ICOTA '98

OPTIMIZATION

TECHNIQUES AND APPLICATIONS

VOLUME 2

Edited by

L. Caccetta[1] , K.L. Teo[1] , P.F. Siew[1] , Y.H. Leung[2] , L.S. Jennings[3], V. Rehbock[1]

[1] School of Mathematics and Statistics
Curtin University of Technology, Australia

[2]Australian Telecommunications Research Institute
Curtin University of Technology, Australia

[3] Department of Mathematics
The University of Western Australia

Scheduling mixed batch/continuous process plants with variable cycle time by splitting the optimisation problem

H.P. Nott and P.L. Lee*
School of Engineering
Murdoch University, Perth WA 6150, Australia

*Author to whom all correspondence should be addressed:
peter@eng.murdoch.edu.au

Abstract

Effective scheduling of operations in the process industry has the potential for high economic returns. Process plants containing both batch and continuous units present a difficult scheduling problem. When batch cycle times are part of the decision process, the complexity is increased significantly. The incorporation of variable batch cycle times into the scheduling model is important as it enables the "best" cycle time to be selected.

This paper considers an alternative way to represent the problem, based on optimising the cycle time outside the remainder of the problem. This is achieved by parameterising the cycle time as a function of continuous variables. The motivation for this work is an existing scheduling problem in the sugar industry. A smaller problem is considered and performance comparisons with the job shop formulation are made. This smaller problem contains the important characteristics of the sugar problem.

Keywords - scheduling, mixed integer programming, mathematical modelling, cycle time.

Motivation

This study is motivated by the need for an optimal scheduling policy in sugar milling. Efficient scheduling of operations from crystallisation through to raw sugar production, incorporating various batch (eg pans) and continuous units (eg fugals and dryers) has the potential for high economic return. The system incorporates numerous process interactions as well as the sharing of equipment between operations. A simplified outline of the milling system is shown in Figure 1. The multi-stage batch crystalliser operation of the pans are shown as a single batch pan system. This is followed by a communal, limited storage facility before the remainder of the downstream process, which is generally considered as operating continuously.

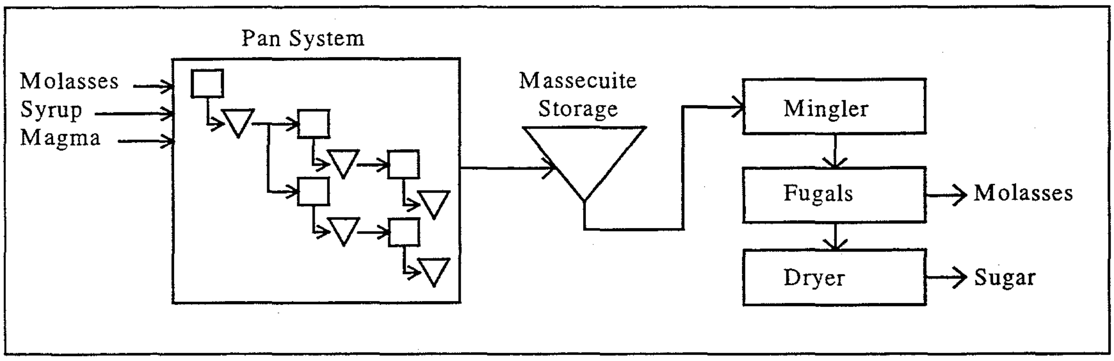

Figure 1 *Sugar mill flow-sheet.*

The presence of both batch and continuous units makes a difficult scheduling problem. Furthermore, although the batch units process a fixed quantity, they may take a variable length of time (cycle time). Constraints of the physical system are specified by bounds on the cycle time of the batch pans, the limited storage facility as well as possible flow rates through the continuous units. A profit is associated with the quantity of sugar produced, while there is a cost incurred in the processing of each pan operation. It also is costly to have a pan idle at any time and it is not advisable to make drastic changes to the continuous flowrate.

Introduction

When both batch and continuous processes are present, scheduling is traditionally accomplished by discretising time and considering the problem as a job shop scheduling problem (Kondili [4]). This produces a large mixed-integer linear programming problem (MILP) which is difficult to solve.

In general, the processing time of many batch operations is not a fixed characteristic of the batch unit or the batch type, and may be decided based on other system behaviour. Selecting the cycle time of a process has been recognised as an important design issue by several authors, see for example Kossik [5] and Montagna [7]. However, this has been implemented by representing cycle times as a function of other process characteristics, such as the size of the batch unit as in Montagna [7].

Beyond this, Nott [9] consider the cycle time of processes as independent decision variables. The increased complexity of solving this problem in the job shop scheduling context is dramatic due to permitting variable cycle times. However, the quality of the solution produced is far superior (Nott [9]).

The motivation of this paper is to trade-off these two issues: model complexity and solution quality. The method proposed optimises the cycle time as an external objective to the rest of the problem. Thus, cycle times are incorporated in the overall set of decision variables and the quality of solution improved accordingly. Furthermore, the inner MILP problem will have in essence fixed or pre-specified cycle times and be much simpler to solve.

Performance comparisons are made by considering a small problem which contains the important characteristics of the sugar mill scheduling problem. Computations are performed on an IBM RS6000/380. MILPs generated are solved using the CPLEX 4.0 (CPLEX [1]) solver (using a branch and bound algorithm), using AMPL (Fourer [2]) as a front end modelling language.

Simple Model

A simple model containing batch and continuous operations is considered. This consists of two batch units 'dropping' into a single storage unit, leading to a continuous production process, assumed to be a stream.

In this system, seen in Figure 2, batch cycle times may be variable and it is possible for the batch units to be idle, although this is discouraged.

Restrictions are imposed on the total amount in storage and the flowrate through the continuous unit. Changes in the continuous flowrate are also deterred.

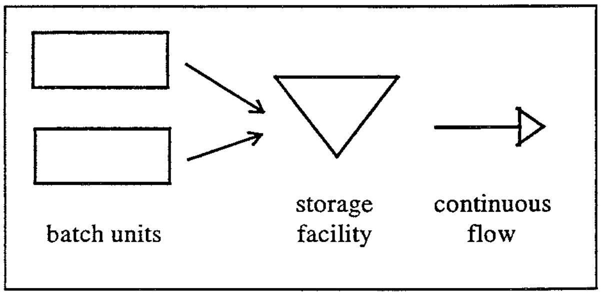

Figure 2 *Simple batch-continuous model.*

Thus the system objective is represented by

maximise total production through the system over the entire solution horizon
- costs associated with beginning individual batches
- penalty associated with scheduling idle periods
- penalty for changes in the continuous flowrate.

Job Shop Formulation And Optimal Objective Function Value

The job shop formulation is based on determining individual solution choices for each component of the system at each time interval over a given period of time, or *solution horizon.* As seen in the appendix, this formulation includes a large number of boolean relationships/constraints involving binary variables, to ensure feasibility. For example to bound cycle times to their feasible range, and to identify idle periods in the batch units.

For problem specifications given in the appendix, the optimal objective function value is 1415.0833. This involves 316260 simplex iterations and traverses 22448 branch and bound nodes. The solution is the found in 585.46 seconds solve time, and 598.20 seconds user time.

Split The Optimisation Problem

Permitting variable cycle times returns significantly better profit solutions. However, it is also responsible for a significant proportion of the complexity of this problem (Nott [8]). Simpler representations of the cycle time, such as being fixed over the solution horizon for each unit, or generating them from a known distribution produced inferior solutions. This section presents a method to optimise the cycle time as an outer objective to the remainder of the optimisation problem, as seen in Figure 3.

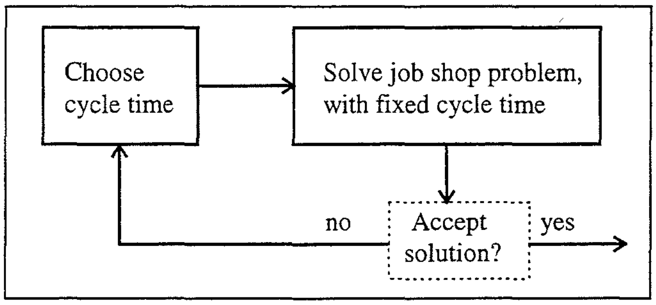

Figure 3 *Outer optimisation of the cycle time decision variable*

The outer optimisation is implemented by parameterising the discretely valued cycle time as a function of continuous variables. This is modelled as the sine-cosine function over time, *t*:

$$p(u,t) = p_1(u,t)\ \sin^2(\theta_u\, t - \phi_u) + p_2(u,t)\cos^2(\theta_u\, t - \phi_u)\,.$$

Time and so the cycle time function, $p(u,t)$ is considered to be a continuous quantity. A batch beginning on batch unit u, at discrete time unit t has fixed cycle time:

$$P(u,t) = \text{round}\ (\ p(u,t))\ \text{for}\ \ t \in \mathbf{Z}.$$

The cycle time function, $p(u,t)$ is bounded by bounded by exponential curves:

$$p1(u,t) = (maxP_u - expP_u + 0.49)\ e^{-b1_u(t-T0)} + expP_u$$

$$p2(u,t) = (minP_u - expP_u - 0.49)\ e^{-b2_u(t-T0)} + expP_u$$

where $maxP_u$ is the maximum, $minP_u$ is the minimum and $expP_u$ is the expected cycle time for batch unit u. The cycle time function, seen in Figure 4 permits choice of the cycle time between the minimum and maximum values. It also does not force cyclic schedules, while being realistic in that it restricts the cycle time toward its expected length further into the solution horizon.

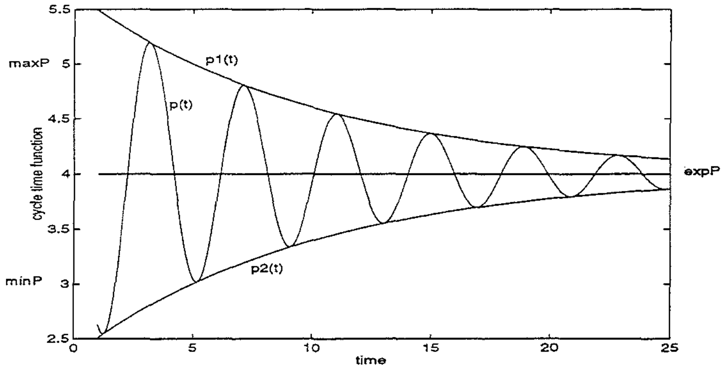

Figure 4 *Functional form of cycle times*

For each batch unit, the decision variables of the outer problem are the parameterisation variables: $b1$, $b2, 6$ and ϕ. These are each responsible for characteristics of the cycle time function:

- The parameterisation variables *b1* and *b2* determine the bounding curves. The least restrictive bounding curves correspond to *p1(t)=maxP+0.49, p2(t)=minP-0.49* for all *t*. Thus, *b1=b2=0* is the lower bound for these parameterisation variables. The most restrictive region has all batches beyond the present batch with cycle times of expected length or:

  $$p1\,(t=T0+P0) = expP+0.49 \text{ and } p2\,(t=T0+P0) = expP\text{-}0.49.$$

  Hence the limits for these parameterisation variables are:

  $$0 \le b1 \le \frac{-1}{P0}\ln\left(\frac{0.49}{maxP - expP + 0.49}\right) \text{ and } 0 \le b2 \le \frac{-1}{P0}\ln\left(\frac{-0.49}{minP - \exp P - 0.49}\right).$$

- Although the parameterisation variable 6 does not strictly represent the frequency of oscillations, the difference is negligible for realistic scenarios. One complete oscillation should be possible for consecutive discrete time points, ie *t=j* to *t=j+1*. Hence $0 < 6 \le \pi$ is assumed to sufficiently define the search region (calculated from *b1=b2=0*).

- The parameterisation variable ϕ enables the total processing time for the batch presently processing, *P0,* to lie on the cycle time curve at the *t=T0*. The choice of ϕ does effect the solution schedule produced and is considered an independent decision variable. All values $0 \le \phi \le \pi$ returning *P(T0)=P0* are considered.

Discussion, Search Strategies and Results

For the problem specified in the appendix, the parameterisation variable specifications given in Table 1 return the known optimal objective function.

Batch unit	*b1*	*b2*	θ	φ
#1	0.1	0.02	1.19	1
#2	0.02	0.03	1.46	1

Table 1 *A solution yielding the optimal*

Using the parameters specified in the appendix, Figure 5 shows a "slice" of the objective function surface (dependent on 6). In this case *b1=b2=0* and ϕ is determined from two feasible values for each unit (one returning a positive gradient, and the other a negative gradient). Figure 5 shows the non-smooth response of the objective function by choice of 6. In fact, the objective function response is extremely non-smooth to all parameterisation variables (Nott [8]).

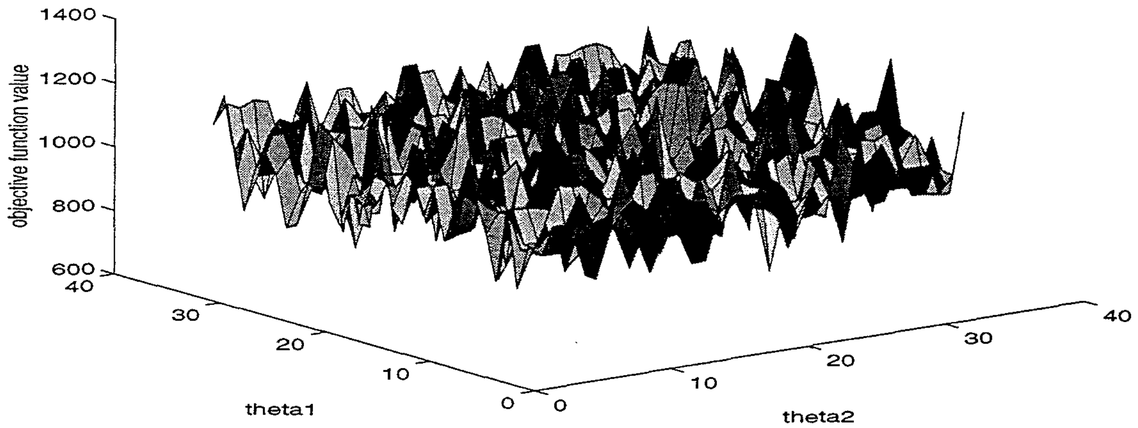

Figure 5 *Objective as function of theta (0.1:0.1:3.2) for two batch unit system.*

Three non-linear search strategies are considered. These are a basic Grid Search, a Nelder-Mead simplex approach and Simulated Annealing.

1. A ***Grid Search*** is implemented with 10,000 grid points, using the grid
   - $b1_1, b2_1, b1_2, b2_2 \in [0.1, 0.2]$;
   - $\theta_1, \theta_2 \in [0.1, 0.85, 1.6, 2.35, 3.1]$;
   - $\phi_1, \phi_2 \in [0, 0.8, 1.6, 2.4, 3.2]$;

The best objective function value of 1234.13 is returned in 302 seconds. The grid search would require an extremely fine grid to enable the optimum to be located. This, in turn would require an enormous number of grid points, and MILP problems solved.

2. The ***Nelder-Mead Search*** algorithm searches for a minima, using a simplex of $n+1$ points in an n-dimensional space. The simplex point yielding the highest objective function value is replaced by a new point defined by *reflection, contraction* or *expansion*. The MATLAB function, *fmins* (MATLAB [6]) was modified so that the initial simplex X is built by:

   $X_i = X0_i + \Delta$ where $X0$ is the initial point and Δ is the simplex generator.

   The quality of the solution returned is dependent on the starting point of the search and the size of the initial simplex. Setting the simplex generator to one third the domain, the results of searching from various starting points are seen in Table 2. Starting the search at the centre of the domain, Table 3 displays the results for various simplex generators. Both were unable to achieve the known optimum objective function value unless given it as the starting point.

$b1_1$	$b2_1$	$b1_2$	$b2_2$	θ_1	θ_2	ϕ_1	ϕ_2	Objective value	# its	user T
0.18	0.18	0.14	0.14	1.58	1.58	1.57	1.57	-1091.7667	171	85.29
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1234.1333	155	65.30
0.2	0.2	0.1	0.1	1.0	1.0	1.0	1.0	-1135.4667	155	68.75
0.3	0.3	0.2	0.2	1.0	2.0	3.0	1.5	-1090.0667	135	57.24
0.1	0.2	0.2	0.2	1.0	1.0	1.0	1.0	-1081.7667	156	70.63
0.1	0.1	0.1	0.1	2.0	2.0	2.0	2.0	-1191.3667	158	65.96
0.3	0.1	0	0	1.5	1.0	2.0	2.5	-1177.5667	164	59.82
0.1	0.05	0.08	0.03	2.3	2.4	2.8	2.8	-1347.3167	159	61.33
0.1	0	0	0	1.2	1.5	1.0	1.0	-1347.3167	162	67.76

0.1	0.02	0.02	0.03	1.19	1.46	1.0	1.0	-1415.0833	156	65.49

Table 2 *Result of various starting points using simplex generator with defined size*

$\Delta b1_1$	$\Delta b2_1$	$\Delta b1_2$	$\Delta b2_2$	$\Delta\theta_1$	x$\Delta\theta_2$	$\Delta\phi_1$	$\Delta\phi_2$	Obj value	# its	UserT
0.1236	0.1236	0.0927	0.0927	1.0472	1.0472	1.0472	1.0472	-1091.7667	171	85.29
0.1	0.1	0.1	0.1	1.0	1.0	1.0	1.0	-1191.3667	162	67.36
0.05	0.05	0.05	0.5	0.5	0.5	0.5	0.5	-1091.7667	147	75.34
0.1	0.1	0.1	0.1	1.0	0.5	1.1	0.7	-1091.7667	162	80.87
0.2	0.2	0.1	0.1	1.5	1.5	1.5	1.5	-1091.7667	154	71.92
-0.1236	-0.1236	-0.0927	-0.0927	-1.0472	-1.0472	-1.0472	-1.0472	-1334.0667	170	67.96
-0.09	-0.14	-0.07	-0.11	0.8	0.8	1.3	1.3	-1330.0667	159	62.13
-0.09	-0.14	-0.07	-0.11	0.8	0.8	1.3	1.3	-1330.0667	159	62.13

Table 3 *Result of various sizes for the simplex generator, using defined starting point*

The Nelder-Mead search method is unsuccessful and it is presumed that this is due to the false assumption of a smooth objective function.

3. Finally, the publicly available ***Adaptive Simulated Annealing*** (Ingber [3]) was used to perform the non-linear search for the cycle time parameterisation variables. Simulated annealing is based on generating new configuration states and deciding whether to accept or reject these states based on reducing "temperature". The solution quality and performance is dependent on the starting point of the search. For the minimisation problem, after some amount of tuning the algorithmic parameters, the results contained in Table 4 show that the ASA search was also unsuccessful unless given the optimum point to begin the search.

$b1_1$	$b2_1$	$b1_2$	$b2_2$	θ_1	θ_2	ϕ_1	ϕ_2	Objective value	User T
0.18	0.18	0.14	0.14	1.58	1.58	1.57	1.57	-1347.3167	583.18
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	-1347.3167	595.35
0.2	0.2	0.1	0.1	1.0	1.0	1.0	1.0	-1347.3167	611.14
0.3	0.3	0.2	0.2	1.0	2.0	3.0	1.5	-1347.3167	596.49
0.1	0.2	0.2	0.2	1.0	1.0	1.0	1.0	0	1.16
0.1	0.1	0.1	0.1	2.0	2.0	2.0	2.0	-1347.3167	587.86
0.3	0.1	0	0	1.5	1.0	2.0	2.5	-1347.3167	597.97
0.1	0.05	0.08	0.03	2.3	2.4	2.8	2.8	0	1.13
0.1	0	0	0	1.2	1.5	1.0	1.0	-1347.3167	571.95
0.1	0.02	0.02	0.03	1.19	1.46	1.0	1.0	-1415.0833	638.69

Table 4 *ASA search from various starting points.*

Thus, a suitable non-linear search strategy was not able to be found that could handle the response surface of the objective function curve.

Conclusions

Modelling batch process with variable cycle times return substantially better quality solutions. However, it is also responsible for a substantial increase in the complexity of the problem. Optimisation of the cycle time outside the remainder of the problem appears a good algorithmic alternative. This would maintain the solution quality associated with variable cycle times, without the associated complexity (in the MILP solution). However attempts at finding an acceptable non-linear search strategy to perform this outer optimisation have been unsuccessful.

Further work is considering other methods to split the optimisation process.

Nomenclature

Parameters Specifications		*Value*
set U=1..num_units	domain of batch units	1..2
set T=T0..TF	the solution horizon	1..25
Z{U}	size of each batch	[8, 10]
minP{U}..maxP{U}	feasible batch cycle times	[3, 4] .. [5,6]
exp{U}	expected batch cycle time	[4, 5]
minX..maxX	feasible flowrates	2.5 .. 5.0
minS..maxS	feasible storage amounts	2..15
S0	initial amount in storage	10
sell	turn-over/unit produced	20
cost{U}	cost price per batch begun	[60, 50]
penI	penalty for idle periods	100
penX	flow rate changes penalty	1.0

Variables

X {t}	total flow in continuous unit: time *t* to *t+1*
S {t}	amount in storage at time *t*
PR {t}	cumulative production record up to time *t*
W {u, t} binary	= 1 if a batch begins on unit *u* at time *t*
C {u, t} integer	# of real/idle batches on *unit u* up to time *t*
Y {u, t} binary	type of batch processing: 0 = idle, 1 = real
B {u, t} binary	= 1 if a real batch begins on unit *u* at time *t*
E {u, t} binary	= 1 if a real batch ends on unit *u* at time *t*
A {u, t} binary	= 1 if unit *u* active at time *t*

Job Shop Formulation

$$\text{maximise } sell * PR_{TF} - \sum_{\substack{u\in U\\ t\in T}} cost_u * B_{u,t} - penI * \sum_{\substack{u\in U\\ t\in T}} (1-Y_{u,t}) - \sum_{t\in T: t>T0} penX * |X_t - X_{t-1}|$$

subject to

initial batch	$\forall u \in U$	$C_{u,T0} = 1$
	$\forall u \in U$	$E_{u,T0} = 0$
batch count	$\forall u \in U, \forall t \in T: t > T0$	$C_{u,t} = C_{u,t-1} + W_{u,t}$
min cycle times	$\forall u \in U, \forall t \in T,$ $\forall tt \in (t+1)..min(t+minP_u-1, TF)$	$C_{u,tt} - C_{u,t} \le (TF-T0)*(1-B_{u,t})$
max cycle times	$\forall u \in U, \forall t \in T: t \ge T0 + maxP_u$	$C_{u,t} - C_{u,t-maxP_u} \ge 1$
unit idle periods	$\forall u \in U, \forall t \in T$	$W_{u,t} + Y_{u,t} \ge 1$
feasibility	$\forall u \in U, \forall t \in T: t > T0$	$Y_{u,t-1} - W_{u,t} \le Y_{u,t} \le Y_{u,t-1} + W_{u,t}$
assign batch begins	$\forall u \in U, \forall t \in T$	$W_{u,t} + Y_{u,t} - 1 \le B_{u,t} \le (W_{u,t} + Y_{u,t})/2$
assign batch ends	$\forall u \in U, \forall t \in T: t > T0$	$W_{u,t} + Y_{u,t-1} - 1 \le E_{u,t} \le (W_{u,t} + Y_{u,t-1})/2$
initial storage		$S_{T0} = S0$
update storage	$\forall t \in T: t > T0$	$S_t = S_{t-1} - X_{t-1} + \sum_{u\in U} E_{u,t} * Z_u$
storage feasibility	$\forall t \in T$	$S_t - X_t \ge minS$
initial production		$PR_{T0} = 0$
update production	$\forall t \in T$	$PR_t = PR_{t-1} + X_{t-1}$

VARIABLES :

$\forall t \in T:\ minX \le X_t \le maxX$ $\quad\forall t \in T:\ minS \le S_t \le maxS$ $\quad\forall t \in T:\ PR_t \ge 0$

$\forall u \in U, \forall t \in T:\ W_{u,t}, C_{u,t}, Y_{u,t}, B_{u,t}, E_{u,t} \in \{0,1\}$

Parameterisation Variables

b1{u}	determines upper bounding curve for unit *u*
b2{u}	determines lower bounding curve for unit *u*
θ{u}	frequency for parameterisation curve for unit *u*
∅{u}	time lag for parameterisation curve for unit *u*
p1{u,t}	upper bounding curve for unit *u* over time *t*
p2{u,t}	lower bounding curve for unit *u* over time *t*
P{u,t}	parameterisation function for the cycle time

$$p1_{u,t} = (maxP_u - expP_u + 0.49)\ e^{-b1_u(t-T0)} + expP_u$$

$$p2_{u,t} = (minP_u - expP_u - 0.49)\ e^{-b2_u(t-T0)} + expP_u$$

$$P_{u,t} = round\left[p1_{u,t}\ \sin^2(\theta_u * t - \phi_u) + p2_{u,t}\ \cos^2(\theta_u * t - \phi_u)\right]$$

Split Optimisation MILP Formulation

$$\text{maximise } sell * PR_{TF} - \sum_{\substack{u\in U\\ t\in T}} cost_u * W_{u,t} - penI * \sum_{\substack{u\in U\\ t\in T}} (1 - A_{u,t}) - \sum_{t\in T:t>T0} penX * |X_t - X_{t-1}|$$

subject to

feasible batch schedule	$\forall u \in U,\ \forall t \in T$	$\sum_{tt\in t..\min(t+P_{u,t}-1,TF)} W_{u,tt} \le M * (1 - W_{u,t})$
initial anount in storage		$S_{T0} = S0$
update storage	$\forall t \in T: t > T0$	$S_t = S_{t-1} - X_{t-1} + \sum_{u\in U} E_{u,t} * Z_u$
feasible storage	$\forall t \in T$	$S_t - X_t \ge minS$
initial production		$PR_{T0} = 0$
update production	$\forall t \in T$	$PR_t = PR_{t-1} + X_{t-1}$

VARIABLES :

$\forall t \in T:\ minX \le X_t \le maxX$ $\quad \forall t \in T:\ minS \le S_t \le maxS$ $\quad \forall t \in T:\ PR_t \ge 0$

$\forall u \in U, \forall t \in T:\ W_{u,t}, E_{u,t}, A_{u,t} \in \{0,1\}$

References

[1] CPLEX Optimization Inc. (1996), CPLEX-MIP, Suite 279, 930 Tahoe Blvd., Bldg. 802, Incline Village, Nevada.

[2] R. Fourer, D.M. Gay and B.W. Kernigham (1993) *AMPL : A Modeling Language for Mathematical Programming,* Boyd Fraser, Massachusetts.

[3] L. Ingber (1997), *Adaptive Simulated Annealing (ASA),* Lester Ingber Research, P.O. Box 857, McLean, VA 22101-0857, http://www.ingber.com.

[4] E. Kondili, C.C. Pantelides and R.W.H. Sargent (1993). A general algorithm for short-term scheduling of batch operations - I. MILP formulation. *Comp. Chem. Engng.* 17(2)211-227.

[5] J.M. Kossik and G. Miller (1994) Optimize cycle times for batch biokill systems. *Chem. Engng. Progress.* 90(10):45-51.

[6] MATLAB (1995), *The Student Edition of MATLAB Version 4 User Guide,* The MATHS WORKS Inc, Prentice Hall, Englewood Cliffs New Jersey.

[7] J.M. Montagna, O.A. Iribarren and F.C. Galiano (1994) The design of multiproduct batch plants with process performance models. *Chem. Engng. Res. and Design.* 72(A6):783-791.

[8] H.P. Nott (1997), *Modelling Alternatives for Scheduling Mixed-Batch/Continuous Process Plants with Variable Cycle Time,* PhD Thesis in preparation, School of Engineering, Murdoch University.

[9] H.P. Nott and P.L. Lee (1997), Complexity of scheduling mixed batch/continuous process with variable cycle time, submitted to *European Journal of Operational Research.*