PREDICTIVE SCHEDULING OF A PENICILLIN BIOPROCESS PLANT

S. Lau, M.J. Willis, G.A. Montague and J. Glassey

School of Chemical Engineering & Advanced Materials,
University of Newcastle, Newcastle-upon-Tyne, NE1 7RU, UK

Abstract: In this paper a predictive strategy for the reactive scheduling of a multi-stage bioprocessing plant is outlined. In the procedure, the various batch stages of the bioprocess are dynamically re-allocated to the appropriate processing units in response to the biological variability inherent in each stage. Forecasts of the process productivity and consequent completion times for the tertiary stages of industrial penicillin fermentations are used in conjunction with a genetic algorithm to solve the scheduling problem. Initial results using data from a commercial penicillin plant demonstrate that the predictive scheduling framework could deliver increased production and, consequently, major financial benefits. *Copyright © 2002 IFAC*

Keywords: genetic algorithms, fermentation processes, optimisation problems, scheduling algorithms, forecast

1. INTRODUCTION

The industrial scheduling problem involves assigning resources to tasks over a (fixed) period of time to fulfil production goals. Scheduling problems are often complicated by a large number of constraints. These constraints may be resource, capacity or production related. A scheduling system makes decisions dynamically to accommodate activities within the framework of available resources to ensure that tasks are completed either at a given time or at a minimum cost. Problems with scheduling occur over a wide range of industries. Examples of these are: filling and emptying of tanks in a ballast water treatment facility at a port (Dahal *et al.* 2001), clothing production (Dessouky *et al.* 1998), scheduling and rearrangements of tasks on field-programmable gate arrays (FPGA) (Middendorf *et al.* 2002), scheduling of multi-product plants (Sand *et al.* 2000) and laboratory management and scheduling of experiments to be performed on chemical workstations (Aarts *et al.* 1995). There is a vast amount of literature providing different solution approaches. Much of the literature is based on non-chemical processing applications e.g. machine shops, discrete assembly manufacture and computer system operation, but due to similarities in the problem structure, some of the literature is applicable to scheduling in the process industries (Reklaitis 1982 and Shah 1999).

Scheduling techniques can be observed from at the most basic level, manual approaches to the more sophisticated artificial intelligence strategies. An overview of these techniques is given by Morton and Pentico (1993). However a common problem that arises with scheduling is the continual need to alter previous schedules due to process problems and production changes that occur. Investigation into reactive scheduling and predictive scheduling in the manufacturing environment is discussed by Sabuncuoglu and Bayiz (2000) and Kizilisik (1999) with a special emphasis on single machine analysis. Fang *et al.* (1993) describes rescheduling in a job shop environment with the application of a genetic algorithm (GA).

The process industry covers a wide range of fields such as chemicals, pharmaceuticals, foods, paints and many others. This paper concentrates on the scheduling of operations on a batch fermentation pharmaceutical plant. There is great interest in the batch processing area as discussed by Orçun *et al.* (2001). It is known that batch processes exhibit a certain degree of variability which is caused by, for instance, changes in operator response time; fluctuations in utility availability; minor equipment malfunctions; recipe inaccuracies and changes in raw material quality (Cott and Macchietto 1989). Fermentation batches are amongst the most challenging of batch operations to schedule effectively due to the high level of inherent biological variability.

This study investigates the integration of a scheduling tool with a forecaster of production levels. At present, there is considerable variation in product concentrations (titres) with respect to processing time in the tertiary fermentation stage (i.e. the main production stage). In particular, some batches take less time to reach the production optimum, while some are left too long and the concentration of the product starts to decline. The current process scheduling method does not take this into account and only uses nominal batch processing times derived from historical plant data. However, to optimise the production within the plant, it is necessary to predict the best time to harvest the product with respect to overall plant output and to re-schedule efficiently so to maximise production.

2. PROCESS DESCRIPTION

The plant shown in Figure 1 is used to produce the antibiotic penicillin. Each batch goes through three main stages of the fermentation process. The first two stages of the fermentation involve the growth of the organism (*Penicillium chrysogenum*). This is to allow time for it to increase its biomass and to adapt to change in volume/size and its environment (changing from growing on solid to liquid medium). The tertiary (final) fermentation stage is the production stage of the product penicillin. The choice of harvesting time for this stage is of great importance because, due to the unstable nature of penicillin, its concentration reduces after it has reached its peak in production. This has important implications for the use of any automated scheduling system. An effective scheduling mechanism must be able to take into account the variability associated with the optimum harvest time, as well as other operational factors such as contamination, maintenance, breakdowns and the like.

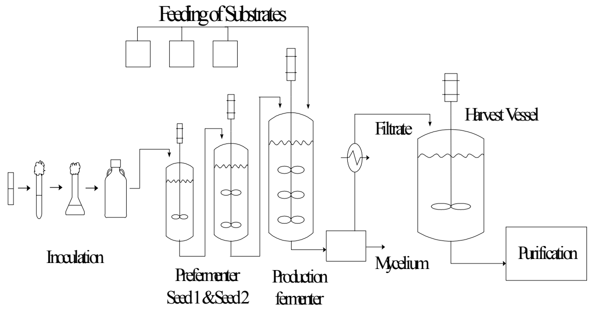

Fig 1. A simplified diagrammatic layout of a penicillin process plant.

2.1. Problem formulation

As the aim of this work is to schedule the various stages of batch operation on a bioprocessing plant, it is necessary to be able to compute the total completion time of all penicillin batches given the allocation of resources dictated by the current schedule. In addition to the nominal completion times for each of the individual stages, factors such as unit clean out times, unit setup times as well as the processing history of the plant determine the total time required to complete all batches. In the case of the predictive strategy, the predicted completion time of the each tertiary processing stage is used in place of the nominal completion time.

In this paper, a simple recursive algorithm, referred to as a recurrence relationship, with a generic structure is used as the basis for the scheduling problem solution. The generic structure means that it can be easily configured for a wide range of scheduling problems of industrial importance. Ku *et al.* (1987) developed similar relationships for jobshop problems, while Kim *et al.* (1996) suggested an algorithm for flowshop problems; however, it took the form of several *if-then* rules and therefore is not as easily configurable. In contrast, the calculation of the total completion time in this paper uses a more convenient matrix/vector formulation.

To determine the batch completion time the batch sequence is characterised by a permutation of integers 1,2,3,..,*n*. Let f_{nI} denote the time at which the n^{th} batch in the sequence leaves stage *I*. To calculate f_{nI} two conditions must be fulfilled:

1. The n^{th} batch in the sequence cannot leave a stage *I* unit until all the 'processing' is complete and to be on a stage *I* unit it must have left stage *I-1*
2. The n^{th} batch in the sequence can only start on stage *I* after one of the previously scheduled batches has finished, the unit has been cleaned, and setup is complete and cannot leave stage *I* until it has been processed.

Consider the following definitions:

n = total number of batches
M = total number of processing stages
P_I = total number of units available for processing a batch at stage *I*
$C_I(n)$ = a (1 x P_I+n-1) vector of stage *I* cleaning times relating the current batch to the previous 1,..,n-1 batches in the sequence, i.e.

$$C_I(n)=[c_I(1),c_I(2),..c_I(n-1)]$$

$f_I(n)$ = the stage *I* completion time of the n^{th} batch in the sequence
$F_I(n)$ = a (1 x P_I+n-1) vector of stage *I* completion times for the previous 1,..,n-1 batches in the sequence.
$r_I(n)$ = are (P_I x 1) vectors used to define the unit allocation of batch *n* in the sequence at stage *I* of the process. (if the *j*th element of element $r_I(n)$ = 1 then batch *n* is allocated to unit *j* at stage *I* and the other elements of $r_I(n)$ is equal to zero)
$t_I(n)$ = a (1 x P_I) vector defining the stage *I* processing time for all available units.

To calculate the completion time of batches scheduled on a *M* stage plant with P_I units at stage *I* the following recurrence relationship is used:

$$f_I(n)=t_I(n)r_I(n)+\max(f_{I-1}(n),r_I^T(n)R_I(n)[F_I^T(n)+C_I^T(n)]) \qquad (1)$$

with:

$$R_I(n)=[R_I(n-1)-r_I(n-1)\{r_I^T(n-1)R_I(n-1)\},r_I(n-1)] \qquad (2)$$

The first part of the relation $f_I(n) = t_I(n)r_I + f_{I-1}(n)$ accounts for condition (1), while the second term $f_I(n) = t_I(n)r_I(n)+r_I^T(n)R_I(n)[F_I^T(n)+C_I^T(n)])$ accounts for condition (2). Operation $r_I^T(n)R_I(n)F_I^T(n)$ is used to calculate the completion time of the previous batch scheduled on the same stage *I* unit as the n^{th} batch in the sequence, while $r_I^T(n)R_I(n)C_I^T(n)$ determines the necessary cleaning time and setup time between two batches. Thus, $R_I(n)$ is a matrix used to indicate which of the previous *n-1* batches were last allocated to a particular unit. Note that the $R_I(n)$ is an augmented matrix comprising a modified $R_I(n-1)$ and the unit allocation vector, $r_I(n-1)$, which indicates the unit used by the previous batch. As $R_I(n)$ is used to indicate the last batch allocated to each unit, by definition $r_I(n-1)$ will be the new indicator for unit '*j*', hence the modification,

$$R_I(n-1)-r_I(n-1)\{r_I^T(n-1)R_I(n-1)\}$$

is used to remove the previous indicator. The algorithm is initialised with $R_I(0) = I\ (P_I \times P_I)$ which will allow the calculation of the completion times for the batches already running on the plant.

2.2. Plant Information

Current plant operation uses fixed (nominal) processing times for each stage. As discussed earlier this is not ideal, as it limits the overall performance of the plant. Consideration of a number of batches of process data of actual penicillin titre (Figure 2) underpins the need to move from fixed batch times. This clearly demonstrates how each batch varies in the rate of production. For instance, with respect to batch four the peak concentration is reached at around 200hrs while for batch two the concentration of penicillin is still rising. For the purposes of this paper, such existing information is used to establish the benefits that may be obtained by stopping at the appropriate production levels rather than running for a fixed period of time.

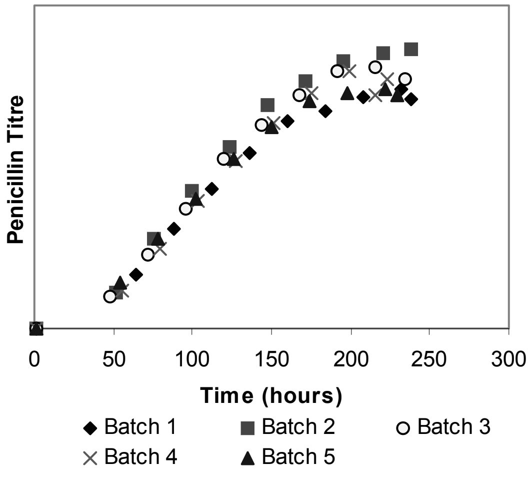

Fig. 2. Penicillin titre of five batches

A fundamental problem in the scheduling of fermentation batches is that variability leads to deviations in ideal completion time. However, it is necessary to know in advance the completion time as the previous stage must be inoculated well prior to production stage completion in order to be in an optimal state for transfer when the new production batch requires it. Thus a predictor of future penicillin titre is essential for scheduling purposes.

A further complication arises from the fact that penicillin titre is only available infrequently by off-line analysis. As a consequence the predictor is required to overcome measurement delay in addition to the horizon for scheduling requirements. One possible solution is to use estimators to determine the current penicillin titre and then use a forecaster for prediction from that point. The application of neural networks as an estimator for the penicillin titre is discussed by Yuan and Vanrolleghem (1998) and Lopes and Menezes (1998). Also, neural network models have been applied in biomass estimation and fault diagnosis for penicillin production as shown by Montague and Morris (1994). Currently estimators of current penicillin titre have not been considered, with the forecaster acting on off-line assays.

The penicillin titre trends have a distinctive shape that closely matches that of the logistic function (equation 3). The forecaster operation involves fitting the coefficients of the logistic function to penicillin titre measurements for the batch in question up to the current time and using the model to forecast future production. Forecasts are not possible in the early stage of the batch as few titre measurements are available but for scheduling purposes forecasts after around 150hrs of operation are sufficient as it is the latter period of the batch that is important.

$$y = \frac{C}{1+Ae^{-Bx}} \qquad (3)$$

The forecaster must be capable of prediction over a forty hour horizon in order to allow the necessary time for the seed batch to be completed. Figure 3 shows the forecaster applied to an example batch. It can be seen that the required forecasting capability is achieved.

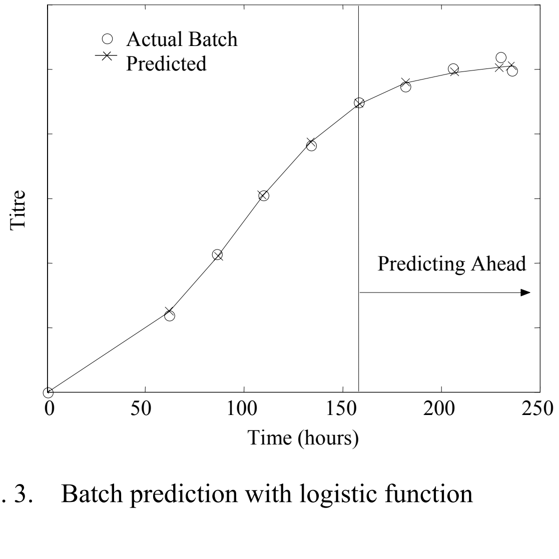

Fig. 3. Batch prediction with logistic function

The decision as to the optimum time to harvest is influenced by the forecast of the rate of penicillin production. When the rate of penicillin production falls below average rate achieved for a new batch (including the batch turn-around time) then the current batch should be terminated. However the scheduler requires the use of the forecaster to determine when this condition is likely to occur and therefore initiate a secondary stage fermentation. When the forecast is made, the best assumption is that a new batch will perform in an average manner. As the secondary stage fermentation progresses assessments of its quality may be possible and if different from average this would impact on termination time of the current tertiary stage vessel. Given the limited supply of information from the secondary stage fermentations, such modifications are currently not considered appropriate.

2.3. Solution of the scheduling problem

As noted earlier, there are many techniques available for the solution of scheduling problems. One method is the use of a genetic algorithm (GA) as this kind of algorithm has been applied successfully to many combinatorial optimisation problems (Azzaro-Pantel *et al.* 1998). The application of GA to scheduling problems has been described in the job-shop environment (Rubin and Ragatz 1995, Lee and Choi 1995 and Della Croce *et al.* 1995). Cartwright and Tuson (1994) implemented a GA to handle an industrial flowshop by optimising both chemical feed order and topology. In their study the GA provided a reliable method for finding near optimum feed order/topology combinations.

2.4. Genetic algorithms (GAs)

Genetic algorithms (GAs) are a type of heuristic optimisation method that are based on the mechanics of genetics (Holland 1975). GAs solve problems by using a process analogous to natural selection to evolve candidate solutions which are typically encoded as a population of abstract mathematical chromosomes e.g. binary, integer, or real valued string sequences. For further details see Goldberg (1989). In the following section, the type of GA that is used to schedule the batch processes is discussed.

Encoding Scheme*:* The order based GA is used, two separate strings are applied to represent batch order and unit allocation. A permutation representation is used for the batch order string, while the actual unit number is used for the unit string. Thus, the following strings are used e.g. to represent six batches and 4 units:

Batch order string	3	4	1	5	6	2
Unit string	2	4	1	3	2	1

This shows that batch 3 is processed on unit 2 first, batch 4 on unit 4 etc.

GA operators *Crossover:* The purpose of this operator is to combine information from relatively successful strings in order to produce better offspring. The classic one and two point crossover operators cannot be applied, as infeasible solutions would be generated. This appears often in strings with permutation. To avoid this a number of crossover schemes have been developed. Goldberg (1989) discusses partially matched crossover (PMX), cycle crossover (CX) and order crossover (OX) and Syswerda (1989) describes uniform crossover. In this study the PMX operator is used. Here, two cutting sites are chosen randomly (point 2 and point 5) PMX defines a matching section that is used to cross through position-by-position exchange operations. This is demonstrated below;

Parent 1	Batch1 string	3	4	1	5	6	2
	Unit1 string	2	4	1	3	2	1
Parent 2	Batch2 string	1	5	2	3	4	6
	Unit2 string	3	1	1	2	4	2

The crossover takes place with a full set of permutations within the batch order string. The unit string is then realigned with the batch order string. The offspring obtained are:

Child 1	Batch1 string	1	6	2	3	4	5
	Unit1 string	1	2	1	2	4	3
Child 2	Batch2 string	2	3	1	5	6	4
	Unit2 string	1	2	1	3	2	4

Mutation: The algorithm uses three types of mutation operator. The first is inversion where both batch order and unit string are simultaneously cut at two points and the information is reversed. The second is batch order mutation where only the batch order string is mutated and the third is unit string mutation where the unit string is mutated. The probability that a particular mutation operator is chosen is equal.

2.5. The Plant Scheduler

Figure 4 illustrates the functionality of the predictive/reactive scheduler. Given a set of process batches and unit resources, the GA optimises the schedule to produce an optimal sequence as well as determining the unit allocation for each of the batches.

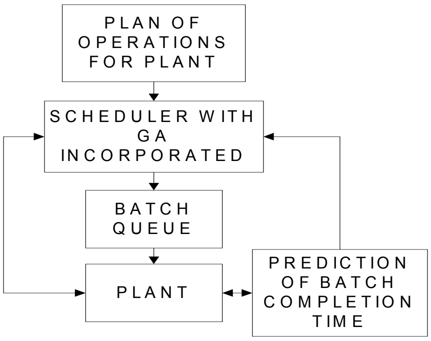
PLAN OF OPERATIONS FOR PLANT
SCHEDULER WITH GA INCORPORATED
BATCH QUEUE
PLANT
PREDICTION OF BATCH COMPLETION TIME

Fig. 4. The plant scheduling system

The first batches are allocated to plant resources and are processed. The remaining jobs are put into the queue (in the order defined by the schedule) and will be processed on the plant when a processing resource becomes available. Thus, the processing time of the batches through each of the stages is monitored and their status is fed back into the scheduler. If the expected processing time of a batch changes from the unit processing time assumed initially there then presents an opportunity to re-schedule. Consequently, the GA then re-optimises the batches held in the queue to generate a new schedule that may be implemented on the plant.

2.6. Case Study

In this work, the concept of applying the knowledge of batch production status together with a forecaster for future production levels in a predictive scheduler is performed using past batch data records from the industrial plant. The past batch data acts as a 'simulation' of behaviour and can be used since in nearly all cases the actual batches were longer than appropriate. To demonstrate the overall effect of this application, 100 batches were scheduled. The batches were to be optimally sequenced through 3 stages, the first and second stage having 4 units and the final tertiary stage 18 units. The predictive scheduler is to be compared with two other schedulers; on plant scheduler and fixed GA scheduler, both of which uses fixed batch processing time. This in order will show the benefits that may be obtained by adopting a more reactive scheduling methodology.

An initial number of batches are placed onto the plant simulation. A queue is formed, as further batches are ready to be scheduled onto the plant simulation. As batch status of the main fermentation is being measured the predicted information of completion time is fed back to the scheduler. 50 runs of the GA based scheduler are used for the 100 batches. This is because different batch runs may have different completion times (as the solution may represent a local rather than global minimum) and therefore a number of runs will allow a fair comparison to be made rather than producing a one-off result.

For all runs, the GA is configured with the following parameters: crossover probability 0.8, mutation probability 0.1, population size 500, and a steady-state reproduction having a population retention of fifty individuals at each generation and each run of the GA is configured for 100 generations.

3. RESULTS

The results obtained after 50 runs are displayed by the box and whisker plots in Figure 5. The line that runs across the figure represents the time taken on plant to complete 100 batches, using the existing scheduling policy. These plots highlight the most salient features of the results obtained from each set of 50 runs. The centreline of the box is the median value of the data, whilst the box itself represents the inter quartile range of the data. From the top and bottom of each box a vertical whisker extends to the extreme values of the data. Furthermore, the notches around the median lines are constructed such that, if there is no overlap between the notches, the medians are significantly different at the 95% confidence level.

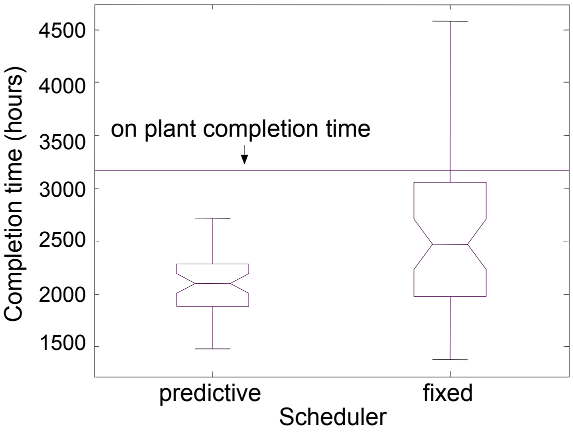

Fig. 5. Comparison of completion times between the predictive and fixed scheduler.

It can be observed from the plots produced in Figure 5 that the predictive scheduler shows a reduction in completion time for the 100 batches in comparison to the fixed scheduler. The distribution of completion times is tighter as the scheduler plans operation in a manner that takes account of DSP availability. The fixed scheduler refers to fixed operational batch length, which is sought but not achieved. Here, batch length variation is a result of schedule limitations and the inability to send the batch to DSP.

Table 1 summarises the mean completion times taken from the three schedulers (on plant, fixed GA and predictive GA scheduler).

Table 1. Mean completion times of processing 100 batches using 3 different schedulers

	On plant	Fixed	Predictive
Mean Completion Times (hrs)	3168	2536	2094

Using the predictive scheduler a saving of approximately 44 days is gained when compared to the on plant scheduler and approximately 18 days when compared to the fixed scheduler over a five month period. The days gained could allow additional batches to be processed. More importantly, as the batches were processed at their near optimum production level, clearly an improvement in the total amount of penicillin produced by the overall process is achieved.

4. CONCLUSIONS

In this paper, predictive scheduling of a bioprocess plant has been considered. The reasons behind the need for predictive scheduling have been discussed and the particular problems encountered as a result of the biological nature of the process considered. It has been demonstrated that the use of such a predictive scheduling strategy will be beneficial. As a result the plant efficiency and productivity could potentially be significantly improved.

A key aspect of the scheduler is the forecasts of future productivity. Whilst in many batch chemical operations failure to transfer at the 'correct' time can result in reduced plant occupancy, in biological systems failure to transfer can additionally mean major productivity losses due to irreparable biological consequences. Whilst this paper attempts to address the problem of forecaster reliability, several aspects still remain to be considered. For instance a research challenge will be how to incorporate the increasing uncertainty associated with the extending forecasting horizon. Application of other models for predicting penicillin could be compared for forecasting accuracy. Subsequent work will consider how the development of an on-line forecaster to allow advance prediction with associated uncertainty will impact on the scheduler. Furthermore, work has recently identified significant variations in inoculum quality which would impact on future production performance and thus optimal completion times. Assessment of secondary stage fermentation information to update completion is an obvious improvement.

This paper has begun to address the issues associated with biological process scheduling on a complex multi-unit fermentation plant but many further advances are still required before an effective predictive scheduler suitable for the industrial environment is achieved. The scale of operation and the major financial opportunities that schedule improvement would provide a strong motivation for tackling the problems.

5. REFERENCES

Aarts, R. J., J. S. Lindsey, *et al*. (1995). Flexible Protocols Improve Parallel Experimentation Throughput. *Clinical Chemistry* **41**(7): 1004-1010.

Azzaro-Pantel, C., L. Bernal-Haro, *et al*. (1998). A Two-Stage Methodology for Short-Term Batch Plant Scheduling: Discrete-Event Simulation and Genetic Algorithm. *Computers and Chemical Engineering* **22**(10): 1461-1481.

Cartwright, H. M. and A. L. Tuson (1994). Genetic Algorithms and Flowshop Scheduling: Towards the development of a real-time process control. *AISB workshop on Evolutionary Computing.*

Cott, B. J. and S. Macchietto (1989). Minimizing the Effects of Batch Process Variablity using Online Schedule Modification. *Computers & Chemical Engineering* **13**(1/2): 105-113.

Dahal, K. P., G. M. Burt, *et al*. (2001). A Case Study of Scheduling Storage Tanks Using a Hybrid Genetic Algorithm. *IEEE Trans. Evolutionary Computation* **5**(3): 283-294.

Della Croce, F., R. Tadei, *et al.* (1995). A Genetic Algorithm for the job shop problem. *Computers & Operations Research* **22**(1): 15-24.

Dessouky, M. I., R. L. Marcellus, *et al.* (1998). Scheduling Identical Jobs on Uniform Parallel Machines with Random Processing Times. *Computers Ind Engineering* **35**(1/2): 109-112.

Fang, H.-L., P. Ross, *et al.* (1993). A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling, and Open-Shop Scheduling Problems. *Proceedings of the Fifth International Conference on Genetic Algorithms,* San Mateo, Morgan Kaufmann.

Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*., Addison Wesley.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Kim, M., J. H. Jung, *et al*. (1996). Intelligent Scheduling and Monitoring for Multi-product Networked Batch Processes. *Computers & Chemical Engineering* **20**(Suppl): S1149-S1154.

Kizilisik, O. B. (1999). Predictive and Reactive Scheduling, Department of Industrial Engineering, Bilkent University, TR-06533 ANKARA**:** 1-8.

Ku, H.-M., D. Rajagopalan, *et al*. (1987). Scheduling in Batch Processes. 35-45.

Lee, C. Y. and J. Y. Choi (1995). A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights. *Computers & Operations Research* **22**(8): 857-869.

Lopes, J.P and J.C. Menezes (1998). Intelligent Systems for Penicillin Fermentation Process Modelling. *Computer Applications in Biotechnology* CAB7, 333-338.

Middendorf, M., B. Scheuermann, *et al.* (2002). An Evolutionary Approach to Dynamic Task Scheduling on FPGAs with Restricted Buffer. *Journal of Parallel and Distributed Computing* **62**(9): 1407-1420.

Montague, G. A. and J. Morris (1994). Neural-network contributions in Biotechnology. *Trends in Biotechnology* **12**: 312-324.

Morton, T. E. and D. W. Pentico (1993). *Heuristic Scheduling Systems:With Applications to Production System and Project Management*, John Wiley & Sons, INC.

Orçun, S., I. K. Altinelb, *et al*. (2001). General continuous time models for production planning and scheduling of batch processing plants: mixed integer linear program formulations and computational issues. *Computers & Chemical Engineering* **25**(2-3): 371-389.

Reklaitis, G. V. (1982). Review of Scheduling of Process Operations. *AIChE Symposium series* **78**(214): 119-132.

Rubin, P. A. and G. L. Ragatz (1995). Scheduling in a sequence dependent setup environment with genetic search. *Computers & Operations Research* **22**(1): 85-99.

Sabuncuoglu, I. and M. Bayiz (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of Oerational Research* **126**: 567-586.

Sand, G., S. Engell, *et al.* (2000). Approximation of an Ideal Online Scheduler for a Multiproduct Batch Plant. *Computers & Chemical Engineering* **24**: 361-367.

Shah, N. (1999). Single-Multisite Planing and Scheduling:Current Status and Future Challenges. *AIChE Symposium series* **320**(94): 75-90.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*., San Mateo, California, Morgan Kaufmann.

Yuan, J. Q. and P. A. Vanrolleghem (1998). One-Step-Ahead Product Predictor for Profit Optimisation of Penicillin Fermentation. *Proceedings 7th IFAC Conference on Computer Applications in Biotechnology*, Osaka Japan.