to $Bcrit2_i$. When larger equipment is selected in a *critical stage*, the total variation in the production times is obtained by the sum of all the variations in the production times, and these variations are estimated considering the variations in batches dimensions B_i.

It is known that

15
$$t_i = \max_j \left\{\frac{S_{ij} Q_i\, CT_i}{dv_{j,s'(j)}}\right\} = Q_i\, CT_i \max_j \left\{\frac{S_{ij}}{dv_{j,s'(j)}}\right\} = n_i\, CT_i \ , \forall i$$

and considering

16
$$n_i = Q_i \max_j \left\{ \frac{S_{ij}}{dv_{j,s'(j)}} \right\} = \frac{Q_i}{B_i},\ \forall i \Rightarrow B_i = \min_j \left\{ \frac{dv_{j,s'(j)}}{S_{ij}} \right\},\ \forall i$$

then the batches dimension, B_i, are obtained from the production times, t_i, and the number of batches, n_i, being these values defined accordingly with the present critical stage. The variation in production times is then expressed by:

17
$$\begin{aligned} \Delta t_i &= tcrit1_i - tcrit2_i \\ &= \frac{Q_i\, CT_i}{Bcrit1_i} - \frac{Q_i\, CT_i}{Bcrit2_i} = \frac{Q_i\, CT_i}{Bcrit1_i} - \frac{Q_i\, CT_i}{Bcrit1_i}\left(\frac{Bcrit1_i}{Bcrit2_i}\right) \\ &= \frac{Q_i\, CT_i}{Bcrit1_i}\left(1 - \frac{Bcrit1_i}{Bcrit2_i}\right) = tcrit1_i\left(1 - \frac{Bcrit1_i}{Bcrit2_i}\right),\ \forall i \end{aligned}$$

When an unfeasible solution is treated in the procedure *APROX1B_SS*, then for each product *i*:

i) One larger size is selected in a given critical stage for product i; the variation in costs is calculated;
ii) The sum of the variations in production times is evaluated;
iii) The variations in times and in equipment costs are compared, using a ratio;
iv) The best ratio is selected, and the corresponding discrete size is increasing, by swapping the related binary variables (assign 0 to the existing selection, and assign 1 to the next larger size);
v) To ensure feasibility, this binary solution is verified using the procedure *VERIFY_SOLUTION_SS*;
vi) If the solution is unfeasible and the iterations limit is not reached, then return to step i).

The procedure *APROX1B_SS* is executed iteratively until a feasible solution is achieved and it performs an *iterated local search* (ILS) considering:

- The searching neighborhood is defined by the increasing sizes of each of the critical stages, $jcrit1_i$;
- The evaluation function corresponds to the ratio between the variations in production times and the variations in costs;
- The alteration in the solution is defined by the neighbor solution with best (maximum) value for the described ratio;
- The termination criteria are stating the end of the procedure when a feasible solution is found or the number of iterations is reached.

The heuristic for model ***SS*** integrates the procedures from the feasibility test to the construction of feasible solutions, passing by the initial guess, the proportionality on times, and the solution verification. The values of several metrics (formulated in Appendix A.3) related to the deviations in the objective function values and in the binary solutions are presented in Table 2, namely:

- Objective function - percentage of feasible instances that are exactly estimated // percentage of unfeasible instances // percentage mean of the errors on the sub-optimal estimations // standard deviation on the percentage mean of the errors;
- Binary solutions - percentage of instances presenting exact binary solution // mean on the number of binary errors for the sub-optimal estimations.

The heuristic for model ***SS*** is satisfactory since:

- if the instance is feasible, then one solution is always obtained;
- optimal solutions are obtained in nearly 63% of the feasible instances;
- sub-optimal solutions present good quality, the errors are about 1%-2% in value, and on average nearly only one discrete size is switched.

Table 2. Computational times and deviations for heuristic of model ***SS***.

Examples	Time *solver (s)*	Time heuristic (s)	Objective Function Deviations	Binary Solution Deviations
EX1A	0.23	0.0006	91.// 0.// 6.8// 4.4	91.// 2.22
EX2A	0.36	0.0028	72.// 8.// 2.4// 1.7	72.// 1.96
EX3A	0.28	0.0005	73. // 56.// 1.2// 0.9	73.// 1.58
EX4A	0.76	0.0016	50. // 4. // 1.5// 1.4	50.// 1.79
EX5A	1.03	0.0020	30.// 0.// 1.5// 1.5	30.// 2.43
EX6A	2.17	0.0056	53.// 0.// 1.1// 0.9	53.// 2.00

Notwithstanding the model ***SS*** showed some weaknesses:

- Unfeasibility occurred on a large number of instances, namely, when the number of products N is increased, or for large values of the demands Q;
- The transition, setup, and cleanup times were not considered;
- The number of batches is neither specified, nor its integrality required.

As described, the model ***SS*** presents some weaknesses that limit its application onto industry's real cases and the design approach is enlarged onto multiple machines in each stage.

4. Model MS analysis

The model ***MS*** (multiple machine and SPC) searches how to choose the cheaper combination of discrete sizes to satisfy products demand Q, on the horizon H, but considering processes in parallel at each stage.

This model allows the implementation of equipments working in parallel in each production stage, supposing that the cycle times can be reduced in concordance with the reduction of the processing times in those stages with multiple machines.

The model ***SS***, with one single machine in each stage, may be addressed as a specific instance in the model MS if the number of processing machines is constrained to $NP = 1$ (Miranda, 2007), and equivalence relations thus occur between variables, objective function, and restrictions of the two models. This theoretic equivalence is also used to validate the computational runs: the machines number is supposed 1 in all the stages of MS and two series of 100 random samples (Appendix A) are treated. Then, the results are compared, and the equivalence between corresponding instances was numerically verified.

In Table 4, the parameters of the numerical examples for model ***MS*** are presented, so as the average of the solution times.

Table 3. Numbers of variables and restrictions in the numerical examples of model ***MS*** (A series).

Examples	Parameters (N, M, NS, NP)	Binary variables	Continuous variables	Restrictions	Time *solver*(s)
EX1A	2, 3, 4, 2	24	52	70	0.30
EX2A	3, 4, 5, 3	60	186	221	10.53
EX3A	5, 4, 6, 3	72	370	425	101.67
EX4A	6, 4, 10, 3	120	732	797	405.99
EX5A	8, 5, 12, 3	180	1456	1566	(4098.31)
EX6A	10, 5, 15, 3	225	2270	2406	(15142.36)

Given the fast increase of solving times, a time limit of 1 hour (3600 s) is stated. The examples EX5A and EX6A largely overpassed this time limit, and their mean times are taken from 30 instances and presented in-between round brackets. In comparison with the solution times for model ***SS***, the cause of the increasing in MS times relies not only in the larger number of binary variables, but also in the much larger number of continuous variables.

These computational difficulties are strengthening the necessity of a heuristic aiming the model ***MS***. The heuristic is considering the following procedures:

1. Feasibility's previous step;
2. Initial guess;
3. Building feasible solutions;
    a) If feasible, solving by ***SS*** heuristic;
    b) Reducing production times, through ILS;
4. Tuning of feasible solutions;
    a) Reducing discrete sizes, through ILS.

Step 1) Feasibility's previous step

A previous procedure *PREVIOUS_MS* is developed to ensure the instance's feasibility. It assumes the satisfaction of the time constraints by selecting the largest number of machines *NP(j)* within the largest size *NS(j)*, in all the stages *j*:

18 $$\begin{cases} p'(j) = NP(j) \\ s'(j) = NS(j) \end{cases}, \forall j$$

If assumption is positively checked then execution continues. Else the instance with multiple machines is unfeasible. In real cases, the cardinality of machines and/or discrete sizes must be enlarged if feasible solutions are desired.

Step 2) Initial guess

The procedure *APROX0_MS* is similar to the initial guess for model ***SS***, since it equally assumes that the time horizon *H* must be satisfied by the sum of production times t_i in all the stages *j*. Then, the minima for the machines number *p_h0* and volume sizes *V_h0*, in each stage *j*, is estimated using the expressions:

19 $$H = \sum_{i=1}^{N} \frac{S_{ij} Q_i \tau_{ij}}{p\_h0 * V\_h0} \Rightarrow p\_h0 * V\_h0 = \sum_{i=1}^{N} \frac{S_{ij} Q_i \tau_{ij}}{H}, \forall j$$

The continuous value of machines and volumes are adjusted to the immediately above discrete numbers of machines $p'(j)$ and sizes $s'(j)$, and the corresponding binary variables are assigned with the value 1 while the other binary variables are set to 0:

20 $$y\_h0(j, s'(j), p'(j)) = 1$$

This initial guess is then checked using the procedure *VERIFY_SOLUTION_MS*, to evaluate costs, variables, and restrictions satisfaction.

Step 3) Building feasible solutions

Two approaches are used to build feasible solutions for model MS. The procedure *APROX1A_MS* is based in the single machine approach, in case of feasibility and since its heuristic is already tuned. In case of the solution at hand is unfeasible, the procedure *APROX1A_MS* performs a local search to increase the number and/or the sizes of equipments, by balancing the reduction in the production times with the increase in costs.

Step 3a) Procedure *APROX1A_MS*

The procedure *APROX1A_MS* is based in the feasibility of the single machine approach. A feasible solution built this way is an upper bound for the multiple machine approach, and further developments can be applied to improve it, *e.g.*, by reducing discrete sizes as described later in *Step 4*. The procedure *APROX1A_MS* is considering some procedures already tuned in the single machine approach: firstly, verification of single machine feasibility; if positive, the related procedures for the initial guess *(APROX0_SS)* and feasible solution search *(APROX1_SS)* are applied.

Step 3b) Procedure *APROX1B_MS* (Local Search)

The procedure *APROX1B_MS* constructs a feasible solution from an unfeasible one originated in prior procedures *APROX0_MS* or *APROX1A_MS*. It searches to reduce the production times t_i in order to satisfy the time horizon H, and evaluating the related increase in costs c_{jsp}. The approach is similar to the one developed for single machine, since it provided satisfactory results. Increasing the discrete size or the number of processing machines, the variation in costs is calculated by a direct subtraction of values.

The variation in production times considers the cycle time of each product i, which depends on the critical stage that may be defined accordingly the size or the number of machines. This is driving the *critical number* of batches or the *critical cycle time*, respectively, j_nb1_i or j_ct1_i. When the critical stage suffers an alteration in size or number, a *second critical stage* is specified for each product, j_nb2_i or j_ct2_i, in descending order of the production times. The total variation in production times is thus evaluated; the comparison with the corresponding variation in costs is performed by a ratio; the best ratio is selected, that is, the maximum value of the variations ratio. In fact, note the relations,

21 $$t_i = \max_j \left\{ \frac{\tau_{ij}}{p'(j)} n_i \right\} = CT_i.n_i, \text{ where } \begin{cases} n_i = \max_j \left\{ \frac{S_{ij}Q_i}{dv_{j,s'(j)}} \right\} \\ CT_i = \max_j \left\{ \frac{\tau_{ij}}{p'(j)} \right\} \end{cases}, \forall i$$

When the size is increased in a critical stage j_nb1_i, the values of the related auxiliary variables $nbcrit1_i$ are modified onto $nbcrit2_i$, since a second critical stage j_nb2_i is then defined. The variation in production times is expressed in terms of the numbers of batches considering that for each product i,

22 $$\begin{aligned} t_i &= tcrit1_i - tcrit2_i \\ &= CT_i.nbcrit1_i - CT_i.nbcrit2_i = CT_i.nbcrit1_i - CT_i.nbcrit1_i \left(\frac{nbcrit2_i}{nbcrit1_i}\right) \\ &= CT_i.nbcrit1_i \left(1 - \frac{nbcrit2_i}{nbcrit1_i}\right) = tcrit1_i \left(1 - \frac{nbcrit2_i}{nbcrit1_i}\right), \ \forall i \end{aligned}$$

Similarly, if the machines number is increased in a critical stage j_ct1_i, then the values of the related variables $CTcrit1_i$ are modified onto $CTcrit2_i$, since a second critical stage j_ct2_i is then defined. For each product i, the variation in production times is expressed in terms of the cycle times,

23 $$\begin{aligned} \Delta t_i &= tcrit1_i - tcrit2_i \\ &= n_i.CTcrit1_i - n_i.CTcrit2_i = n_i.CTcrit1_i - n_i.CTcrit1_i \left(\frac{CTcrit2_i}{CTcrit1_i}\right) \\ &= n_i.CTcrit1_i \left(1 - \frac{CTcrit2_i}{CTcrit1_i}\right) = tcrit1_i \left(1 - \frac{CTcrit2_i}{CTcrit1_i}\right), \ \forall i \end{aligned}$$

The procedure *APROX1B_MS* performs an approach that is similar to the prior ILS procedure in the single machine approach, namely, in the neighborhood search, evaluation function, alteration selection, and termination step. *APROX1B_MS* initiates with an unfeasible solution and for each product i:

i) One larger size (or number of machines) is selected in a critical stage for product i; the variation in costs is calculated;
ii) The sum of the variations in production times is evaluated;
iii) The variations in times and in equipment costs are compared by its ratio, and considering the numbers of batches (or the cycle times);
iv) The best ratio is selected, and the corresponding discrete size is increasing, by swapping the related binary variables (assign 0 to the existing selection, and assign 1 to the next larger size);
v) To ensure its feasibility, the present binary solution is verified using the procedure *VERIFY_SOLUTION_MS*.
vi) If the solution is unfeasible and the iterations limit is not reached, then return to step i).

Step 4) Tuning of feasible solutions

The procedure *APROX2_MS* searches to improve a feasible solution that was generated in the prior procedures *APROX1A_MS* or *APROX1B_MS*. It aims to reduce equipment costs c_{jsp} for the *non-critical stages*: stages where the processing time for a given product is shorter than its cycle time. Note that the maximum of the processing times for a given product occurs in its critical stage, where a bottleneck exists for the product. Then, bottlenecks never occur in the *non-critical stages*. The reduction in costs is obtained from a reduction in the discrete size or in the number of machines of a *non-critical stage*. The production times may (or may not) increase, but it is ensured that the satisfaction of the time horizon *H* remains. The type of reduction (in size or in number of machines) is selected in concordance with the best ratio between the corresponding variations in costs and in production times. This approach is evolving in opposite sense of the prior procedures in *Step3*, but the reasoning is based in the same kind of relations: *i)* reducing the sizes $dv(j, s'(j))$, then the related numbers of batches ni are increasing, or vice versa; or *ii)* reducing the numbers of processing machines *p(j)* then the related cycle times CT_i are increasing, or vice versa.

Step 4a) Procedure *APROX2_MS* (Local Search)

The procedure *APROX2_MS* searches to diminish the equipment costs, but it proceeds in opposite sense of the procedures that aim to construct a feasible solution. When the equipment size is reduced, the variation in times is related with the increasing (or not) numbers of batches,

24
$$\begin{aligned} t_i &= tcrit2_i - tcrit1_i \\ &= CT_i.nbcrit2_i - CT_i.nbcrit1_i = CT_i.nbcrit1_i\left(\frac{nbcrit2_i}{nbcrit1_i}\right) - CT_i.nbcrit1_i \\ &= CT_i.nbcrit1_i\left(\frac{nbcrit2_i}{nbcrit1_i} - 1\right) = tcrit1_i\left(\frac{nbcrit2_i}{nbcrit1_i} - 1\right), \ \forall i \end{aligned}$$

If the number of processing machine is reduced, similarly, the variation in production times is related with the increasing (or not) of cycle times:

25
$$\begin{aligned} \Delta t_i &= tcrit2_i - tcrit1_i \\ &= n_i.CTcrit2_i - n_i.CTcrit1_i = n_i.CTcrit1_i\left(\frac{CTcrit2_i}{CTcrit1_i}\right) - n_i.CTcrit1_i \\ &= n_i.CTcrit1_i\left(\frac{CTcrit2_i}{CTcrit1_i} - 1\right) = tcrit1_i\left(\frac{CTcrit2_i}{CTcrit1_i} - 1\right), \ \forall i \end{aligned}$$

The procedure *APROX2_MS* initiates with a feasible solution. For a given non-critical stage, the alteration by reducing the discrete size (or the number of machines) is analyzed, and the variation in costs is compared with the total variation of the production times; the ratio between the referred variations is evaluated and the best ratio is selected; and the binary variables corresponding to the alteration are adjusted. Finally, the feasibility of the solution is verified in the procedure *VERIFY_SOLUTION_MS*. This procedure *APROX2_MS* also develops a local search, where the termination criteria are both the exhausting of feasible solutions and the iterations limit. The described procedures are integrated in the heuristic for model ***MS***, from *Step1* to *Step4*. The values of the metrics related to the errors in the objective function and in the binary solutions are presented in Table 4:

Table 4. Execution times and deviations for heuristic of model ***MS***.

Examples	Time ***solver***(s)	Time heuristic (s)	Objective Function Deviations	Binary Solution Deviations
EX1A	0.30	0.0027	84.// 0.// 7.6// 4.9	84.// 2.19
EX2A	10.53	0.0038	68.// 0. // 3.4// 3.8	68.// 2.03
EX3A	101.67	0.0042	42.// 0.// 7.0// 6.4	42.// 2.48
EX4A	405.99	0.0072	48.// 0.// 1.7// 2.1	48.// 1.88
EX5A	(4098.31)	0.0166	37.// 0.// 1.5// 1.5	37.// 2.58
EX6A	(15142.36)	0.0250	53.// 0.// 0.9// 0.9	53.// 2.00

The heuristic developed for model ***MS*** is satisfactory since:

i) Feasible solutions are always obtained;
ii) Optimal solutions are obtained in nearly 56% of the feasible instances;
iii) Sub-optimal solutions are of good quality, namely, errors are less than 2% in value for large instances, and almost only one discrete size switched.

Anyway, the model MS presented strong and weak points:

- Feasible solutions are always obtained;
- For comparable instances, the ***MS*** solutions are better or equal to the solutions in ***SS***;
- For large instances, with numerous continuous variables and constraints, the bound (3600 s) in execution time is often exceeded;
- Transition, setup, and cleanup times were not considered;
- The numbers of batches are specified, but their integrality is not required or imposed.

The model ***MS*** presented some solving difficulties, related with the treatment of the large number of continuous variables and restrictions. Nevertheless the satisfactory results from the heuristic, this model presents some weak points in the modeling of real cases, and the design is enlarged to MPC policy in next section.

5. Analysis of models *SM* and *MM*

The model ***SM*** searches how to choose the smallest (cheapest) size to satisfy products demand Q, on a given time horizon H, but considering the scheduling of several products in each production campaign (MPC).

In addition to the data for the SPC models, it is necessary to specify the slack times SL_{ikj} that occur in the transition from product i to product k, in the processing machines of stage j. The slack times are calculated by a recursive algorithm (Biegler *et al.*, 1997) that requires the cleanup times CL_{ikj} of the corresponding transitions.

However, the solving difficulties are harder than in SPC, because the integration of the transition times within the ZW policy is requiring the entirely definition of the cyclic scheduling. In the opposite sense, notice that for UIS and zero values for cleanup times ($CL_{ikj} = 0$), the cycle times of each product are analytically calculated. However, when addressing UIS and non-zero cleanup times, the cyclic scheduling is required too.

The model ***SS***, addressing SPC production policy, may be treated as a specific instance of the model ***SM*** (Miranda, 2007), and equivalence relations occur between variables, objective function, and restrictions of these two models. For that, the number of transitions for each product i is constrained to:

26 $$Nch_{ii} = n_i - 1, \ \forall i$$

The theoretic equivalence is again used to validate the computational efforts: in model ***SM***, the number of transitions is satisfying *equation 26* for all the products; and cleanup and slack times are considered in the B series of random instances. The results are compared, and the equivalence between corresponding values was verified. Thus, the optimal value of model ***SS*** represents an upper bound of model ***SM***, since the latter considers MPC policy. In fact, the feasible space of MPC solutions is enlarging the set of solutions when SPC is addressed.

In Table 5, the parameters of the B series of examples for model ***SM*** are presented, so as the mean values of execution times. The instances in EX5B and EX6B are unfeasible (* mark), due to the large number of products and thus their demands are not achievable in the horizon time.

The execution times are moderate when compared with those of model ***MS*** (multiple machines and SPC). The solution in ***SM*** is quite efficient due to the aggregated TSP's formulation (Birewar & Grossmann, 1989*a*) and (Birewar & Grossmann, 1989*b*) of the transition times: this formulation allows the specification of the transition times by LP. Since the execution times are moderate, the development of a dedicated heuristic for model ***SM*** is weighted, and it is found not necessary.

Comparing the models ***SS*** and ***SM*** that are supposing SPC and MPC policies, respectively, the metrics (defined in Appendix) in Table 6 allow the analysis of the objective function and the binary solutions in comparable instances:

Table 5. Numbers of variables and restrictions in ***SM*** examples (B series).

Examples	Parameters (N, M, NS, NP)	Binary variables	Continuous variables	Restrictions	Time (s)
EX1B	3, 4, 5	20	12	29	0.46
EX2B	6, 4, 10	40	42	50	1.42
EX3B	8, 5, 12	60	72	74	2.38
EX4B	12, 6, 15	90	156	120	2.89
EX5B	20, 7, 17	119	420	214	*
EX6B	30, 9, 20	180	930	378	*

- Objective function - percentage of improved instances (negative deviations in costs, for the feasible instances) // percentage of unfeasible instances // percentage mean of the cost improvements // standard deviation on the percentage mean of cost improvements;
- Binary solutions - percentage of instances presenting the same binary solution // mean of the binary deviations for the comparable instances.

Table 6. Comparison between models ***SM*** and ***SS*** (MPC *vs*. SPC).

Examples	Parameters (N, M, NS, NP)	Improvements in Objective Function	Binary Solution
EX1B	3, 4, 5	43. // 6.7// 4.9// 2.7	57.// 1.6
EX2B	6, 4, 10	97. // 0. // 5.7// 3.6	3.// 2.5
EX3B	8, 5, 12	100. // 13.3// 5.2// 3.1	0.// 3.1
EX4B	12, 6, 15	100. // 66.7// 5.8// 0.8	0.// 4.4
EX5B	20, 7, 17	* // 100. // *// *	*
EX6B	30, 9, 20	* // 100. // *// *	*

When the optimal solutions of model ***SM*** and ***SS*** are compared, the MPC model presents cost improvements in most of instances, and the reduction in costs is a stable value of about 5%. The MPC policy is more efficient and minimizes transition times: production times increase, then shorter sizes are selected and costs are reduced.

When the number of products N increases, the necessity in production capacities is also increasing. Unfeasibility can occur if the machine with the largest size is not satisfying the demands within the time horizon. In Table 6, for examples EX5B an EX6B that are presenting the number of parameters in conformity with the real cases of chemical industry, the unfeasibility rules.

For model ***SM***, the development of heuristic or approximation procedures is found not essential because the reformulation using a TSP aggregated structure leads to LP's type solution and computational time (Birewar & Grossmann, 1989*a*) and (Birewar & Grossmann, 1989*b*). In plus: *i)* the cyclic scheduling in MPC has been computationally developed (Miranda, 2007); *ii)* the setup or cleanup times are addressed; and iii) in comparison with model SS, the results from SM showed about 5% reduction in investment cost.

However, some weaknesses are remaining in model SM:

- Unfeasibility occurs in a large number of instances, namely, when large values for N or Q are given;
- Integrality is not required for ni and Nchik.

In despite of the MPC policy, model ***SM*** showed that the single machine approach is quite limitative. The application onto real cases requires that multiple machines can be selected in each stage, as discussed in next section for model ***MM***.

5.2) Discussion of model MM

The model ***MM*** (multiple machine and MPC) searches the cheaper combination of machines and their discrete sizes to satisfy products demand Q, on the horizon, H, but considering the scheduling of several products in each production campaign (MPC) and multiple processes in parallel at each stage.

The model ***MM*** does not treat the slacks corresponding to the production transitions. Implicitly, it considers unlimited intermediate storage (UIS) instead of zero wait (ZW) policy. Then, there is a contradiction in the conditions assumed: MPC vs. zero production transitions. If the slack times are null, there is no necessity to optimize them using MPC.

This contradiction drives inconsistency on numerical results. That is, facing the same instances as ***SM*** (single machine and MPC), the model ***MM*** equally selects configurations of single machine but with smaller sizes and cheaper equipments. This occurs because the slack times are made zero.

To compare the MPC models, **SM** (single machine and MPC) and ***MM***, the number of processing machines is constrained to *NP = 1* and the transition times are made $SL_{ikj} = 0$, in the sense to adjust the instances in both models. Running two series of 30 random instances, the results obtained are equal. Beyond the accuracy of the computational implementation, the theoretic assumption that ***SM*** is a specific instance of ***MM*** (Miranda, 2007) is verified. Also, the optimal value of model ***SM*** drives an upper bound for the ***MM*** objective function, and it can be used as a cutting value.

Also the model MS (multiple machine and SPC) may be treated as a specific instance of the model MM, if the numbers of transitions Nchikj are satisfying equation 26. Then, equivalence relations occur between variables, objective function, and restrictions of these two models (Miranda, 2007).

Table 7 shows the number of binary and continuous variables, the number of restrictions, and the mean execution times for the ***MM*** numerical examples (B series).

Table 7. Numbers of variables and restrictions in the numerical examples of model ***MM*** (B series).

Examples	Binary (N, M, NS, NP)	Continuous Variables	Restrictions Variables	Restrictions	Time solver (s)
EX1B	3, 4, 5, 3	60	12	29	1.65
EX2B	6, 4, 10, 3	120	42	50	3.49

Table 7 presents only two examples, since results inconsistency is observed. The results from model ***MM*** were compared with the results from model ***SM***, and:

- The unfeasible instances in single machine are becoming feasible in the multiple machines, as expected;
- For the feasible instances in SM, the results from MM are better, and it was expected that results never be worse;
- However, when the solutions that require only one machine at each stage are compared, lesser costs are observed, since shorter sizes were selected for this single machine.

That is, facing the same instance, with equal values on parameters and data, when ***MM*** selects only one processing machine at each stage, it selects shorter sizes that are driving lesser costs. This incoherence can be explained since the available time for production is larger due to the nullification of transition times. In fact, ***MM*** will not treat the slack times in production transitions when it already assumed that these slacks are zero. Implicitly, model ***MM*** is considering unlimited intermediate storage (UIS) in contradiction with the ZW policy, as it was stated.

This contradiction is driven from a theoretic limitation, since the products sequencing in model SM (single machine and MPC) used the TSP aggregated structure that allows LP solving, as described by (Pekny & Miller, 1991). In ***MM***, it would be necessary to combine a TSP problem for each of the machines implemented in parallel, which seems not reasonable.

The model ***MM*** is the more enlarged formulation in analysis, since it generalizes both the machines number and products number in the production campaigns (MPC). The limited numerical runs are revealing concordance with the described theoretic considerations:

- In comparison with model SM, the costs are diminishing in MM;
- For all instances, feasibility rules;
- Integrality is not required for n_i and Nch_{ik};
- The setup or cleanup times are formulated, but not numerically treated;
- The implicit UIS costs and parameters are not considered.

Initially, ***MM*** was the model that presented better expectations, but in despite of the reduction in investment costs, the application of the corresponding solutions is not realistic. The inconsistence in the numerical results is due to the implicit contradiction in the operation mode, UIS vs. ZW, and then its application to real cases is not possible.

6. Conclusions

Several models aimed to simultaneously design and scheduling batch chemical processes are selected from literature, compared, and various heuristics are developed in order to obtain satisfactory solutions.

The selected models simultaneously address the scheduling of the production cycles embedded on the design problem of batch chemical processes, considering multiproduct environment (flowshop) and ZW storage policy. The models differ in the number of processes considered at each stage (single machine vs. multiple machine) and in the production policy (SPC *vs*. MPC).

In a combinatory way, four different models are studied (***SS, MS, SM,*** and ***MM***), their characteristics and limitations are referred, and the probabilistic analysis of the heuristics is performed. The main points of the analysis are:

i) For model ***SS***, several approximation procedures are built and integrated in a heuristic; a significant fraction of optimal solutions is obtained, with errors of about 1% in value;
ii) For ***MS*** (multiple machines and SPC), the solving difficulties are increasing; a heuristic is developed too, and a similar level of quality is obtained;
iii) For ***SM*** (single machine and MPC), the solver execution was efficient but it is assumed the integrality relaxation in the number of batches; when compared with the model ***SS***, the model ***SM*** predicts reduction of about 5% in the investment costs due to the MPC policy;
iv) The model ***MM*** allowed feasible solutions in realistic instances with high number of products or quantities that are not achievable through the previous model ***SM*** (single machine and MPC); however, theoretical contradictions are driving numerical incoherence; namely, using comparable data and constraining the number of machines to 1, the results from ***MM*** are significantly better than those from ***SM***, but they are not realistic.

From the comparative analysis performed, the combination of multiple machines with SPC was found to be the most promising from a computational standpoint. On a generalization approach, the model ***MS*** is thus selected toward a stochastic environment.

Acknowledgement

This work was supported in part by the Centre for Chemical Processes (CPQ) at the IST/UTL. The Author also thanks the support at ESTG/IPP.

Nomenclature

Index and sets

M	- number of stages *j*;
NC	- number of components or products *i*;
NP(j)	- number of processes *p(j)* per stage;
NS(*j*)	- number of discrete dimensions *s(j)* in the process of stage *j*;

Parameters

τ_{ij}	- processing times (h), for each product *i* in stage *j*;
c_{jsp}	- equipment cost of process *p(j)* and size *s(j)* in stage *j*;
dv_j	- discrete volume (equipment size) in each stage *j*;
H	- time horizon;
nc_{ijsp}^{Upp}	- upper bound on the number of batches of product *i*, disaggregated by process *p(j)* and size *s(j)* in each stage *j*;
p(j)	- (ordinal) number of processes in stage j;
Q_i	- demand quantities (uncertain) for each product *i*;
s(j)	- (ordinal) number of process discrete dimensions in stage *j*;
S_{ij}	- dimension factor (L/kg), for each product *i* in stage *j*;
V_j	- equipment volume (continuous value) in each stage *j*;

Variables

n_i	- number of batches of product *i*;
nc_{ispj}	- number of batches of product *i*, disaggregated by process *p(j)* and size *s(j)* in each stage *j*;
Nch_{ik}	- number of transitions (changes) from product *i* to *k*;
t_i	- production times of each product *i*;
y_{jsp}	- binary decision toward process *p(j)* and size *s(j)* in stage *j*;

Appendix - Estimators for analysis of the deviations

In the probabilistic analysis of heuristics, several estimators are evaluated and presented in Table 2 and Table 4. Namely, excluding unfeasible instances and feasible instances that are exactly estimated, the percentage errors between the corresponding values of the optimal objective function ("OPT" indexes) and the sub-optimal heuristic ("H" indexes) are evaluated using the expression:

A.1 $$\%dsv\_z = \frac{|z_{OPT} - z_H|}{z_{OPT}}.100.$$

Again excluding unfeasible instances and instances presenting exact binary solution, the numbers of binary errors (Ndsv_y) for the heuristics sub-optimal solutions are evaluated by:

A.2 $$Ndsv\_y = \sum_{j=1}^{M} \sum_{s=1}^{NS} \frac{|y_{OPT}(j,s) - y_H(j,s)|}{2}.$$

In Table 6, considering only the feasible and comparable instances, the percentage improvements in the corresponding values of the objective function in MPC ("M" indexes) and SPC ("S" indexes) are given by:

A.3 $$\%dsv\_z = \frac{|z_M - z_S|}{z_M}.100.$$

Again excluding the instances presenting the same binary solution, the numbers of binary deviations for the improved MPC instances, in relation to the SPC ones, are obtained from:

A.4 $$Ndsv\_y = \sum_{j=1}^{M} \sum_{s=1}^{NS} \frac{|y_M(j,s) - y_S(j,s)|}{2}.$$

References

Ahmed, S. and N. V. Sahinidis (2000). Analytical investigations of the process planning problem. *Computers and Chemical Engineering* **23**, 1605–1625.

Ahmed, S. and N. V. Sahinidis (2003). An approximation scheme for stochastic integer programs arising in capacity expansion. *Operations Research* **51**(3), 461–471.

Barbosa-Póvoa, A. P. (2007). A critical review on the design and retrofit of batch plants. *Computers and Chemical Engineering* (31), 833–855.

Biegler, L. T., I. E. Grossmann and A. W. Westerberg (1997). *Systematic Methods of Chemical Process Design*. Prentice-Hall, New Jersey.

Birewar, D. B. and I. E. Grossmann (1989*a*). Efficient optimization algorithms for zero-wait scheduling of multiproduct batch plants. *Ind. Eng. Chem. Res.* **28**, 1333–1345.

Birewar, D. B. and I. E. Grossmann (1989*b*). Incorporating scheduling in the optimal design of multiproduct batch plants. *Computers and Chemical Engineering* **13**, 141–161.

Cavin, L., U. Fischer, F. Glover and K. Hungerbhler (2004). Multi-objective process design in multi-purpose batch plants using a tabu search optimization algorithm. *Computers and Chemical Engineering* (28), 459–478.

Garey, M. R., D. S. Johnson and R. Sethi (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research* **1**(2), 117–129.

Jayaraman, V. K., B. D. Kulkarni, S. Karale and P. Shelokar (2000). Ant colony framework for optimal design and scheduling of batch plants. *Computers and Chemical Engineering* **24**, 1901–1912.

Miranda, J. L. (2007). Optimização em Sistemas de Processos Químicos: Generalização de Modelos com Planeamento e Sequenciamento. PhD dissertation. Instituto Superior Técnico, Technical University of Lisbon, Lisboa.

Pekny, J. F. (2002). Algorithm architectures to support large-scale process systems engineering applications involving combinatorics, uncertainty, and risk management. *Computers and Chemical Engineering* **26**, 239–267.

Pekny, J. F. and D. L. Miller (1991). Exact solution of the no-wait flowshop scheduling problem with a comparison to heuristic methods. *Computers and Chemical Engineering* **15**(11), 741–748.

Tan, S. and R. S. H. Mah (1998). Evolutionary design of noncontinuous plants. *Computers and Chemical Engineering* **22**(1/2), 69–85.

Voudouris, V. T. and I. E. Grossmann (1992). Industrial and engineering chemistry research. *Computers and Chemical Engineering* **31**, 1315–1325.