Open Archive Toulouse Archive Ouverte (OATAO)
OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://oatao.univ-toulouse.fr/
Eprints ID: 4999

To link to this article:
http://dx.doi.org/10.1016/j.compchemeng.2010.01.010

To cite this version : Baez Senties, O. and Azzaro-Pantel, Catherine and Pibouleau, Luc and Domenech, Serge. *Multiobjective scheduling for semiconductor manufacturing plants.* (2010) Computers & Chemical Engineering, vol. 34 (n° 4). pp. 555-566. ISSN 0098-1354

Any correspondence concerning this service should be sent to the repository
administrator: staff-oatao@inp-toulouse.fr

Multiobjective scheduling for semiconductor manufacturing plants

O. Baez Senties, C. Azzaro-Pantel*, L. Pibouleau, S. Domenech

Université de Toulouse, Laboratoire de Génie Chimique, UMR 5503, ENSIACET INPT, 4, allée Emile Monso - BP 44362, 31432 Toulouse, Cedex 4, France

A B S T R A C T

Scheduling of semiconductor wafer manufacturing system is identified as a complex problem, involving multiple and conflicting objectives (minimization of facility average utilization, minimization of waiting time and storage, for instance) to simultaneously satisfy. In this study, we propose an efficient approach based on an artificial neural network technique embedded into a multiobjective genetic algorithm for multi-decision scheduling problems in a semiconductor wafer fabrication environment.

Keywords:
Discrete event simulation
Artificial neural networks
Multiobjective genetic algorithm
Semiconductor manufacturing

1. Introduction

Scheduling of semiconductor wafer manufacturing system is identified as a highly complex job shop task, mainly because of the typical features of the process scheme, such as complex product flows (the so-called wafer fab is indeed a multipurpose plant), rapidly changing demands and product mixes with sometimes very brief product life cycles (Ellis, Lu, & Bish, 2004). It is thus a significant challenge to develop effective scheduling methods in wafer fabrication. Discrete event simulation (DES) is one of the most widely used tools to study, analyze, design, and improve manufacturing systems. The combined used of a DES and an optimization procedure based on a genetic algorithm was an efficient solution to short-term job-shop scheduling problems and was implemented in our research team (Charles, Floru, Azzaro-Pantel, Pibouleau, & Domenech, 2003).

In spite of its acknowledged benefits, this kind of approach often reaches its limits in the industrial practice because of the highly combinatorial nature of the problems. In addition, the main emphasis of much of the work on scheduling has been on the development of predictive methodologies with a single objective to optimize. A lot of scheduling techniques for semiconductor manufacturing have been stated in publications over the last years (Uzsoy, Lee, & Martin-Vega, 1992a; Uzsoy, Lee, & Martin-Vega, 1992b). Most approaches to the semiconductor manufacturing scheduling problem can be classified into four categories: heuristic rules, mathematical programming techniques, neighbourhood search methods, and artificial intelligence techniques. A thorough review is proposed in (Gupta & Sivakumar, 2006).

In industrial practice, wafer fabrication is a complex, dynamic and stochastic task in which satisfying multiple objectives might be more realistic than only optimally meeting a single criterion. Actually, production managers have to cope with various objectives, which contributes to scheduling complexity: meeting due dates is an important goal in low-volume and high variety production circumstances within competitive market environments. Another major objective in scheduling of semiconductor wafer fabrication is reducing waiting time for work in process (WIP) inventory to improve responsiveness to customers. In addition, the shorter the period that wafers are exposed to contaminants while waiting for process, the smaller the yield loss. Increasing the throughput is also an important stake, since the investment in fabrication equipment is capital intensive.

In this study, an approach based on an artificial neural network (ANN) technique coupled with a multiobjective genetic algorithm (MUGA) for multi-decision scheduling problems in semiconductor wafer fabrication, is proposed. Indeed, instead of using a classical DES procedure for simulating the manufacturing system, an ANN is preferred for response time reasons. Furthermore, the multiobjective optimization problem arising for the multi-decision scheduling problem is tackled with a genetic algorithm, insofar as this type of procedure is particularly well fitted to simultaneously handle several competitive objectives. Fig. 1 summarizes the proposed methodology, which will be presented in the following sections.

The paper is organized as follows. Section 2 is devoted to the general process description. Section 3 recalls the principles of discrete event simulation. Then, Section 4 presents the methodology for ANN modelling. In Section 5, the principles of the multiob-

* Corresponding author. Tel.: +33 5 34 32 33 00; fax: +33 5 34 32 33 99.
E-mail address: Catherine.AzzaroPantel@ensiacet.fr (C. Azzaro-Pantel).

Nomenclature

ACT	average cycle time (days)
AWT	average waiting time (min)
ANN	artificial neural network
B_i	bias
DES	discrete event simulator
FAU	facility average utilization (%)
$g(k)$	kth output of the ANN
MNR	maximum number of recipes
MSE	mean square error
MUGA	multiobjective genetic algorithm
NB	number of batches in the WIP
N_{bits}	number of bits in the binary code
NHN	number of hidden neurons
NSL	number of sequencing batches (batches)
N_{Te}	number of couples for the test phase
N_{Tr}	number of couples for the training phase
n	size of the data base
PEC	percentage of each family
R	coefficient correlation
RMSE	root square of mean square error
S	binary string coding the value of the chromosome
SD	standard deviation of the average cycle time (days)
TBB	time between batches (min)
TBC	time between campaigns (min)
Te	test
TP	total production (batches)
Tr	training
TS	total storage of batches
WIP	work in process
w	weight
x_i	input variable
$x_{\max}$	upper bound on the input variable x_i
$x_{\min}$	lower bound on the input variable x_i
x_{norm}	normalized value of the input variable x_i
$y(k)$	kth output of the ANN

jective optimization procedure are presented. Section 6 discusses some typical results in the case of bicriteria and tricriteria optimizations. Then, the conclusions and perspectives are reported in the last section.

2. Process description

In this investigation, a didactic example of small size, representing the behaviour of a real fab is considered. The process takes into account 14 different equipment units used (from which 12 are different) and involves 24 operating stages. To answer to the request of the market, the workshop has, for the majority of the manufacturing stages, several equipment units operating in parallel able to carry out the same operations. Two shared equipment units (units C and D) are used up to 6 times in the manufacturing sequence, to reproduce the so-called re-entrant flow, in which a similar sequence of processing step is repeated several times. Some units are arranged in parallel, which confers more flexibility to the process. The production of five different products with their associated recipes identified by an integer number (1–5) is considered. This workshop must satisfy a production varying from 4 to 5 products. It is described in Table 1, where the treatments for the five products and the stages of the manufacturing process.

For each stage, the membership zone, the name of the stage, the involved equipment and processing times of each product are specified. A processing time equal to zero means that the product recipe does not use the corresponding equipment. The equipment units E and F, like G and H, are identical equipment placed in parallel for carrying out the same operation.

A batch is constituted by 50 wafers and 3 levels of productions are to be considered according to market demand, i.e. 32, 64 and 96 batches (noted TP as total production). The example, which serves as an illustration of the procedure, is presented in Fig. 2 and mimics the manufacturing plant of a typical semiconductor process. It reproduces all the characteristics of a real production system.

3. Discrete event simulation

3.1. MELISSA simulator

The conventional search and optimization methods used for batch plant scheduling and reviewed in (Gupta & Sivakumar, 2006) are generally intensive in computation time as even the simple manufacturing scheduling problems are NP-hard. The complexity of scheduling problem increases more in semiconductor manufacturing because of the presence of different types of work-centers, very large and changing varieties of the products, re-entrant process flow, and contradicting multiple objective functions, etc. In modelling the scheduling problem of semiconductor manufacturing, the use of discrete event simulation method helps in overcoming many of the limitations of the conventional approaches (Sivakumar, 2001).

Let us recall that simulation modelling has become an absolutely essential tool to accurately assess the capacity of a facility. Simulation modelling has been widely used in the semiconductor industry for a wide range of strategic objectives such as fab design, equipment selection, capacity and cycle time planning, etc. (Min & Yih, 2003). Its use in the tactical area has been relatively limited. The involved areas include short-interval scheduling, dispatching of individual batches, equipment scheduling, short-term human

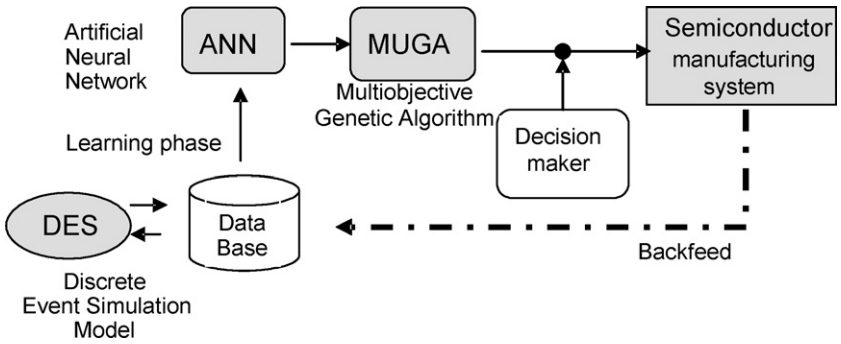

Fig. 1. General methodology.

Table 1
Processing times of the batches by product.

Phase	Zone	Equipment	Capacity [lots]	Process times by lot [min]				
				Product 1	Product 2	Product 3	Product 4	Product 5
1	Diffusion	A	1	120	120	120	120	120
2	Diffusion	B	4	0	0	1000	700	850
3	Photo	C	1	0	0	20	20	20
4	Engraving	D	1	0	0	15	15	15
5	Diffusion	E	2	500	200	500	400	300
6	Diffusion	F	2	500	200	500	400	300
7	Photo	C	1	20	20	20	20	20
8	Engraving	D	1	15	15	15	15	15
9	Diffusion	G	2	700	600	500	700	400
10	Diffusion	H	2	700	600	500	700	400
11	Photo	C	1	20	20	20	0	0
12	Engraving	D	1	15	15	15	0	0
13	Test	I	1	1	1	1	1	1
14	Diffusion	J	2	350	400	500	0	0
15	Photo	C	1	0	0	0	30	30
16	Engraving	D	1	0	0	0	20	20
17	Diffusion	K	2	0	0	0	400	500
13	Photo	C	1	30	30	30	30	30
19	Engraving	D	1	20	20	20	20	20
20	Engraving	L	1	140	180	200	0	0
21	Metal	M	1	120	120	120	120	120
22	Photo	C	1	20	20	20	20	20
23	Engraving	D	1	20	20	20	20	20
24	Metal	N	1	20	20	20	20	20

Note: 1 lot = 50 silicon plaquetes.

resource assignment, etc. To model the wafer fab in a high degree of detail, discrete event simulation (DES) techniques were previously implemented, leading to the development of the MELISSA software in collaboration with the Motorola Company in Toulouse, France (Peyrol, Pibouleau, Floquet, Couderc, & Domenech, 1993). Process planning simulations evaluate assignments of jobs to machines and routings for those jobs through the shop. Scheduling simulations seek solutions to daily issues including on-time order completion, priority changes, and unexpected changes in resource availability. Typical events taken into account and managed in the simulation core have been widely presented in Bérard et al. (1999) and will not be recalled here. The software tool was widely used and validated in previous works and reflects the behaviour of the plant. The different runs performed with MELISSA allow identifying the more sensitive variables on some performance criteria, such as facilities utilization, cycle time, limitation of WIP, etc.

MELISSA was developed to represent the path followed by the batches of products in a manufacturing plant of electronic components, in order to help the production manager, who has to cope with several difficult problems, such as multiproduct manufacturing on general-purpose equipment in order to organize the product flows and to use efficiently the available resources.

For MELISSA design, the following assumptions were used.

- The equipment units are reliable at 100% and are not subjected to preventive maintenances.
- The workshop runs 24 h a day and 7 days a week.
- The production is organized in periodic production campaigns: the planning horizon has been partitioned into a number of time periods (campaigns), each dedicated to the production of a subset of products; campaigns are assumed to be identical (same length and same set of products for each campaign).
- The batches of a given product are all of the same size.
- Storages have a sufficient capacity, in order not to introduce constraints of availability.
- In the same way, the initial levels of raw material storage are sufficiently important not to generate constraints of availability.
- All the batches are assigned to each recipe and are managed according to the same rule of priority, i.e. FIFO.
- The percentage of each family (a number of batches of each family to be processed in a campaign) can be equal to 12.5% or 25%, or

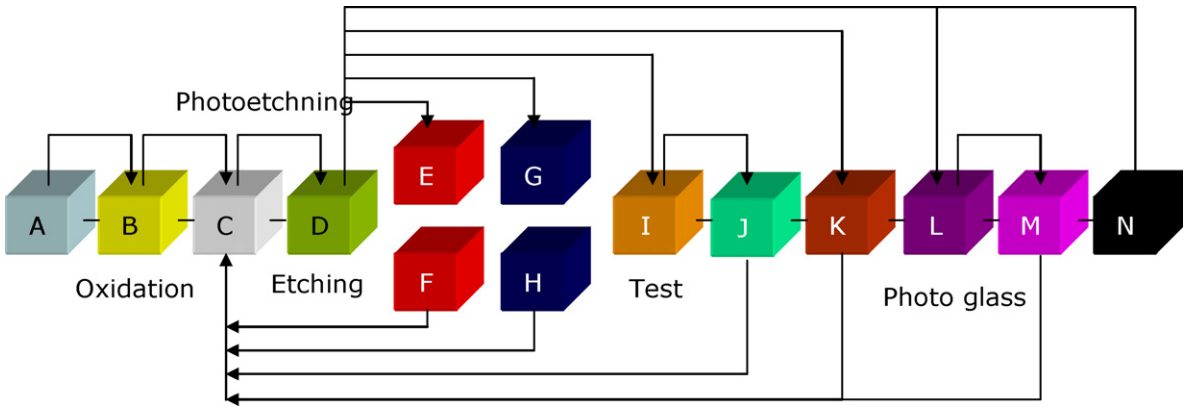

Fig. 2. Typical semiconductor plant.

Table 2
Decision variables.

Decision variable	Value
Maximum number of recipes	5: R_1, R_2, R_3, R_4, R_5
Maximum number of recipes treated simultaneously (MNR)	4 among the 5 possible recipes Ex: R_1, R_2, R_3, R_4
Percentage of each family (PEC)	3 combinations: 25% R_1, 25% R_2, 25% R_3, 25% R_4, 12.5% R_1, 37.5% R_2, 25% R_3, 25% R_4, 37.5% R_1, 12.5% R_2, 25% R_3, 25% R_4
Total production (TP)	3 discrete values: 32, 64, 96
Number of sequencing batches (NSL)	2 discrete values: 8 and 16
Time between campaigns (TBC)	10 discrete values (min): 1500, 2200, 2800, 3300, 3800, 4400, 5000, 5500, 6000, 6500
Time between batches (TBB)	Continuous value between 0 and TBC

37.5% (this variable is often called "mix of production").
- The quantities of products, both equipment and storage capacities are expressed in number of batches.
- The equipment units located in parallel are identical on all the aspects.
- The processing times for all the stages are deterministic.
- The loading/unloading times of the products are taken into account in the processing time.
- The "operator" resource is not modelled.

To sum up, the data, which are necessary for simulation purpose involve workshop architecture, product recipe, production data (simulation horizon, batch treatment priority, batch release order either imposed or fixed by heuristic rules, and heuristic rules for conflict management). Since the MELISSA discrete event simulator has been already presented in detail (Peyrol et al., 1991) only the key points that will be used for ANN design will be presented in what follows.

3.2. Input–output data of MELISSA

This section describes the data set, which will be used in the simulation for training the multi-layer perceptron (MLP) neural networks presented below.

3.2.1. Input data: decision variables

The input data for the simulation, which are the decision variables for the optimization phase, and their corresponding values are reported in Table 2.

3.2.2. Output variables: performance criteria

There are multiple criteria that can be used in evaluating the system performance and system status of the semiconductor fabrication. They are mainly based on inventory level, waiting time, or facility utilization.

Six criteria related to equipment and products were selected here:

1. Facility average utilization (FAU).
2. Average cycle time (ACT).
3. Standard deviation of the average cycle time (SD).
4. Average waiting time (AWT).
5. Number of batches in the WIP (NB).
6. Total number of batch storages (TS).

These criteria are computed according to the following relations:

$$FAU(\%) = \frac{\sum_{j=1}^{E} UT_j}{E} \times 100 \quad (1)$$

In this expression,

$$UT_j(\%) = \frac{NUE_j}{TBC} \times 100 \quad (2)$$

NUE_j corresponds to the net utilization time for equipment j and TBC is the campaign duration (time between two consecutive campaigns). UT_j corresponds to the net utilization time expressed in percentage of the campaign duration.

E corresponds to the total number of equipment items.

$$ACT = \frac{\sum_{j=1}^{NEL}(ED_j - ID_j)}{NEL} \quad (3)$$

ID is the input date of batch j, ED represents the exit date of batch j and NEL is the number of exit batches.

$$SD = \sqrt{\frac{1}{NEL}\left(\sum_{j=1}^{NEL} ACT_j{}^2\right) - ACT^2} \quad (4)$$

ACT_j corresponds to the average cycle time of batch j.

$$AWT = \frac{\sum_{j=1}^{TS} WT_j}{TS} \quad (5)$$

WT_j represents the waiting time for each batch that has been stored in the sequence. A batch may have been stored several times in front of an equipment unit of its processing recipe.

The criteria NB and TS are directly given by the simulator.

Of course, the time criteria are converted in the most representative time units in the final presentation of the results.

4. ANN modelling

4.1. ANN building

The embedding of the DES simulation model in an optimization procedure may require a lot of CPU time, which renders its use impractical for handling the problem of multi-decision scheduling of semiconductor manufacturing processes. For this reason, an artificial neural network (ANN) technique (Dreyfus et al., 2004) has been implemented to model the semiconductor plant, since its efficiency has been successfully demonstrated in semiconductor processing by many researchers (Min & Yih, 2003; Sung & Choung, 1999). The neural network stores the information in the strength of the neuron interconnection through the so-called weights. Because the weights of the links between the neurons cannot be predefined for neural networks used in real-life applications, the learning phase, in which all the examples are presented to the ANN repeatedly, is necessary to adjust the weights. The DES simulator is used for performing this learning phase. The multi-layer perceptron (MLP) neural network has been used in this study, since this kind of networks is typically used in supervised learning problems (see Fig. 3) with a back propagation (BP) algorithm scheme as a learning algorithm used to train multi-layer networks in a very general mode. Such networks can model functions of almost arbitrary complexity, with the number of layers, and the number of units in each layer, determining the function complexity. Important issues in multi-layer perceptrons (MLP) design include specification of the number of hidden layers and the number of units in these layers (Haykin, 1994). One hidden layer has been considered to be sufficient in this work to represent the DES behaviour.

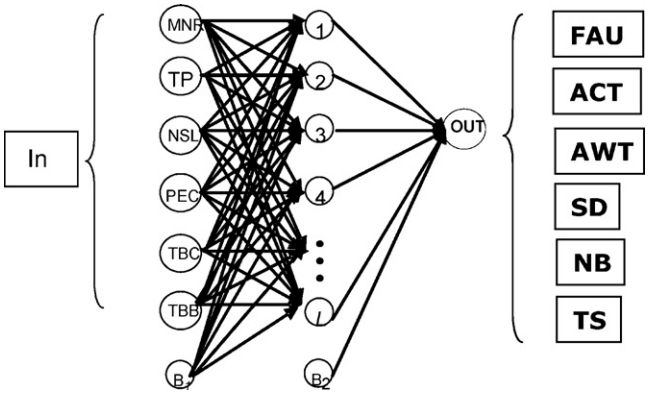

Fig. 3. Multi-layer perceptron neural network.

In Fig. 3, the input layer is made up by the decision variables (see Table 2), the intermediate layer is the hidden layer, and the performance criteria (see Section 3.2.2) are the outputs of the network. The terms B_1 and B_2 refer to biases of the network.

The choice of an adequate activation function proves to be an important component of the ANN's. Several types of activation functions can be classically used: some are linear, exponential, with threshold, Gaussian.... The activation function adopted in this study is the hyperbolic tangent that turns out to be ideal for customization of multi-layer perceptrons, particularly the hidden layers.

$$y = \tanh[B_i + \sum_{i=1}^{p} w_p x_p] \tag{6}$$

where B_i is the bias ($i = 1$ or 2), p is the number of neurons of the input layer or the hidden layer, x_i are the variables and w_i are the weights.

The implementation of a neural network begins with the construction of a data base and the choice of the samples. According to Miller, Freund, and Johnson (1990), the size n of the data base is given by the relation:

$$n = \frac{Z^2 pq}{d^2} \tag{7}$$

where Z is the normalized Gaussian variable corresponding to a probability of 95% ($Z = 1.96$), $p = q = 0.5$, and d is the target ($d = 2\%$). The value of n is truncated at 2250.

Data normalization is especially useful for modelling applications where the inputs are generally on different scales. A min–max normalization technique has been adopted in this study, since it has the advantage of preserving exactly all the relationships in the data and it does not introduce any bias. A linear interpolation formula such as that proposed in (8) has been used for rescaling: the values of input/output variables of the network are in the range [−1, 1]:

$$x_{\text{norm}} = 2\frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} - 1 \tag{8}$$

where $x_{\min}$ and $x_{\max}$ are the lower and upper bounds of variable x_i. For each decision variable, the bounds values are reported in Table 3.

During the training phase, the network adapts its structure (i.e. weights of connections) in order to obtain the desired values on its exit neurons. The data base for performing the training is built up by using the DES simulator. In the supervised training phase implemented here, after initialization of the weights of the network (in general with random values), couples of input (decision variables) and output (performance criteria) are extracted from the data base.

Table 3
Bounds on the decision variables.

Decision variable	Lower bound	Upper bound
MNR	1	5
PEC	1	3
TP	32	96
NSL	8	16
TBC	1500	6500
TBB	0	TBC

The error during the learning is the so-called root-mean square error (RMSE) and defined as follows:

$$\text{RMSE}_{Tr} = \sqrt{\text{MSE}_{Tr}} \tag{9}$$

with the mean square error (MSE) given by

$$\text{MSE}_{Tr} = \frac{1}{N_{Tr}} \sum_{k=1}^{N_{Tr}} (y(k) - g(k))^2 \tag{10}$$

RMSE gives a measure of the prediction accuracy, where N_{Tr} is the pattern number, that is the number of couples used for the training phase (we have chosen N_{Tr} equal to $2/3n$, that is to say N_{Tr} is equal to 1485), $g(k)$ represents the kth target value computed with MELISSA and $y(k)$ is the kth output computed by the ANN. The MSE_{Tr} minimization is performed by using the Levenberg–Marquart algorithm.

The process of MSE_{Tr} minimization does not guarantee that the training of the network is complete. An over-training of the network may occur when the network is trained excessively and/or the network architecture comprises a number of hidden neurons (NHN) more important than necessary (over-parameterization). There are various procedures to define the network architecture (Nandi, Ghosh, Tambe, & Kulkarni, 2001); the following two-phase procedure is used in this study.

Phase 1: Determination of the number of neurons for the hidden layer. Fix an iteration count (50 for example). Discretize the set of the possible values for the number of hidden neurons (for example between 10 and 100 with a step of 5). Determine the value of the number of neurons of the hidden layer from the minimum value of the error of training MSE_{Tr}. Improve this value while discretizing again more accurately around the value previously found and repeating the procedure.
Phase 2: Determination of the optimum number of iterations by using the number of hidden neurons obtained with phase 1. Use the same type of iterative procedure by discretizing the set of the possible values of the iteration count (for example between 50 and 800).

Once the network is determined (weights, number of hidden neurons and number of iterations), the last step is the test phase. The data not taken into account in the training phase, namely those constituting the data base of test (in this case they are $1/3n = 750$ couples) are then used to quantify the capacity of the network to extrapolate.

Fig. 4 illustrates the results of the training procedure for the facility average utilization (FAU) the optimal topology of the ANN corresponds to 12 hidden neurons. The error RMSE_{Te} has a similar definition than the RMSE_{Tr} (it is computed on the N_{Te} pairs used the in the test phase). Table 4 presents the results related to the number of hidden neurons for all the performance criteria considered.

For the FAU criterion, the number of hidden neurons being then fixed at 12, the determination of the optimal number of iterations is illustrated in Fig. 5. The best results were obtained for approximately 450 iterations. For the FAU performance index, the RSMSE

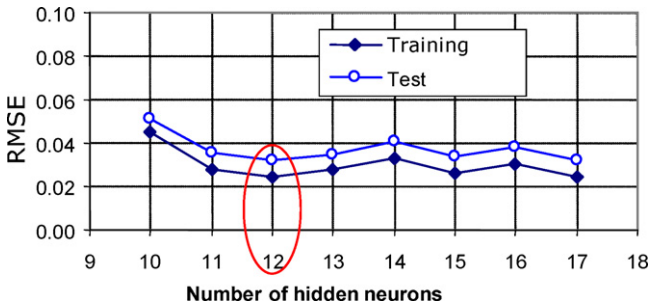

Fig. 4. RMSE for calculating the number of hidden neurons.

Table 4
Optimal topology of the ANN for each criterion.

Criterion	NHN	$RMSE_{Tr}$	$RMSE_{Te}$
FAU	12	0.02427	0.03257
SD	12	0.11546	0.14712
ACT	14	0.04358	0.06021
AWT	15	0.05250	0.08007
NB	18	0.03452	0.05330
TS	18	0.13212	0.18689

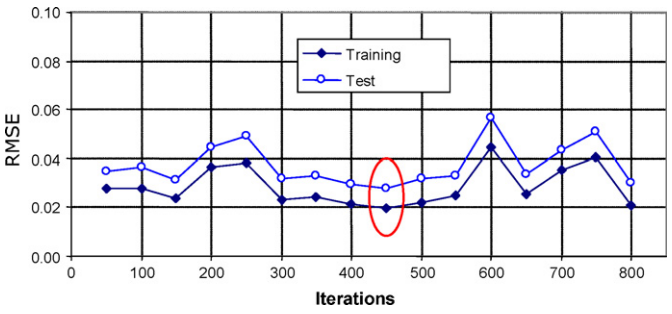

Fig. 5. RMSE for calculating the number of iterations.

is about 0.0242 for the training and 0.0325 for the test. The coefficients of correlation (R) for the training and the test are respectively equal to 0.99987 and 0.99791. The same process was employed to create various models with all the other criteria.

Once all the ANNs are determined for all the performance indexes listed in Section 3.2.2 (indeed one ANN is implemented for each criterion), they are used to obtain reliable responses to unspecified inputs. Table 5 involves the results for all the exits considered.

4.2. ANN validation

Fig. 6 compares the computed values by the ANN and those measured by MELISSA, for the FAU criterion.

For the other performance indexes, i.e. ACT, SD, AWT, NB and TS, the comparison between the computed and measured values are respectively reported in Figs. 7–11.

Table 5
Optimal number of iterations for each criterion.

Criterion	Iterations	$RMSE_{Tr}$	$RMSE_{Te}$
FAU	450	0.01989	0.02791
SD	700	0.09151	0.10062
ACT	550	0.04050	0.05951
AWT	500	0.05076	0.06907
NB	250	0.03315	0.04888
TS	650	0.10356	0.15399

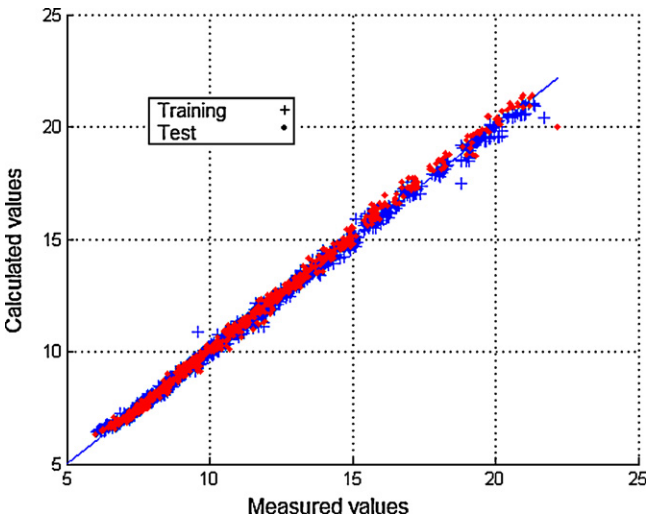

Fig. 6. Comparison between ANN and MELISSA for the FAU criterion [%].

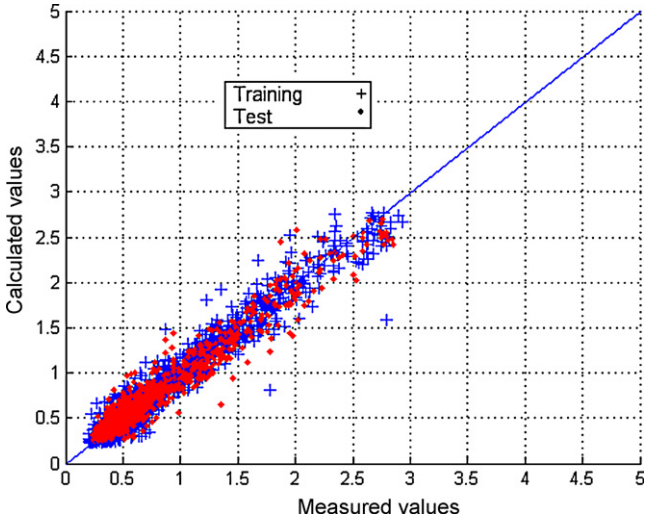

Fig. 7. Comparison between ANN and MELISSA for the ACT criterion [days].

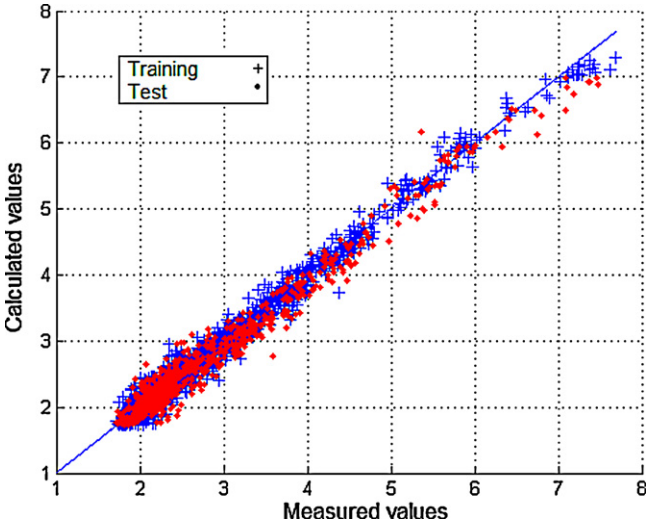

Fig. 8. Comparison between ANN and MELISSA for the SD criterion [days].

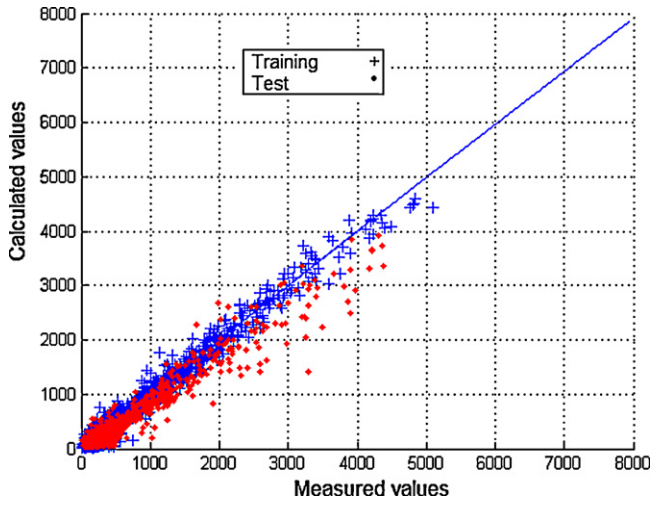

Fig. 9. Comparison between ANN and MELISSA for the AWT criterion [min].

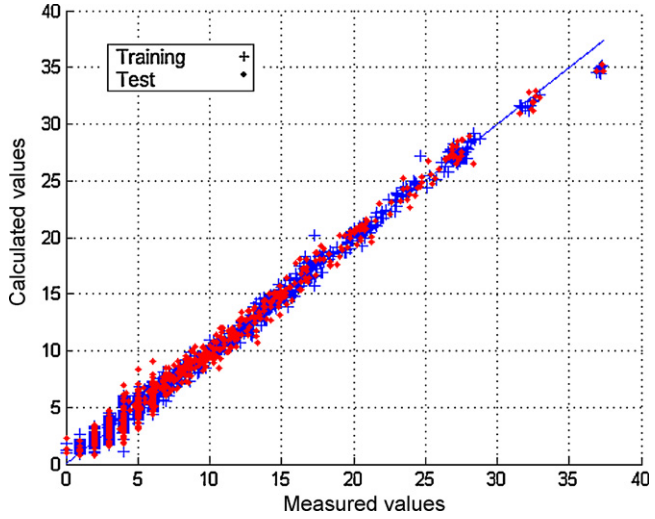

Fig. 10. Comparison between ANN and MELISSA for the NB criterion [batches].

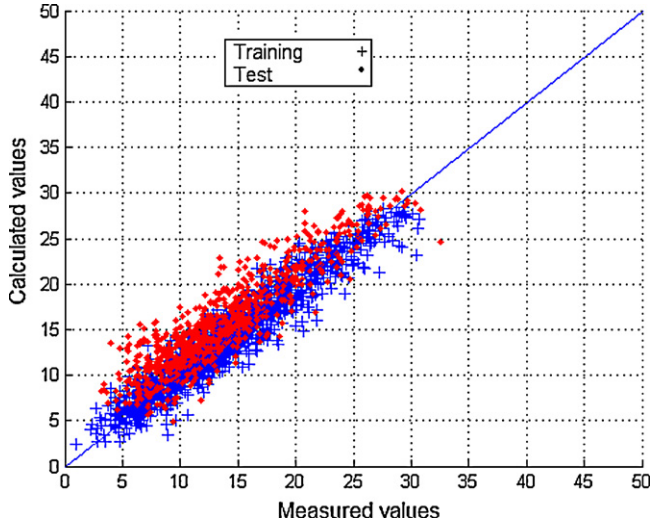

Fig. 11. Comparison between ANN and MELISSA for the TS criterion [batches].

Table 6
Fixed values for variables MNR, PEC, TP, NSL.

Variable	Fixed value
MNR	R_1, R_2, R_3, R_4 (R_i represents the ith recipe)
PEC	25% R_1, 25% R_2, 25% R_3, 25% R_4
TP	64 batches
NSL	8 batches

These figures emphasize the good agreement between the computed and measured values, and so the effectiveness of the ANN's to model these indexes of performance. In terms of computational times, the use of an ANN instead of the simulator MELISSA reduces the time by a factor near of 10. These ANN based models can be now integrated in an optimization procedure in order to determine the optimal values of the decision variables to satisfy multiple production targets.

5. Multiobjective optimization

Most of the optimization problems resulting from the real world are multiobjective optimization problems, because it is rare in practice to be able to determine a perfect solution from all points of view. The subject of multiobjective optimization has been extensively published (for instance, Collette & Siarry, 2003; Sawaragi, Nakayama, Tanino, 1985). In general, there may not be a particular optimal solution to a multiobjective problem, as one objective function gains only at the deterioration of the other objectives, due to their conflicting nature: optimality is thus an illusion when the objectives are conflicting. Therefore, one must be satisfied with obtaining the Pareto optimal solutions. Let us recall that a Pareto optimal solution corresponds to a solution in which no decrease can be obtained in any of the objectives without causing a simultaneous increase in at least one of the other objectives. A Pareto optimal solution is also called as non-dominated. The solution x^* is efficient to the problem defined if and only if there does not exist any $x \in S$ such that $f_j(x) \leq f_j(x^*)$ for all j and $f_j(x) < f_j(x^*)$ for at least one j.

Among multiobjective optimization techniques reported in (Collette & Siarry, 2003), the major advantage of genetic algorithms over other methods, particularly over other stochastic procedures such as Simulated Annealing, is that a GA manipulates not only a single individual but a population of individuals. The use of GA is all the more justified here as decision variables are discrete and as the criteria are evaluated by the DES simulator (the investigation of the mathematical properties of the functions is not required).

In this section, the methodology of resolution of multiobjective optimization problems is based on multiobjective genetic algorithms (MUGA) combined with the ANNs previously presented, used to compute the fitness according to the various performance criteria to be optimized.

In order to obtain a problem with a reasonable combinatorial aspect, the decision variables selected for optimizing the various objectives are the time between campaigns (TBC) and the time between batches (TBB_i, where i refers to the ith batch of the campaign). The range of variation of these two variables is reported in Table 3. The other input variables (see Section 3.2.1 and Table 2) are fixed during the optimization phase (see Table 6).

The MUGA is implemented for optimizing the decision variables and to deal with the set of compromise solutions for the studied criteria, thus giving the optimal Pareto zone solutions (Baez, Azzaro-Pantel, Pibouleau, Domenech, & Davin, 2005).

Recently, a large development of different types of multiobjective genetic algorithms appeared in the literature. A quite general guideline was proposed by Dietz, Azzaro-Pantel, Pibouleau, and Domenech (2005), Dietz, Azzaro-Pantel, Pibouleau, and Domenech

(2006) for implementing a MUGA, and this approach has been adopted in this work, where the code is developed by using the C++ language.

In genetic algorithm-based optimization, the initial population constitutes a basic point of the search, insofar as the later efficiency of the algorithm is closely related to quality of the first generation of individuals. The three classical methods for generating the initial population are the random initialization, the use of heuristic or a combination of the two techniques. The strategy chosen here is the random generation of the chromosomes. This method has the advantage of proposing a varied population, ensuring a good mapping of the search space.

In addition to the generation of the initial population, the size of populations (the initial one and the following populations, kept constant during the search) is an important feature for the success of a genetic algorithm. A population of too small size will not be able to evolve satisfactorily, because a bad solution will have a very important influence on the average fitness of the whole population. One the other hand, large sized populations may induce too high computational times. So, the size of populations was fixed at 100 individuals, which is a classical trade-off between an efficient searching process and the avoidance of pre-mature convergence.

A genetic algorithm operates on numerical chromosomes, in order to compute the fitness for performing genetic operators. Procedure encoding aims at representing chromosomes in the form of chains of bits containing all the necessary information. Several codes are available: absolute code, coding of Gray and natural binary code, which is most frequently used (Dagli & Sittisathanchai, 1995). It is the latter which was adopted in the algorithm implemented in this study. The determination of the coded variables for the MUGA procedure was carried out by using the following relation (11) for converting each chromosome k:

$$x_k = x_{\min,k} + \frac{(x_{\max,k} - x_{\min,k})}{2^{N_{\text{bits}}} - 1} S_k \tag{11}$$

where x_k is the real value of the variable TBB or TBC, $x_{\min,k}$ and $x_{\max,k}$ the lower and upper bounds (see Table 6), S_K the binary string representing the variable and N_{bits} the number of bits used for the binary code. For obtaining the desired degree of accuracy (see below), the value of N_{bits} was fixed at 8. For example, the binary string:

$S_k = [0\,1\,0\,0\,1\,1\,0\,0]$ represents a value of *TBC* equals to:

$$TBC = x_k = 1500 + \frac{(6500 - 1500)}{255} S_k$$
$$= 1500 + \frac{(6500 - 1500)}{255} 76 = 2990.19 \text{ min} \tag{12}$$

For the variable *TBB*, S_k codes the value:

$$TBB = x_k = 0 + \frac{(TBC - 0)}{255} S_k = 0 + \frac{(2990.19 - 0)}{255} 76 = 891.19 \text{ min} \tag{13}$$

For *TBC*, the accuracy of coding is given by:

$$A_{TBC} = \frac{(6500 - 1500)}{255} = 19.60 \text{ min} \tag{14a}$$

Similarly the lower and upper accuracies for TBB are the followings:

$$A_{LTBB} = \frac{1500}{255} = 5.88 \text{ min} \tag{14b}$$

$$A_{UTBB} = \frac{6500}{255} = 25.49 \text{ min} \tag{14c}$$

The phase of evaluation consists in calculating the fitness of adaptation of each individual of the population. In this work, the output of an ANN is used as the input to the GA: this is quite simple here since the output of the ANN is directly an evaluation function used in the GA The genetic algorithm maximizes this function during the successive populations to lead to a population adapted very well. In this study, the classical formulation of the fitness was chosen:

$$F(x) = C_{\max} - C(x) \tag{15}$$

where $C(x)$ represents an objective function (i.e. FAU, ACT, SD, AWT, NB, TS computed by using the corresponding ANN) and $C_{\max}$ will have to be selected so that the value of the firness remains always positive. It can be, for example, the greatest actual value of $C(x)$, either on the current population, or since the beginning of the search.

Four typical genetic operators were used to alter the composition of the offspring in the next generation, based on the pre-set probability values reported in Table 7. More details concerning genetic operators can be found in Dietz et al. (2005, 2006). These operators are the followings.

Table 7
Genetic algorithm parameters.

Population size	100
Number of generations	250
Selection probability	0.5
Mutation probability	0.1

1. *Selection operator*: Based on the ranked fitness values for each criterion, the selection of surviving individuals in the new population is made up by using the classical biased Golberg Wheel method (Goldberg, 1994). For each criterion, a particular Goldberg Wheel is used for carrying out the selection.
2. *Crossover operator*: Recombination of individuals is carried out to complete the new population. The crossover is a classical 1-point crossover.
3. *Mutation operator*: This genetic operation is performed by replacing a bit randomly selected its complementary binary value.
4. *Elitism*: The elitism is used to avoid the loss of the best current solution during the jump to a generation to the following one. The best solution of the current generation systematically replaces the worst solution in the following generation.

The above steps are repeated cyclically and the algorithm is stopped when a fixed number of generations is reached. This number must be sufficient to allow a correct scanning of the search space, but not too large not to induce too high computing times.

When the search is stopped, a procedure of sorting (Pareto's sort) is used for all the individuals evaluated during the search in order to identify all the individuals giving the set of Pareto-optimal solutions. These sets are known as Pareto's fronts. The differentiation between several Pareto-optimal solutions goes beyond the limits of this study and is left with the appreciation of the decision maker himself. Fig. 12 sums up the procedure for the ANN/MUGA strategy.

6. Results and discussion

6.1. Bicriteria optimization

The purpose of the present study is the simultaneous optimization of all the criteria taken by pairs. The results presented are those obtained after the final sorting of Pareto, applied to the whole of the solutions scanned during the search. Three runs of the MUGA were implemented for each combination of pairs of objective functions.

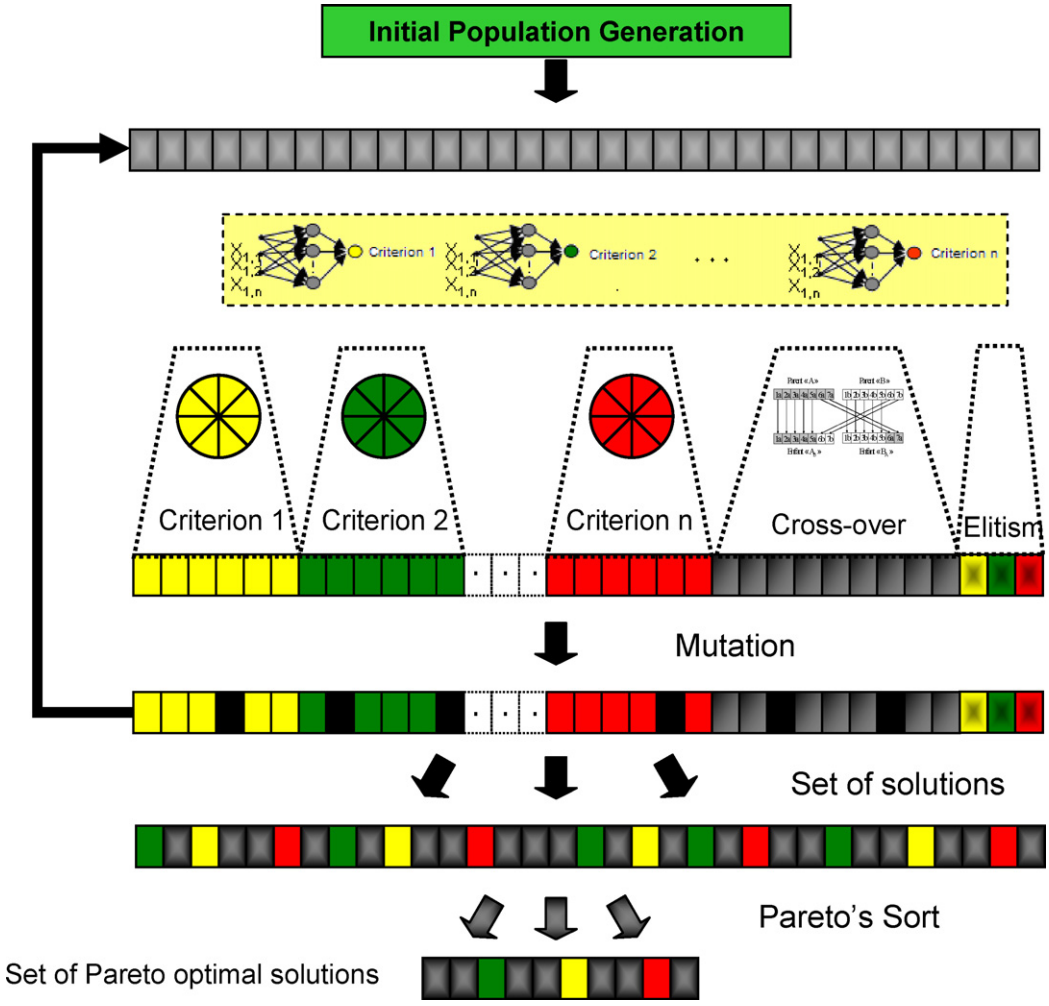

Fig. 12. Combination of the neural network and the multiobjective genetic algorithm.

6.1.1. Facility average utilization (FAU)—average cycle time ACT

The reduction of the average cycle time (ACT) contributes to a product manufacturing within the shortest processing time. Nevertheless, the maximization of the facility average utilization (FAU) constitutes an important contradictory force. Fig. 13 shows all the Pareto's solutions (153), presenting an important variation of the Average cycle time (several days).

In addition to the antagonistic effect of the two criteria considered, it appears that the higher the duration of campaigns is, the less the facility average utilization. This way of operating avoids the bottlenecks, which induces a less long average cycle time. In addition, about 70% of the solutions correspond to a TBB value lower than 3000 min.

6.1.2. Facility average utilization (FAU)—standard deviation of the average cycle time SD

The consideration of these two criteria is justified by their antagonistic behaviour. The results are presented in Fig. 14. The total number of compromise solutions found is equal to 62.

The figure shows that it is impossible to increase FAU while decreasing in same time SD. One can note an isolated point which corresponds to the monoobjective optimization of FAU: the near-

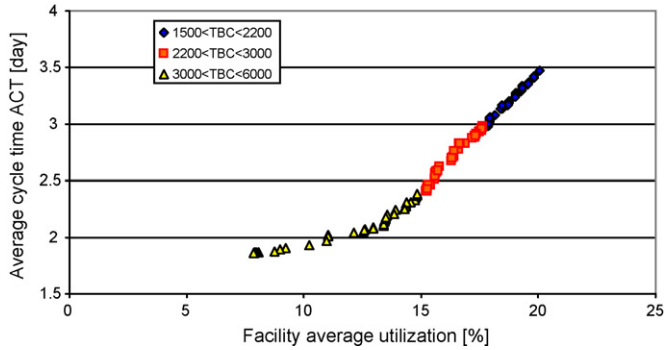

Fig. 13. Pareto optimal solutions for FAU-ACT criteria.

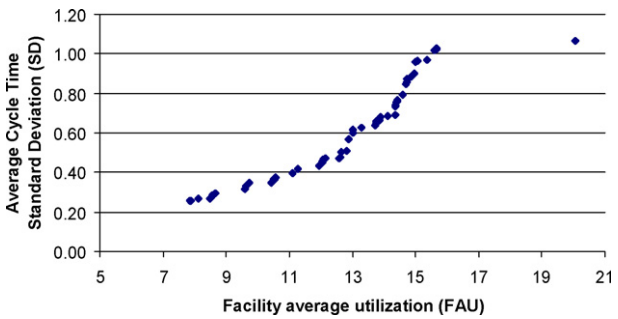

Fig. 14. Pareto optimal solutions for FAU-SD criteria.

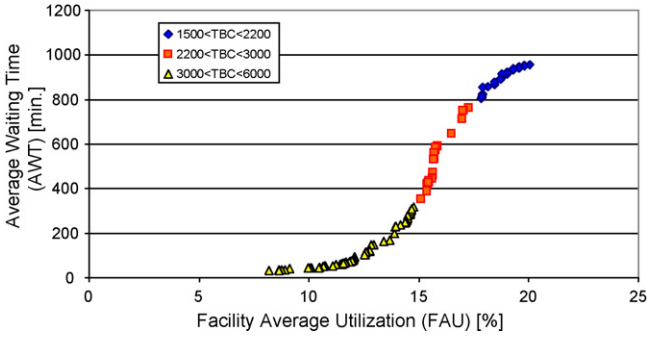

Fig. 15. Pareto optimal solutions for FAU-AWT criteria.

est point has an associated very close value for SD, whereas the variation in term of FAU is very important.

6.1.3. Facility average utilization (FAU)—average waiting time AWT

Fig. 15 highlights an existing contradictory behaviour between the two criteria considered, as it can be expected. These criteria are of the highest importance to avoid any form of exposure to the pollution of the silicon wafers during manufacture.

As previously, the Pareto front resulting from three runs of the MUGA can be analyzed according to the values of the campaign durations TBB. The inflection point of the curve can be explained by the existence of a plateau for the FAU. Indeed, when the FAU lies between 15 and 17, the corresponding AWT passes from 350 to 800 min, that is to say an increase of almost 230%. It is thus clear that the solution (FAU = 15) is of much better quality than that corresponding to a FAU equals to 17. The three zones of the Pareto front highlight three different behaviours according to the values of the campaign duration TBB. The number of Pareto solutions is 136.

6.1.4. Facility average utilization (FAU)—number of batches in the WIP (NB)

The Pareto front of Fig. 16 highlights the existence of an antagonism between the two criteria. The number of solutions is 103.

6.1.5. Facility average utilization (FAU)—total number of batch storages (TS)

The last combination of pairs of objectives relates to the criteria representing the number of stored batches. The 162 Pareto-optimal solutions can be observed in Fig. 17.

6.2. Tricriteria optimization

This last study concerning the tricriteria optimization allows a more complete consideration of the various compromises to be carried out between the various criteria, and thus provides more promising solutions among which the decision maker will be able to carry out a final selection.

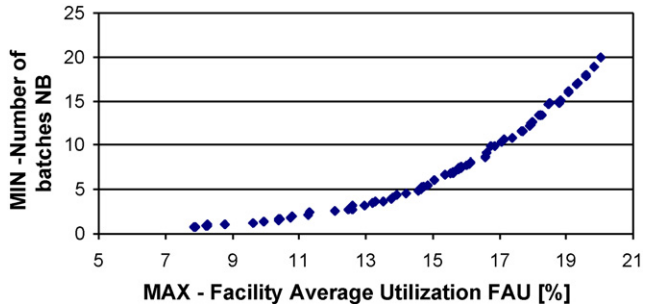

Fig. 16. Pareto optimal solutions for FAU-NB criteria.

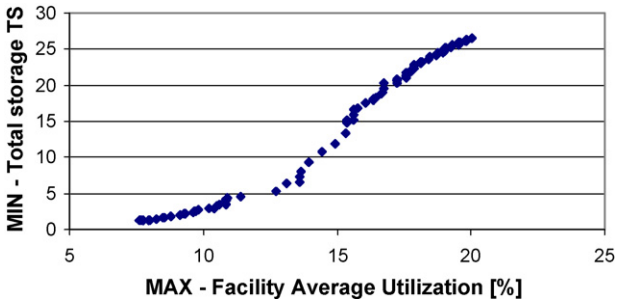

Fig. 17. Pareto optimal solutions for FAU-TS criteria.

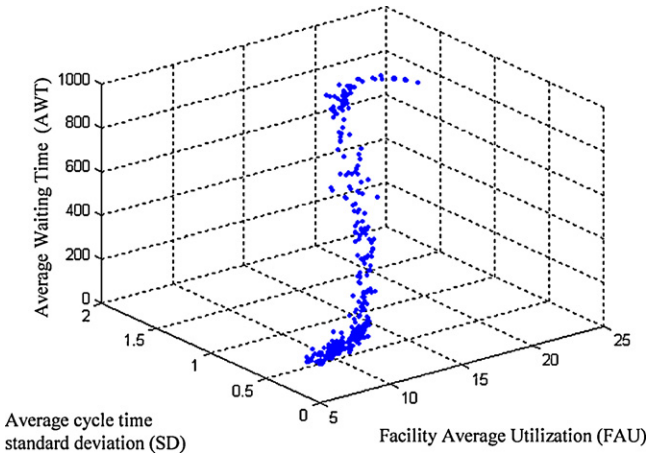

Fig. 18. Pareto optimal solutions for FAU-SD-AWT criteria.

6.3. Facility average utilization (FAU) – standard deviation of the average cycle time SD – average waiting time AWT

The conflict between the minimization of criteria AWT and SD and the maximization of the use of the equipment FAU, can be observed in Fig. 18. The trends highlighted in the bicriteria optimization section can be transposed here. Indeed, the influence of the campaign duration causing the inflection points previously observed in the bicriteria part, gives place in a three-dimensional space, to the shape of spiral appearing in Fig. 18. The number of solutions is 434.

The projections on two two-dimensional spaces are presented in Figs. 19 and 20. They illustrate the influence of a third criterion on the Pareto front. For a FAU equals to 17%, the optimal solutions tend to decrease, because of the presence of a third criterion.

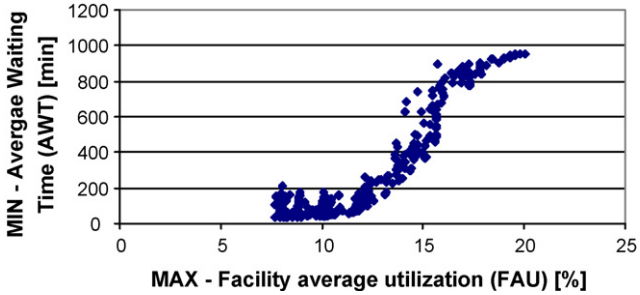

Fig. 19. Pareto optimal solutions for FAU-AWT criteria.

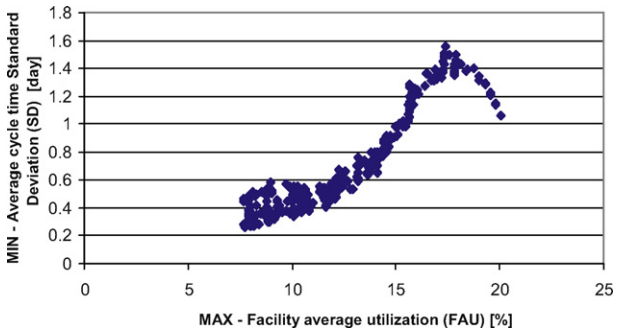

Fig. 20. Pareto optimal solutions for FAU-SD criteria.

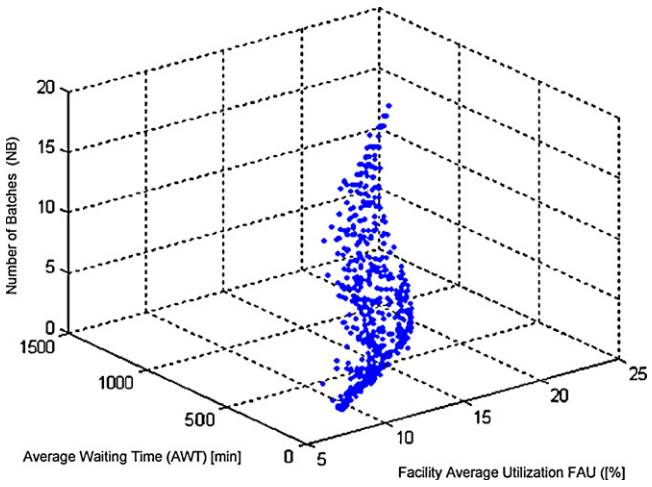

Fig. 21. Pareto optimal solutions for FAU-AWT-NB criteria.

6.3.1. Facility average utilization (FAU) – average waiting time AWT – number of batches in the WIP (NB)

For this objective function combination, no inflection point was observed for the Pareto front in the bicriteria optimization FAU-AWT, thus, the shape in spiral does not appear here. The number of solutions reported in Fig. 21 is 780.

6.3.2. Facility average utilization (FAU) – average cycle time ACT – number of batches in the WIP (NB)

The conclusions drawn from the combination of the three precedent criteria are applicable here (see Fig. 22). The number of solutions is 558.

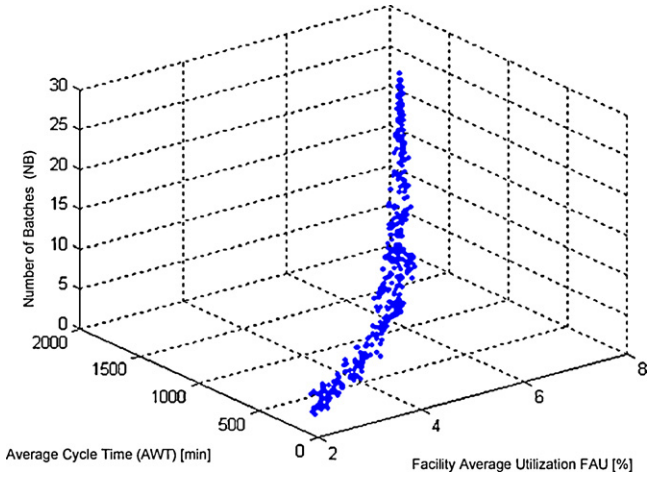

Fig. 22. Pareto optimal solutions for FAU-ACT-NB criteria.

Finally, in conclusion on this tricriteria optimization section, it can be pointed out that, even if some minor differences appear due to local inflection points for the bicriteria Pareto fronts, the general trends of surfaces in three dimensions are very similar. The antagonism of the five criteria (ACT, SD, AWT, NB and TS) versus FAU let suggest that these optimizations are sufficient to determine total compromise solutions, without having to carry out other calculations with more criteria to be optimized together.

7. Conclusions and perspectives

Efficient scheduling of wafer fabrication process involves an important number of decisions, leading to quite complex multiobjective optimization problems. The objective of this paper is to propose an optimization strategy in order to assign appropriate values to decision variables. More precisely, a scheduler for the selection of decision variables in order to obtain desired performance indexes at the end of a given production horizon is developed. In the proposed methodology, the combined use of a discrete event simulation technique, a neural network and a multiobjective genetic algorithm is suggested, to simultaneously optimize several objectives for the wafer fab. In terms of computational times, the use of an ANN instead of the DES simulator MELISSA reduces the computing time by a factor near of 10. The ANN/MUGA hybrid method constitutes an efficient tool for decision-making aid in the short-term scheduling of the manufacturing of electronic components. A didactic example of small size, representing the behaviour of a real fab is considered. Six criteria related to the equipments (FAU: facility average utilization) and the products (ACT: average cycle time, SD: standard deviation of ACT, AWT: average waiting time, NB: number of batches in the WIP and TS: total storage) were chosen as performance indexes of the workshop.

In a first step, the criteria are simultaneously optimized by pairs (FAU versus each performance index linked to the products). The Pareto fronts obtained from the MUGA provide for each pair a set of optimal solutions. However, the differentiation between several Pareto-optimal solutions goes beyond the limits of this study and is left with the appreciation of the decision maker himself. Then, in the last part of the paper, three objectives are simultaneously optimized (FAU versus a pair of criteria related to the products). The obtained three-dimensional Pareto fronts allow the decision maker to refine its decision about the short-term scheduling of the manufacturing process.

The methodology reported can be applied easily to the complex job shop scheduling problems such as in semiconductor manufacturing and significant benefits can be achieved in terms of cycle time distribution, facility average utilization, average waiting time and storage. Pareto optimal solutions can be consistently achieved in dynamic manufacturing environment, using the proposed approaches. The limitation of the approach lies in the necessary update, if a retrofit of the plant is implemented. The solutions proposed here are only valid within the production range related to parameters considered in this study. It must be yet said that the update can be performed in parallel without penalizing the global time in order to increase the model validity.

References

Baez, O., Azzaro-Pantel, C., Pibouleau, L., Domenech, S., & Davin, A. (2005). Aide à la décision pour la supervision d'un atelier discontinu: Application à la fabrication de composants électroniques. *Récents Progrès en Génie des Procédés*, *92* [Toulouse].

Bérard, F., Azzaro-Pantel, C., Pibouleau, L., Domenech, S., Navarre, D., & Pantel, M. (1999). Towards an incremental development of discrete-event simulators for batch plants: Use of object-oriented concepts, comm. escape 9, Budapest. *Computers & Chemical Engineering Supplement*, S565–S568.

Charles, A. S., Floru, I. R., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2003). Optimisation of preventive maintenance strategies in a semiconductor batch plant. *Computers & Chemical Engineering Journal*, *27*, 449–467.

Collette, Y., & Siarry, P. (2003). *Multiobjective optimization principles and case studies*. Springer–Verlag GmbH.

Dagli, C. H., & Sittisathanchai, S. (1995). Genetic neuro-scheduler: A new approach for job-shop scheduling. *International Journal of Production Engineering*.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2005). A framework for multiproduct batch plant design with environmental considerations: Application to protein production. *Industrial Engineering and Chemistry Research*, *44*, 2191–2206.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2006). Multiobjective optimization for multiproduct batch plant design under economic and environmental considerations. *Computers & Chemical Engineering Journal*, *30*, 599–613.

Dreyfus, G., Samuelides, M., Martinez, J., Gordon, M., Badran, F., Thiria, S., & Hérault, L. (2004). *Réseaux de neurones—Méthodologies et applications*. Eyrolles.

Ellis, K. P., Lu, Y., & Bish, E. K. (2004). Scheduling of wafers test processes in semiconductor manufacturing. *International Journal of Production Research*, *42*, 215–242.

Goldberg, D. A. (1994). Algorithmes génétiques. *Addison-Welsley*.

Gupta, A. K., & Sivakumar, A. I. (2006). Job shop scheduling techniques in semiconductor manufacturing. *International Journal of Advanced Manufacturing Technology*, *27*, 1163–1169.

Haykin, S. (1994). *Neural networks. A comprehensive foundation*. New York, NY: Macmillan.

Miller, I., Freund, J. E., & Johnson, R. A. (1990). *Probability and statistics for engineers*. Prentice-Hall.

Min, H. S., & Yih, Y. (2003). Development of real-time multiobjetive scheduler for a semiconductor fabrication system. *International Journal of Production Research*, *41*, 2345–2364.

Nandi, S., Ghosh, S., Tambe, S. S., & Kulkarni, B. D. (2001). Artificial neural network assisted stochastic process optimization strategies. *AIChE Journal*, *47*, 126–141.

Peyrol, E., Pibouleau, L., Couderc, J. P., & Domenech, S. (1991). Semiconductor circuit fabrication plant management by discrete simulation. In W. Fichtner, & D. Aemmer (Eds.), *Simulation of Semiconductor Devices and Processes Vol.4* (pp. 571–577). Zurich (Switzerland): Hartung-Gorre. September 12-14.

Peyrol, E., Pibouleau, L., Floquet, P., Couderc, J. P., & Domenech, S. (1993). *Simulation par événements discrets d'un atelier de fabrication de composants électroniques, Vol. 176*. Entropie., pp. 3–16.

Sawaragi, Y., Nakayama, H., & Tanino, T. (1985). *Theory of multiobjective optimization*. Orlando, Florida: Academic Press Inc.

Sivakumar, A. I. (2001). Multiobjective dynamic scheduling using discrete event simulation. *International Journal of Computer Integrated Manufacturing*, *14*(2), 54–167.

Sung, C. S., & Choung, Y. I. (1999). A neural approach for batching decisions in wafer fabrication. *International Journal of Production Research*, *437*, 3101–3114.

Uzsoy, R., Lee, Ch., & Martin-Vega, L. A. (1992a). A review of production planning and scheduling models in the semiconductor industry part I: System characteristics, performance evaluation, and production planning. *IIE Transactions*, *244*, 47–60.

Uzsoy, R., Lee, Ch., & Martin-Vega, L. A. (1992b). A review of production planning and scheduling models in the semiconductor industry part II: Shop-floor control. *IIE Transactions*, *265*, 44–55.