

Research Article

Hamzaoui YE. J Pharma Pharma Sci: 4: 184.
DOI: 10.29011/2574-7711.100084

Understanding Machine Learning Algorithms into Multiproduct Batch Plant Design of Protein Production

Youness El Hamzaoui*

Faculty of Engineering, Autonomous University of Carmen, Mexico

*Corresponding author: Youness El Hamzaoui, Faculty of Engineering, Autonomous University of Carmen, Mexico

Citation: Hamzaoui YE (2020) Understanding Machine Learning Algorithms into Multiproduct Batch Plant Design of Protein Production. J Pharma Pharma Sci: 4: 184. DOI: 10.29011/2574-7711.100084

Received Date: 15 January, 2020; **Accepted Date:** 24 January, 2020; **Published Date:** 31 January, 2020

Abstract

The problem of optimization of Multiproduct Batch Plant Design (MBPD) in chemical engineering systems where the design variables are the size of the equipment elements and the setting of operating conditions. The application is a multiproduct batch plant for the manufacture of four recombinant proteins as insulin, chymosin, vaccine and protease. However, addressing an important class of optimization problems handed over the serious combinatorial aspect of the complication. The procedure implemented consists in using machine learning algorithms, in order to minimize the investment cost and find out the number and size of parallel equipment units in each stage. The calculation results (investment cost, number and size of equipment, computational time, CPU time, idle times in plant, production rate, annual demand rate, setup cost, holding cost, variable cost, selling price, inspection rate, return cost, penalty cost, screening cost) obtained by metaheuristics machine learning tools are better than mixed integer no linear programming. This approach can facilitate the manufacturers of pharmaceutical drug to get an optimal design and makes up a remarkably suggested plan for having a benefit of efficient results.

Keywords: Batch Plant Design; Chemical Engineering Optimization; Machine Learning Algorithms; Mathematical Modeling

Introduction

Pharmaceutical researchers and biotechnology companies are devoted to developing medicines, such as: therapeutic proteins, human insulin, vaccines for hepatitis, food grade protein, chymosin detergent enzyme, and cryophilic protease. This allows patients to live longer, healthier, and more productive. Within this context, there is a high degree of consensus in the biomanufacturing industry that product quality, customer service, and cost efficiency are fundamental for success. The pharmaceutical industry must join its effort with government and the health professions to seek new, innovative, and cost effective approaches in the development process. However, the pharmaceutical process is characterized by a membership function involved in the field of chemical engineering.

Nevertheless, for understanding a chemical engineering system, we have to go back to the mathematical modeling, as we know the mathematical modeling is a powerful tool to

solve different problems which arise in chemical engineering optimization. Problems as designing a plant, determining the number of units for a specific task, assigning raw materials to different production processes and deciding the production planning or production targets are some of the issues that can be solved through mathematical modeling. In other words, the mathematical formulations are used to make decisions at different levels, from the synthesis and design of the process up to its operation and scheduling [1]. In spite of that, precisely in recent years, there has been an increased interest development of systematic method for the design of batch process in chemicals, food products, and pharmaceutical industries. Basically, batch plants are composed of items operating in a discontinuous way. Each batch then visits a fixed number of equipment items, as required by a given synthesis sequence so called production recipe. That means, the design of batch plants requires involving how equipment may be utilized. In addition, the optimal design of a multiproduct batch chemical process involves the production requirement of each product and the total production time available for all products has been considered. The number and size of parallel equipment units in each stage as well as the location and size of intermediate storage are to be determined in order to minimize the investment cost.

Many works in the literature on batch process design are based on expressions that relate the batch sizes linearly with the equipment sizes [2], made a comprehensive framework for optimal design of batch plants. Dietz, Azzaro-Pantel, Pibouleau & Domenech [3], developed an approach of multiobjective optimization for multiproduct batch plant design under economic and environmental considerations. Ponsich, Azzaro-Pantel, Domenech & Pibouleau, illustrated some guidelines for genetic algorithms implementation in MINLP batch plant design problem [4]. Dietz, Aguilar-Lasserre, Azzaro-Pantel, Pibouleau & Domenech, presented of fuzzy multiobjective algorithm for multiproduct batch plant [5]. In addition, they've used genetic algorithms to solve multiobjective optimization problem with an application to optimal batch plant design in process system engineering. Aguilar-Lasserre, Bautista Bautista, Ponsich & González Huerta, developed an AHP-based decision making tool for the solution of multiproduct batch plant design problem under imprecise demand [6]. Aguilar-Lasserre, Giner, Azzaro-Pantel, Guillermo, Constantino, Pibouleau & Rubén, illustrated the problem of the optimal design of batch plants with imprecise demands using concepts of fuzzy logic [7]. Borisenko, Kegel & Gorlatch, developed and performed a parallel algorithm for finding optimal design for multiproduct batch plants [8].

In the conventional optimal design of a multiproduct batch chemical plant [9], a designer specifies the production requirements for each product and total production time for all products [10]. The number required of volume and size of parallel equipment units in each stage is to be determined in order to minimize the investment cost.

The case of study is a multiproduct batch plant for the production of proteins taken from the literature, we will only consider multiproduct batch plants, which means that all the products follow the same operating steps [11,12], the structure of the variables are the equipment sizes and number of each unit operation that generally take discrete values. Generally, optimization of multiple parameters is an arduous and time consuming task. In this context, we emphasize referring to the work of Montagna, et al. [13], and Asenjo, et al. [14] about the strategy based on monoproduction campaigns was assumed, even when considering the design of multiproduct batch plant. Therefore, machine learning applications are everywhere, from self-driving cars, spam detection, document search, trading strategies, and even speech recognition. This makes machine learning suitable for the era of big data era and data science, especially in pharmaceutical and pharmacological sciences. The main challenge is how to convert data to see what is possible.

The aim of this work is to solve the multiproduct batch plant design problem using (PSA) and (GAs), respectively. The model

presented is general, it takes into account all the available options to increase the efficiency of the batch plant design: unit duplication in-phase and out-phase and intermediate storage tanks.

We have found out that PSA performs effectively and gives a solution, but we would like to solve the problem more effectively, that's why we proposed to apply GAs, an intelligent problem-solving method that has demonstrated its effectiveness in solving combinatorial optimization problem, and satisfactory results are obtained [15].

The paper is organized as follows, section 2 is devoted to the materials and methods including the system description and experimental data, problem statement, model equations and the methodology. While, the results and discussions are handling and reported in section 3. Finally, the conclusions of this work are drawn.

Materials and Methods

System Description and Experimental Data

The case study, taken from the literature, is a multiproduct batch plant for the production of proteins [16]. This example is used as a test bench since it provides models describing the unit operations involved in the process. The batch plant involves eight stages for producing four recombinant proteins, on one hand, two therapeutic proteins, human insulin (A) and vaccine for hepatitis (B) and, on the other hand, a food grade protein, chymosin (C), and a detergent enzyme, cryophilic protease (D). Figure 1 is the flowsheet of the multiproduct batch plant considered in this study.

All the proteins are produced as cells grow in the fermenter. It is hardly necessary to say that the number of intermediate storage tanks is an important constituent of our process: Three tanks have been selected: the first after the fermenter, the second after the first ultrafilter, and the third after the second ultrafilter.

Vaccines and protease are considered to be intracellular. The first microfilter is used to concentrate the cell suspension, which is then sent to the homogenizer for the second microfilter, which is used to remove the cell debris from the solution proteins. The first ultrafiltration step is designed to concentrate the solution in order to minimize the extractor volume. In the liquid-liquid extractor, salt concentration (NaCl) is used as solution in order to minimize the extractor volume. In the liquid-liquid extractor, salt concentration (NaCl) is used to first drive the product to a Poly-Ethylene-Glycol (PEG) phase and again into an aqueous saline solution in the back extraction. The second ultrafiltration is used again to concentrate the solution. The last stage is chromatography, during which selective binding is used to better separate the product of interest from the other proteins.

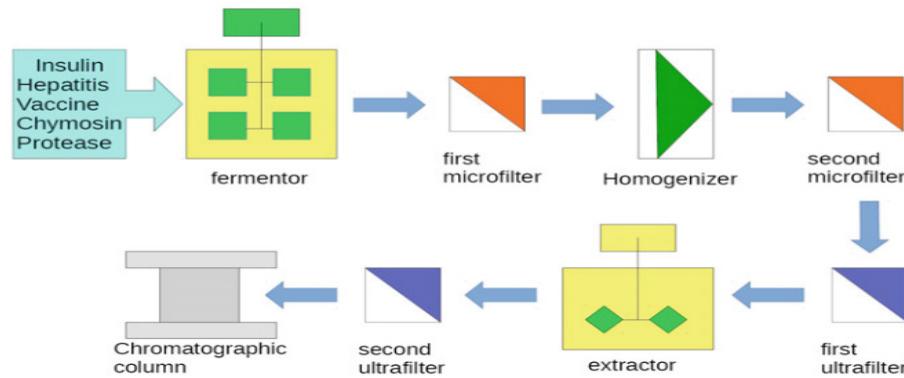


Figure 1: Multiproduct batch plant for protein production.

Insulin and chymosin are extracellular products. Proteins are separated from the cells in the first microfilter, where cells and some of the supernatant liquid stay behind. To reduce the amount of valuable products lost in the retentate, extra water is added to the cell suspension. The homogenizer and the second microfilter for cell debris removal are not used when the product is extracellular. Nevertheless, the first ultrafilter is necessary to concentrate the dilute solution prior to extraction. The final step of extraction, second ultrafiltration, and chromatography are common to both the extracellular and intracellular products. In Table 1 we make an estimation of production targets and product prices [17-19].

Product	Name	Production (kg/year)	Price (dollars/kg)
1	Insulin	1500	8000
2	Vaccine	1000	7500
3	Chymosin	3000	1000
4	Protease	6000	500

Table 1: Product prices and demands.

Problem Statement

The model formulation for DMBP's problem approach adopted in this section is based on Montagna, et al. [16]. It considers not only treatment in batch steps, which usually appear in all types of formulation, but also represents semi continuous units that are part of the whole process (pumps, heat exchangers, etc). A semi-continuous unit is defined as a continuous unit alternating idle times and normal activity periods. Besides, this formulation takes into account mid-term intermediate storage tanks, the obligatory mass balance at the intermediate storage stage, which is one of the most efficient strategies to decouple bottlenecks in batch plant design. They are just used to divide the whole process into subprocesses in order to store an amount of materials corresponding to the difference of each sub-process productivity. In this section we describe the unit models from a conceptual standpoint and also the procedure to derive the data needed for solving the mathematical model. These data are summarized in Tables 2 and 3.

Stage (j)	Unit	S_{ij} (m ³ /kg)			
		Insulin	Vaccine	Chymosin	Protease
1	Fermenter	1.250	0.625	0.415	0.3125
2	Microfilter I	r: 1.25 p: 2.5	r: 0.625 p: no	r: 0.415 p: 0.830	r: 0.3125 p: no
3	Homogenizer	No	0.155	No	0.08
4	Microfilter II	No	r: 0.155 p: 0.31	No	r: 0.08 p: 0.16
5	Ultrafilter I	2.50	0.31	0.830	0.16
6	Extractor	0.40	0.20	0.135	0.10
7	Ultrafilter II	0.40	0.20	0.135	0.10
8	Chromatographer	0.05	0.05	0.05	0.05

Table 2: Size factors S_{ij} (r, retentate; p, permeate).

Stage		T _{ij} (h)			
j	Unit	Insulin	Vaccine	Chymosin	Protease
1	Fermenter	24	24	24	24
2	Microfilter I	12.5 A ⁻¹ B _i	2.5 A ⁻¹ B _i	4.15 A ⁻¹ B _i	1.25 A ⁻¹ B _i
3	Homogenizer	No	0.465 cap ⁻¹ B _i	No	0.24 cap ⁻¹ B _i
4	Microfilter II	No	3.1 A ⁻¹ B _i	No	1.6 A ⁻¹ B _i
5	Ultrafilter I	105A ⁻¹ B _i	5.5 A ⁻¹ B _i	35 A ⁻¹ B _i	3 A ⁻¹ B _i
6	Extractor	1.5	1.5	1.5	1.5
7	Ultrafilter II	18A ⁻¹ B _i	8 A ⁻¹ B _i	4.75 A ⁻¹ B _i	3 A ⁻¹ B _i
8	Chromatographer	0.5	0.5	0.5	0.5

[A: area of microfilters and ultrafilters; B_i: batch size of product i]

Table 3: Time factors T_{ij} [B_i (kg)].

Most of the separation processes information are taken from Asenjo and Patrick [20], the posynomial modeling approach is taken from Salomone and Iribarren [21]. The general batch process literature as reported by Biegler, et al. [22], describes batch stages *j* through a sizing equation and a cycle time that are applied for a product *i* as follows:

$$V_j \geq S_{ij} B_i \quad (1)$$

Where *V_j* is the size of stage *j*, e.g., m³ of the vessel, *B_i* is the batch size for product *i*, e.g., kg of product exiting from the last stage, *S_{ij}* is the size factor of stage *j* product *i*, i.e., the size needed at stage *j* to produce 1kg of final product *i* and *T_{ij}* is the time required to process a batch of product *i* in stage *j* considering the fermentor and the insulin product as an example. If we estimate a final concentration of 50kg dry biomass/m³, that 0.4 of this biomass is proteins and 0.05 of these proteins is insulin, and an overall yield estimate of the process of 0.8 (0.8 of the insulin produced in the fermenter exits the chromatographic column), then the size factor for the fermenter for producing insulin can be estimate as

$$S_{ij} = \frac{m^3}{50 \text{ kg} \times 0.4 \times 0.05 \times 0.8} = 1.25 \text{ m}^3 \quad (2)$$

Similarly, vaccine, chymosine, and cryophilic protease were estimated to be 0.1, 0.15, and 0.2 of total proteins of the biomass, respectively. The batch stage description is completed by estimating a processing time *T_{ij}* for stage *j* when producing product *i*. For the fermenter, we estimate *T_{ij}*=24hrs for all products, which includes time for charging, cell growth, and discharging.

This model of batch stages given by constraint (1a) is the simplest one. Its level of detail suffices for the fermenter and the extractor. These units are truly batch items that hold the load to be processed and whose operations are governed by kinetics, and hence, the operating time does not depend on the batch size.

As a first approximation for the extractor, we take a phase ratio of (1b) for all products. Therefore, the required extractor volume is twice the inlet batch volume, while the inlet and outlet aqueous saline batches are of the same volume. It is also assumed, as a result of preliminary balances, that this operation reduces the total amount of proteins to about twice the amount of the target protein. With respect to the kinetic effects we take as first estimates [23] the following times: 15 min stirring to approach phase equilibrium, 30 min settling to get almost complete disengaging of the phases, and 20 min for charging and discharging. A special consideration must be done in the case of the microfiltration, homogenization, and ultrafiltration stages. Although the mathematical model considers them batch stages, their corresponding equipment consists of holding vessels and semicontinuous units that operate on the material that is recirculated into the holding vessel. The batch items are sized as described before. For example, for the homogenizer processing cryophilic protease, we estimated that the fermentor broth is concentrated 4 times up to 200kg/m³ at microfilter 1 and considered a yield of 1 because

the intracellular protease is fully retained at the microfilter. Then the size factor of the homogenizer vessel is 4 times smaller than the fermenters, i.e., $S_{ij}=0.08\text{m}^3/\text{kg}$ protease. The sizing equation for semicontinuous items can also be found in the general batch processes literature [24]:

$$R_j = D_{ij} \frac{B_i}{\theta_{ij}} \quad (3)$$

Where R_j is the size of the semicontinuous item j , usually a rate of processing. For example, in the case of the homogenizer, it is the capacity in cubic meters of suspension per hour, but in the case of the filters R_j is their area of filtration $A_j(\text{m}^2)$. B_i is again the batch size, θ_{ij} is the operating time that the semicontinuous item j needs to process a batch of product i , and D_{ij} is the duty factor (a size factor for semicontinuous items), i.e., the size needed at stage j to process 1 kg of product i in 1 h. For example, if we adopt three passes through the homogenizer, its duty factor is the vessel size factor $0.08\text{m}^3/\text{kg}$, i.e. $D_{ij}=0.24\text{m}^3/\text{kg}$. The meaning of a capacity of $0.24\text{m}^3/\text{h}$ is that it allows 1kg of final product cryophilic protease to be processed in 1hr.

The general batch processes literature considers semicontinuous units to work in series with batch units so that their operating time are the times for filling or emptying the batch units. However, in the process considered, pumps are the only semicontinuous units, which transfer batches between the units. As the pumps cost does not have a relevant impact on the plant design, they were not explicitly modeled. The times for filling and emptying batch items were estimated and included in the batch cycle times. On the other hand, the process does have special semicontinuous units with an important economic impact on the cost. They are the homogenizer and ultrafilters, but their operating time is the batch processing time of the respective stage. Their mathematical model has been introduced by Salomone and Iribarren, 1994. A size factor for the batch item and a time expression for the stage that depends on both the batch size and the size of the semicontinuous item are as follows:

$$V_j \geq S_{ij} B_i \quad (4a)$$

$$T_{ij} = T_{ij}^0 + T_{ij}^1 \frac{B_i}{R_j} \quad (4b)$$

Where R_j refers to the size of the semicontinuous item that

operates on the batch size at stage j . T_{ij}^0 and T_{ij}^1 are appropriate time factors that take into account contributions to the total cycle time of the stage that are either fixed amounts of time or proportional to the batch size and inversely proportional to the size

of the semicontinuous item. For the homogenizer, R_j is its capacity, T_{ij}^1 the duty factor of the homogenizer itself, and T_{ij}^0 includes the estimated times for filling and emptying the homogenizer holding vessel. In the case of ultrafilters, a fixed permeate flux model was considered with a rate of $20\text{ml}/\text{m}^2$ of membrane area/h. In this case, the size of the semicontinuous item R_j is the filtration area. T_{ij}^0 is again the time for filling and emptying the retentate holding vessel, and T_{ij}^1 is the inverse of the permeate flux times the ratio ($\text{m}^3/\text{permeate}/\text{kg}$).

This ratio is estimated from a mass balance taking into account that the ultrafilters are used for a water removal from solutions up to $50\text{kg}/\text{L}$ of total proteins. Ultrafilters are used to reduce the volume required at the liquid extractor and the chromatographic column. The upper bound on concentration is a constraint that avoids protein precipitation. The microfilter model is quite similar to that of the ultrafilter, but there are two batch items associated to them instead of one, the retentate and the permeate vessels, plus the semicontinuous item area of filtration. For microfilter 1 a fixed permeate flux of $200\text{L}/\text{m}^2\text{h}$ is adopted. For extracellular insulin and chymosin, we estimate a total permeate (feedwater plus make up water) twice the feed, while for intracellular protease and vaccine we estimate it in 75% of the feed (the retentate is concentrated four times). For microfilter 2 a fixed permeate flux model is also used. In this case, the flux is smaller than the one in microfilter 1 because the pore size to retain cell debris is smaller than the one for whole cells. As a first estimation we take $100\text{L}/\text{m}^2\text{h}$ and a total permeate (feed plus make up water) twice the feed. With respect to the chromatographic column, an adsorptive type chromatography is considered, with a binding capacity of $20\text{kg}/\text{m}^3$ of column packing. The size factor of this unit is the inverse of that binding capacity. As a first approximation, a fixed total operating time of 0.5h was estimated for loading, eluting, and washing regeneration.

Finally, the stage model is completed with a cost model that expresses the cost of each unit as a function of its size, in the form of a power law. These expressions are summarized in Table 4, with most of the cost data taken from Petrides, et al. [19].

Unit	Size	Cost
Fermenter	$V_j (\text{m}^3)$	$63400 V_j^{0.6}$
Micro and ultrafilters	$V_{\text{retentate}} (\text{m}^3)$	$5750 V_{\text{retentate}}^{0.6}$
Homogenizer	$V_{\text{holding}} (\text{m}^3)$	$5750 V_{\text{holding}}^{0.6}$
	$\text{Cap} (\text{m}^3/\text{h})$	$12100 \text{ cap}^{0.75}$
Extractor	$V_{\text{extr}} (\text{m}^3)$	$23100 V_{\text{extr}}^{0.65}$
Chromatography	$V_{\text{chrom}} (\text{m}^3)$	$360000 V_{\text{chrom}}^{0.995}$

Table 4: Cost of equipment (U.S.dollars).

Model Equations

The mathematical optimization model for designing the multiproduct batch plant is described in this section. The model

includes the stage models described in the previous section plus additional constraints that are explained in this section. The plant consists of M batch stages (in our case 8 batch stages). Each stage j has a size V_j (m^3), and more than one unit can be installed in parallel. They can work either in-phase (starting operation simultaneously) or out of phase (starting times are distributed equally spaced between them). The duplication in phase is adopted in case the required stage size exceeds the specific upper bound. In this case G_j units are selected, splitting the incoming batch into G_j smaller batches, which are processed simultaneously by the G_j units. After processing, the batches are added again into a unique outgoing batch. Otherwise, duplication out-of-phase is used for time-limiting stages, if a stage has the largest processing time, then it is a bottleneck for the production rate. Assigning M_j units at this stage, working in out of phase mode, reduces the limiting processing time and thus increases the production rate of the train. For this case, the batches coming from the upstream stages are not split. Instead, successive batches produced by the upstream stage are received by different units of stage j , which in turn pass them at equally spaced times onto the downstream batch stage. The allocation and sizing of intermediate storage has been included in the model to get a more efficient plant design. The goal is to increase unit utilization. The insertion of a storage tank decouples the process into two subprocesses: one upstream from the tank, and the other downstream. This allows the adoption of independent batch sizes and limiting cycle times for each subprocess.

Therefore, the previously unique B_i is changed to batch sizes B_{ij} defined for product i in stage j . Appropriate constraints adjust the batch sizes among different units. The objective is to minimize the capital cost of the plant. The decision variables in the model are as follows: At each batch stage the number of parallel units in phase and out of phase and their size, and the installation or absence of intermediate storage between the batch stages and their size. The plant is designed to satisfy a demand of Q_i (kg) of each product i , for the P product considered, within a time horizon H (h).

In summary, the objective function to be optimized is

$$MinCost = \sum_{j=1}^M M_j G_j a_j V_j^{\alpha_j} + \sum_{j=1}^M VT_j^{\eta_j} \quad (5)$$

Where α_j and η_j , C_j and n_j are appropriate cost coefficients that depend on the type of equipment being considered. VT_j is the size of the storage tank allocated after stage j . The size of each unit has to be large enough to be able to process every product:

$$V_j \geq \frac{S_j B_j}{G_j} \quad \forall i = 1, \dots, P; \forall j = 1, \dots, M \quad (6)$$

Where S_j is the size factor for product i in stage j . In case of

parallel units working in phase, the division of B_{ij} by the number of units G_j takes into account the reduction in the batch size to be processed by these units. The operation time T_{ij} to process product i at stage j has the general following form:

$$T_{ij} = T_i^0 + T_i^1 \frac{B_{ij}}{R_j} \quad \forall i = 1, \dots, P; \forall j = 1, \dots, M \quad (7)$$

Where T_i^0 and T_i^1 are appropriate constants that depend on both the product and the stage. Expression 7 accounts for a fixed and variable contribution to the total operating time. The last term in Eq 7 depends on both the batch size and the size of the semicontinuous item associated to this batch stage, as was already discussed previously.

The limiting cycle time for product i in the subprocess h , TL_i^h , is the largest processing time in this production train:

$$TL_i^h \geq \frac{T_{ij}}{M_j} \quad \forall i = 1, \dots, P; \forall j \in J_h; \forall h \quad (8)$$

Where J_h is the set of units which conform the subprocess h the division by the number of units in parallel working out of phase, M_j takes into account the reduction in the cycle time of this stage due to the operation of M_j units that alternatively process the consecutive batches. To avoid accumulation of material, the processing rate of both subprocess downstream and upstream of the storage tank must be the same:

$$\left(\frac{B_i^d}{TL_i^d} \right) = \left(\frac{B_i^u}{TL_i^u} \right) \quad \forall i = 1, 2, \dots, P \quad (9)$$

The constraints 9 equalizes the production rate upstream and downstream of the storage tank. To express 9 in a simple form, the inverse of the production rate of product i (E_i), is defined as

$$E_i = \frac{TL_i^h}{B_{ij}} \quad \forall i = 1, 2, \dots, P; \forall j \in J_h; \forall h \quad (10)$$

Expression 10 is used to replace T_{ij} in constraint 8, dropping constraint 9. The production constraint is posed as follows: during the time horizon H the plant must produce the target production quantities Q_i of each product i . The number of batches of each

product i to be produced during time H is $\frac{Q_i}{E_i}$, and the production of each batch demands a time TL_i . The following constraints holds:

$$\sum_{i=1}^P Q_i E_i \leq H \quad (11)$$

The size of the storage tank VT_j , allocated after batch stage j , is given by the following expression [25]:

$$VT_j \geq ST_j (B_i + B_{i+1}) \forall i = 1, \dots, M; \forall j = 1, \dots, M-1 \quad (12)$$

Where ST_j is the size factor corresponding to the intermediate storage tank, with identical definition to the batch stages. As no a priori tank allocation is given, binary variables y_j are used to select their allocation. The value of variables y_j is 1 if a tank is placed in position j , or zero otherwise. Constraint 12 is generalized to size the tank only if it exists:

$$VT_j \geq ST_j (B_i + B_{i+1}) - F_j (1 - y_j) \forall i = 1, \dots, P; \forall j = 1, \dots, M-1 \quad (13)$$

Where F_j is a constant value sufficiently large such that when y_j is 0 (the tank does not exist), the constraint is trivially satisfied for any value of VT_j .

In particular, the cost minimization will drive $VT_j=0$. When the tank exists ($y_j=1$) the term with F_j vanishes, and the original constraint (12) holds. If the storage tank does not exist between two consecutive stages, then their batch sizes are constrained to be equal. Otherwise, this constraint is relaxed. This effect is imposed by the following constraints [26]:

$$1 + \left(\frac{1}{\Phi} - 1 \right) y_j \leq \frac{B_i}{B_{i+1}} \leq 1 + (\Phi - 1) y_j \quad \forall i = 1, \dots, P; \forall j = 1, \dots, M-1 \quad (14)$$

Where Φ is a constant value corresponding to the maximum ratio allowed between two consecutive batch sizes.

In summary, the multiproduct plant design model that includes the options of parallel units in-phase and/or out of phase and provision of intermediate storage, consists of the objective function 5 subject to constraints 6, 8, 11, 13, and 14, plus the upper and lower bounds that may apply. An important feature of the model is that both the objective function and the constraints are posynomial expressions that possess a unique local (and thus, global) solution [27]. This basic model has been adapted to handle the particular feature of the composite stages (homogenizer, ultrafilters, and microfilters). In this case, constraint 6 is applied not to a general batch stage size but to each of the items that compose it. So in the case of microfilters, constraint 6 applies to both the retentate and the permeate vessels. A new parameter SR_j was introduced to represent the size factor of the retentate vessel, while S_j was left for the permeate vessel. Also in this case, the objective function must account for all the stage components. The notation α_j and β_j were left for the cost coefficients of the permeate vessel, b_j and β_j for the retentate vessel, and d_j and g_j for the filtration area. A similar approach was implemented for the ultrafilters (retentate vessel and ultrafiltration area) and homogenizer (holding vessel and the homogenizer itself).

Methodology

Between 1960s and 1970s witnessed a tremendous development in the size and complexity of industrial organizations. Administrative decision-making has become very complex and involves large numbers of workers, materials and equipment. A decision is a recommendation for the best design or operation in a given system or process engineering, so as to minimize the costs or maximize the gains [28]. Using the term “best” implies that there is a choice or set of alternative strategies of action to make decisions. The term optimal is usually used to denote the maximum or minimum of the objective function and the overall process of maximizing or minimizing is called optimization. The optimization problems are not only in the design of industrial systems and services, but also apply in the manufacturing and operation of these systems once they are designed. Including various methods of optimization, we can mention: MINLP, Particle Swarm Optimization and Genetics Algorithms.

Particle Swarm Algorithms

The PSA is a population-based optimization algorithm, which was inspired by the social behavior of animals such as fish schooling and birds flocking, it can solve a variety of hard optimization problems. It can handle constraints with mixed variables requiring only a few parameters to be tuned, making it attractive from an implementation viewpoint [29]. In PSA, its population is called a swarm and each individual is called a particle. Each particle flies through the problem space to search for optima. Each particle represents a potential solution of solution space; all particles form a swarm. The best position passed through by a flying particle is the optimal solution of this particle and is called *pbest*, and the best position passed through by a swarm is considered as optimal solution of the global and is called *gbest*. Each particle updates itself by *pbest* and *gbest*. A new generation is produced by this updating. The quality of a particle is evaluated by value the adaptability of an optimal function. In PSA, each particle can be regarded as a point of solution space. Assume the number of particles in a group is *M*, and the dimension of variable of a particle is *N*. The *i*th particle at iteration *k* has the following two attributes:

(1) A current position in an *N*-dimensional search space which represents a potential solution: $X_i^k = (x_{i,1}^k, \dots, x_{i,n}^k, \dots, x_{i,N}^k)$, where $x_{i,n}^k \in [l_n, u_n]$ is the *n*th dimensional variable, $1 \leq n \leq N$, l_n and u_n are the lower and upper bounds for the *n*th dimension, respectively.

(2) A current velocity, $V_i^k = (v_{i,1}^k, \dots, v_{i,n}^k, \dots, v_{i,N}^k)$, which controls its fly speed and direction. V_i^k is restricted to a maximum velocity $V_{\max}^k = (v_{\max,1}^k, \dots, v_{\max,n}^k, \dots, v_{\max,N}^k)$. At each iteration, the swarm is updated by the following equations:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (15)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (16)$$

Where P_i is the best previous position of the *i*th particle (also known as *pbest*) and P_g is the global best position among all the particles in the swarm (also known as *gbest*). They are given by the following equations:

$$P_i = \begin{cases} P_i : f(X_i) \geq P_i \\ X_i : f(X_i) < P_i \end{cases} \quad (17)$$

$$P_g \in \{P_0, P_1, \dots, P_M\} \quad f(P_g) = \min(f(P_0), f(P_1), \dots, f(P_M)) \quad (18)$$

Where f is the objective function, *M* is the total number of particles. r_1 and r_2 are the elements generated from two uniform random sequences on the interval $[0,1]$: $r_1 \sim U(0,1)$; $r_2 \sim U(0,1)$ and ω is an inertia weight [30] which is typically chosen in the range of $[0,1]$. A larger inertia weight facilitates global exploration and

a smaller inertia weight tends to facilitate local exploration to fine tune the current search area. Therefore the inertia weight ω is critical for the PSO's convergence behavior. A suitable value for the inertia weight ω usually provides balance between global and local exploration abilities and consequently results in a better optimum solution. Initially the inertia weight was kept constant. However, some literatures indicated that it is better to initially set the inertia to a large value, in order to promote global exploration of the search space, and gradually decrease it to get more refined solutions. C_1 and C_2 are acceleration constants which also control how far a particle will move in a single iteration.

Genetic Algorithms Approach

GA, proposed in this paper based on the work of Wang, et al. [30], are related to the mechanics of natural selection and natural genetics. They combine the survival of the fittest among string structures with a structured yet randomized information exchange to form search algorithms with some of the innovative flair of human search. In every generation, a new set of individuals (strings) is created using bits and pieces of the fittest of the old individuals; while randomized, a GA are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance [30]. According to Wang, et al. [30], the canonical steps of the GA can be described as follows:

(1) The problem to be addressed is defined and captured in an objective function that indicated the fitness of any potential solution.

(2) A population of candidate solutions is initialized subject to certain constraints. Typically, each trial solution is coded as a vector X , termed a chromosome, with elements being described as solutions represented by binary strings. The desired degree of precision would indicate the appropriate length of the binary coding.

(3) Each chromosome $X_{r_i} = 1, 2, \dots, P$, in the population is decoded into a form appropriate for evaluation and is then assigned a fitness score, $\mu(X_r)$ according to the objective.

(4) Selection in genetics algorithms is often accomplished via differential reproduction according to fitness. In a typical approach, each chromosome is assigned a probability of reproduction, $P_{r_i} = 1, 2, \dots, P$, so that its likelihood of being selected is proportional to its fitness relative to the other chromosomes in the population. If the fitness of each chromosome is a strictly positive number to be maximized, this is often accomplished using roulette wheel selection (Goldberg, 1989). Successive trials are conducted in which a chromosome is selected, until all available positions are filled. Those chromosomes with above-average fitness will tend to generate more copies than those with below-average fitness.

(5) According to the assigned probabilities of reproduction, $P_{r_i}=1,2,\dots,P$, a new population of chromosomes is generated by probabilistically selecting strings from the current population. The selected chromosomes generate “offspring” via the use of specific genetic operators, such as crossover and bit mutation. Crossover is applied to two chromosomes (parents) and creates two new chromosomes (offspring) by selecting a random position along the coding and splicing the section that appears before the selected position in the first string with the section that appears after the selected position in the second string and vice versa. Bit mutation simply offers the chance to flip each bit in the coding of a new solution.

Algorithms	Parameters	Value
GA	Population size	200
	Number of generations	1000
	Crossover probability	0.6
	Mutation probability	0.4
	Elitism	1
PSA	Number of particles	200
	Number of generations	1000
	Inertial weight	1.00
	Acceleration constants	2.00

Table 5: The parameters used for running GA and PSA.

According to our experiments, the parameters used for running GA and PSA are showed in Table 5.

Statistical Analysis Methods

The interest in statistical analysis methods has grown recently in the field of computational intelligence. In this section, I will discuss the basic and give a survey of a complete set of variance analysis procedures developed to perform the comparison between PSA and GA, via the use of describing a test of the null hypothesis, which applies to independent random samples from two normal populations of size n_1 and n_2 are taken from normal population having the same variance, it follows F distribution with $n_1 - 1$ and $n_2 - 1$ degrees of freedom, according to this equation: $F = \frac{s_1^2}{s_2^2}$

Results and Discussions

The problem could be formulated as the minimization of the investment cost for equipment and storage tanks. Given that the problem modeled has non linear objective function. For the purpose of optimization problem, the model developed has been solved with PSA and Gas Matlab Toolbox respectively, which is included in the Matlab optimization modeling software, using the data shown in Tables 1, 2, 3, 4. A horizon time of 6000 h has been considered.

Unit ^a	ST _{i,j} size factor for product i in stage j			
	Insulin	Vaccine	Chymosin	Protease
Fermenter	1.25	0.625	0.415	0.3125
Microfilter I	2.50	0.155	0.83	0.08
Homogenizer	2.50	0.155	0.83	0.08
Microfilter II	2.50	0.31	0.83	0.16
Ultrafilter I	0.40	0.20	0.135	0.10
Extractor	0.40	0.20	0.135	0.10
Ultrafilter II	0.05	0.05	0.05	0.05
Chromatography	0	0	0	0

^aTank cost coefficient: $c_j = 5750$; $n_j = 0.6$

Table 6: Intermediate storage cost coefficients and size factors.

Table 7 shows the best, the average and the worst among the final fitness values and the related standard deviation obtained in the 30 runs of PSA and GA, respectively.

Values	PSA (\$)	GA (\$)
Best	912,450	833,647
Average	948,948	850,319.9401
Worst	976,321.5001	865,492.3154
Standard deviation	9.7558	1.5327

Table 7: Comparison of results for 30 runs between PSA and GA.

It is clear from the summary of the results shown in Table 7, that the performance of both PSA and GA produce adequate values regarding the cost for equipment and storage tanks. However, GA performs better than the PSA in terms of the average and the worst fitness values and the standard deviation. Table 7, also, shows the best final solution found in the 30 runs of PSA and GA. According to our knowledge, the case study about the optimal design of protein production plant has been taken from Montagna, et al. [16]. However, they solved the problem using rigorous mathematical programing (MINLP), their model includes 104 binary variables and has been convexified using the transformation proposed by Kocis and Grossman. The MINLP model has been solved using DICOPT++, which is included in the GAMS optimization modeling software. The algorithm implemented in DICOPT++ relies on the Outer Approximation/Equality Relaxation/Augmented Penalty (OA/ER/AP) method. The OA/ER/AP solution method consists of the decomposition of the original MINLP problems into a sequence of two subproblems: a Non Linear Programming (NLP) subproblem and a Mixed Integer Linear Programming (MILP) subproblem also known as the Master problem, which is solved to global optimality (minimize the capital cost \$829,500). However, in previous work of Montagna, et al [16], their model needed a long computational time (more than 86400 seconds) and require several initial values to the optimization variables, they also showed in their paper that the behavior of the demand was completely deterministic.

Whilst, this assumption does not seem to be always a reliable representation of the reality, since in practice the demand of pharmaceutical products resulting from the batch industry is usually variable. Simulations outcomes were then compared with experimental data in order to check the accuracy of the method. The error from the optimal solution is given by:

$$\text{error \%} = 100 \frac{|x_{\text{exp}} - x_{\text{cal}}|}{x_{\text{exp}}} \quad (19)$$

In this research, x_{exp} is considered to be the optimal solution founded by Montagna (Plant cost \$829,500), where the equation 19 is a criterion to confirm the optimal values. Table 8 presents the results obtained in different optimization runs for multiproduct batch plant design. For each simulator run, the average numerical effort spent on solving the problem on LINUX System, Intel® D, CPU2.80 Ghz, 2.99 of RAM. Table 8 shows plant cost, % from optimal solution and CPU time obtaining during 30 runs. PSA and GA performed effectively and give a solution within 10 and 0.5% of the global optimal \$912,450 and \$833,647, respectively. Furthermore, the important feedback could be taken from Table 8, is the GA results in a faster convergence than PSA and the MINLP algorithm. In addition, the GA is so close to the global optimal of MBPD (0.5% from optimal solution) and provides also an interesting solution, in terms of quality as well as of computational time as illustrated in Table 8, while Table 9 presents the sizes for the units involving a set of discrete equipment structure given by PSA. The inconvenience of this configuration is just stopped at 6000h with risk of failing to fulfill the potential future demand coming from a fluctuation of the market.

Technique	Plant cost (\$)	% from optimal solution	CPU time (s)
PSA	912,450	10	800
GA	833,647	0.5	100

Table 8: Optimization runs results for the investment cost founded by PSA and GA during 30 runs.

Stage	1	2	3	4	5	6	7	8
V_j	24.7456	p: 9.922	1.1814	p: 2.3628 4	9.922	0.8921	0.6017	0.0825
		r: 4.961		r: 1.181				
R_j	NA	A: 16.2041	cap: 1.0989	A: 8.668	A: 109.3301	NA	A: 17.8134	NA
VT_j	29.7066	NA	NA	NA	2.2154	NA	0.3795	NA
M_j	3	3	3	3	3	3	3	3
G_j	3	3	3	3	3	3	3	3

Table 9: Equipment structure calculated by PSA.

Stage	1	2	3	4	5	6	7	8
V_j	22.6085	p: 9.0651 r: 4.5325	1.0794	p: 2.1587 r: 1.0794	9.0651	0.8151	0.5497	0.0754
R_j	NA	A: 14.8047	cap: 1.004	A: 7.9194	A: 99.8880	NA	A: 16.2750	NA
VT_j	27.1410	NA	NA	NA	2.0241	NA	0.3467	NA
M_j	1	1	1	1	1	1	1	1
G_j	1	1	1	1	1	1	1	1

Table 10: Equipment structure calculated by GA.

In order to show how the evolution process is going on for both PSA and GAs, respectively, the convergence of the best fitness values is shown in Figure 1. The convergence rate of objective function values as a function of generations for both PSA and GAs is shown in Figure 1, where for clarity only 1000 generations are shown. It is clear from this figure that, for the optimization problem considered, GAs decrease rapidly and converge at a faster rate (around 500 generations) compared to that for PSA (about 800 generations), from which it is clear that GAs seem to perform better compared to PSA. So, for the present problem the performance of the GAs is better than PSA from an evolutionary point of view.

To compare the computational time, the swarm/population size is fixed to 200 for both PSA and GAs algorithms. Whereas, the generation number is varied. Simulation were carried out and conducted on LINUX System, Intel (R) D, CPU 2.80 Ghz, 2.99 of RAM Computer, in the MATLAB 7.0.1 environment. Here the result in the form of graph is shown in Figure 1. It is clear from Figure 1 that the computational time for GAs is very low compared to the PSA optimization algorithm. Further, it can also be observed from Figure 11 that in case of GAs the computational time increases linearly with the number of generations, whereas for PSA the computational time increases almost exponentially with the number of generations. The higher computational time for PSA is due to the communication between the particles after each generation. Hence as the number of generations increases, the computational time increases almost exponentially.

Table 9 presents the sizes for the units involving a set of discrete equipment structure given by PSA. The inconvenience of this configuration is just stopped at 6000 hours with risk of failing to fulfill the potential future demand coming from a fluctuation changing of the market.

On the other hand, the calculation of the structure of equipment using GA is illustrated in Table 10. The total production time, also, computed by GA is 5491.12 hours to fulfill the eventual increase of future demand caused by market fluctuation. In addition, the GA results in a faster convergence. However, the equipment structure showed by PSA is very expensive. Furthermore, the PSA approach has the disadvantage of long CPU time.

At the same time as, the GA allow the reduction of the idle time to the stage, in any way, Table 11 and Table 12 show the idle times obtained by PSA and GA respectively.

Unit	1	2	3	4	5	6	7	8
Product	1	2	3	4	5	6	7	8
Insulin	0	0	NA	NA	0	57.7	NA	67.11
Vaccine	0	54	0	0	60.79	57.7	22.9	67.11
Chymosin	0	17	NA	NA	17.54	57.7	27.9	67.11
Protease	0	63	16	15	63.07	57.7	55.03	67.11

Table 11: Idle times in plant calculated by PSA (seconds).

Unit	1	2	3	4	5	6	7	8
Product	1	2	3	4	5	6	7	8
Insulin	0	0	NA	NA	0	0.01	0	0
Vaccine	0	1.93	0.04	0	2.91	0	0.17	0
Chymosin	0	0.01	NA	NA	0	0	0.31	0.17
Protease	0	2.09	0	0	3.07	0	0.5	0

Table 12: Idle times in plant calculated by GA (seconds).

However, some observations about some important aspects in our implication of GAs and some problems in practice: the most important of all is the method of coding, because the codification is very important issue when a genetic algorithm is designed to dealing with combinatorial problem, also of the characteristics and inner structure of the DMBP.

The commonly adopter concatenated, multi-paramer, mapped, fixed point coding are not effective in searching for the global optimum. According to the inner structure of the design problem of multiproduct batch that gives us some clues for designing the above mixed continuous discrete coding method with a four-point crossover operator. As is evident from the results of application, this coding method is well fit for the proposed problem. Another aspect that affects the effectiveness of our Genetic Algorithms procedure considerably is crossover.

Corresponding to the proposed coding method, we adopted a four-point crossover. It is commonly believed that multipoint crossover is more effective than the traditional one-point crossover method. It is also important to note that the selection of crossover points as well as the way to carry out the crossover should take in account the bit string structure, as is the case in our codification.

One problem in practice is the premature loss of diversity in the population, which results in premature convergence, because premature convergence is so often the case in the implementation of GA according to our calculation experience. Our experience makes it clear that the Elitism parameter can solve the premature problem effectively and conveniently. However, a numerical calculation of the model under machine learning approach is examined in table 13.

Efficiency results	Symbol	Machine Learning Algorithms 1	Machine Learning Algorithms 2
Production rate	P	70000	55000
Annual demand rate	D	50000	30000
Setup cost	K	100	70
Holding cost	H	5	4
Variable cost	C	25	20
Selling price	S	50	40
Inspection rate	X	100000	60000
Return cost	R	15	10
Penalty cost	Pi	7	5
Screening cost	I	0.5	0.03

Table 13: Comparison of the machine learning algorithms.

In order to further explain the effects of these algorithms on solving the MBPD problem, the variance analysis was performed. Each of the PSA and GA algorithms was run 30 times. The Minitab software was used to analyze the results. Therefore, the results are given in table 14 and 15.

Algorithm	N	Avg	SD	Standard error		95% confidence interval of mean		Min	Max
						Min	Max		
PSA	30	1859.0000	8.48935	2.68743		1833.9205	1845.0795	1828	1857
GA	30	1838.0000	5.49936	2.08701		1828.2733	1837.7201	1828	1845

Table 14: The results of two algorithms solving MBPD problem.

	Quadratic sum	Free degree	Mean Square	F	Significance
SDB	2339.676	3	779.895	15.455	0.000
SDI	1814.100	36	50.392	-	-
SUM	4154.775	39	-	-	-

Table 15: Variance analysis result of MBPD problem.

Table 14 indicates that, the mean square deviation between groups (SDB) is 779.895. The mean square deviation within groups (SDI) is 50.392. The test statistic $F = 15.477$. If significance level $\alpha = 0.05$, then the critical value

$2.92 \leq F_a(3.36) \leq 2.84$. Thus, $F > F_a(3.36)$ indicating that the difference between the average figures is significant, that is, the performance difference of algorithms is significant.

Nevertheless, these techniques are not a panacea, despite their apparent robustness, there are control “parameters” involved in these metaheuristics and appropriate setting of these parameters is a key point for success.

Conclusions

Techniques such as PSA and GA are inspired by nature, and have proved themselves to be effective solutions to optimization problems. We applied Genetic Algorithms with an effective mixed continues discrete coding method with a four crossover point to solve the problem of DMBP. GA perform effectively and give a solution within 0.5% of the global optimum. Whilst, it is observed that, in terms of computational time, the GAs approach is faster. The computational time increases linearly with the number of generations for GA, whereas for PSA the computational time increases almost exponentially with the number of generations, interpreting that, the higher computational time for PSA is due to the communication between the particles after each generation. Furthermore, the results provided by GA are much better with respect to PSA. In this paper, the GA gave us the highest efficiency and justifies its use for solving nonlinear mathematical models. Therefore, this work provides an interesting decision/making approach to improve the design of multiproduct batch plants under conflicting goals.

Acknowledgements

The author express their gratitude, appreciation and acknowledgements to the financial support provided by PRODEP (Programa para el Desarrollo Profesional Docente para el tipo Superior) under the Research Project 511-6/18-8724.

References

1. Corsano G, Montagna JM, Iribarren OA, Aguirre PA (2009) Mathematical Modeling Approaches for Optimization of Chemical Processes. Nova Science Publishers, Inc: New York, USA.
2. Dietz A, Azzaro-Pantel C, Pibouleau L, Domenech S (2006) Multiobjective optimization for multiproduct batch plant design under economic and environmental considerations. *Comput Chem Eng* 30: 599-613.
3. Dietz A, Azzaro-Pantel C, Pibouleau L, Domenech S (2008) Strategies for multiobjective genetic algorithm development: application to optimal batch plant design in process systems engineering. *Comput Ind Eng* 54: 539-569.
4. Ponsich A, Azzaro-Pantel C, Domenech S, Pibouleau L (2007) Mixed-integer nonlinear programming optimization strategies for batch plant design problems. *Ind Eng Chem Res* 46: 854-863.
5. Dietz A, Aguilar-Lasserre A, Azzaro-Pantel C, Pibouleau L, Domenech S (2008) A fuzzy multiobjective algorithm for multiproduct batch plant: application to protein production. *Comput Chem Eng* 32: 292-306.
6. Aguilar-Lasserre AA, Bautista Bautista MA, Ponsich A, González Huerta MA (2009) An AHP-based decision-making tool for the solution of multiproduct batch plant design problema under imprecise demand. *Comput Oper Res* 36: 711-736.
7. Aguilar-Lasserre A, Giner AH, Azzaro-Pantel C, Guillermo CR, Constantino MS, et al. (2009) Multiobjective multiproduct batch plant design under uncertainty: application to protein production. In: 10th International Symposium on Process Systems Engineering - PSE2009. Salvador, Brazil. pp: 15-20.

8. Borisenko A, Kegel P, Gorlatch S (2011) Optimal design of multi-product batch plants using a parallel branch-and-bound method. In: Malyshkin V (Edn.) *PaCT* 2011. LNCS, 6873, pp. 417-430.
9. Takamatsu T, Hashimoto I, Hasebe (1998) Optimal scheduling and minimum storage tank capacities in a process system with parallel batch units. *Comput Chem Eng* 3: 185-195.
10. Floudas A (1994) Global optimization in the 21st century: advances and challenges. *Comput Chem Eng* 29: 1185.
11. Cao DM, Yuan XG (2002) Optimal design of batch plants with uncertain demands considering switch over of operating modes of parallel units. *Ind Eng Chem Res* 41: 4616-4625.
12. Crougham M, Caldwell V, Randlev B, Billeci K, Nieder M (1997) Prediction of culture performance through cell cycle analysis: potential tool in operations scheduling. In: *Proceedings of the Biochemical Engineering*.
13. Montagna JM, Iribarren OA, Galiano FC (1994) The design of multiproduct batch plans with process performance models. *Trans. IChemE*, 72: 783.
14. Azzaro-Pantel C, Dietz A, Pibouleau L, Domenech S (2005) A framework for multiproduct batch plant design with environmental consideration: application to protein production. *Ind Eng Chem Res* 44: 2191-2206.
15. Dessouky Y, Bayer A (2002) A simulation and design of experiment modeling approach to analyze facility maintenance costs. *Comput Ind Eng* 43: 423-436.
16. Montagna JM, Vecchietti AR, Iribarren OA, Pinto JM, Asenjo JA (2000) Optimal design of protein production plants with time and size factor process models. *Biotechnol Program* 16: 228-237.
17. Birattari M, Zlochin M, Dorigo M (2006) Towards a theory of practice in metaheuristics design: a machine learning perspective. *RAIRO – Theor Inf Appl* 40: 353-369.
18. Datar R, Rosen CG (1990) *Downstream Process Economics in Separation Processes in Biotechnology*. New York Press Inc, New York.
19. Petrides D, Sapidou E, Calandranis J (1995) Computer aided process analysis and economic evaluation. *Biotechnol. Bioeng* 48: 529-541.
20. Asenjo JA, Patrick I (1990) *Large Scale Protein Purification in Protein Purification Applications: A Practical Approach*. Oxford Press Inc, Oxford.
21. Salomone HE, Iribarren OA (1992) Posynomial modeling of batch plants. *Comput Chem En* 16: 173-184.
22. Biegler LT, Grossmann IE, Westerberg AW (1997) *Systematic Methods of Chemical Process Design*. Prentice Hall, New Jersey, USA.
23. Mistry SL, Kaul A, Merchuk JC, Asenjo JA (1996) Mathematical modeling and computer simulation of aqueous two phase continuous protein extraction. *J Chromatogr A* 741: 151-163.
24. Reklaitis GV, Ravindran A, Ragsdell KM (1983) *Engineering Optimization Methods and Applications*. Wiley, New York, USA.
25. Modi AK, Karimi IA (1989) Design of multiproduct batch processes with finite intermediate storage. *Comput Chem Eng* 13: 127-139.
26. Ravemark D (1995) Optimization models for design and operation of chemical batch processes. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich.
27. Salvendy G (1982) *Industrial Engineering Handbook*. Wiley, New York.
28. Grossmann IE, Sargent RW (1979) Optimal design of multipurpose chemical plants. *Ind Eng Chem Process Des* 18: 343-348.
29. Shi Y, Eberhart RA (1998) Modified particle swarm optimizer, In: *IEEE Proceeding*, pp. 69-73.
30. Wang C, Quan H, Xu X (1996) Optimal design of multiproduct batch chemical process using genetic algorithm. *Ind Eng Chem Res* 35: 3560-3566.