AIChE WILEY InterScience DISCOVER SOMETHING GREAT

Process Scheduling Under Uncertainty Using Multiparametric Programming

Zukui Li and Marianthi G. Ierapetritou
Dept. of Chemical and Biochemical Engineering, Rutgers University, Piscataway, NJ 08854

DOI 10.1002/aic.11351
Published online October 29, 2007 in Wiley InterScience (www.interscience.wiley.com).

In this article, the problem of process scheduling under uncertainty was studied using multiparametric programming method. Based on the uncertainty type (prices, demands, and processing times), the scheduling formulation results in different parametric problems including multiparametric mixed integer linear (mpMILP), quadratic (mpMIQP), and general nonlinear programming (mpMINLP) problem. This article analyzes the solution characteristics and proposes a novel solution framework for specific mpMILP/mpMIQP problems addressing a wide variety of scheduling problems under different types of uncertainty, which are modeled as coefficient in objective function, as coefficient of integer variable and right hand side vector of the constraints. The main idea of the proposed framework is to decompose the problem into a series of smaller subproblems, each of them producing the parametric information around a given parameter value. The parametric solution of every subproblem is retrieved by solving a series of multiparametric linear programming (mpLP) and mixed integer linear/nonlinear programming problems (MILP/MINLP). Several examples were solved to analyze the complexity and the effectiveness of the proposed method. © 2007 American Institute of Chemical Engineers *AIChE J,* 53: 3183–3203, 2007
Keywords: process scheduling, uncertainty, multiparametric programming

Introduction

In real plants, uncertainty is a very important concern that is coupled with the scheduling process since many of the parameters that are associated with scheduling are not known exactly. Parameters like raw material availability, prices, machine reliability, and market requirements vary with respect to time and are often subject to unexpected deviations. Having ways to systematically consider uncertainty is as important as having the scheduling model itself. Methodologies for process scheduling under uncertainty aim at producing feasible, robust, and optimal schedules. In essence, uncertainty consideration plays the role of validating the use of mathematical models and preserving plant feasibility and viability during operations. According to the different treatment of uncertainty, process scheduling methods can be classified into two groups: reactive scheduling and preventive scheduling.

Reactive scheduling is a process of modifying the existing schedule during the process operation to adapt to changes (uncertainty) in production environment, such as disruptive events, rush order arrivals, order cancellations, or machine breakdowns. For this type of uncertainty there is not enough information prior to realization of the uncertain parameters that will allow a preventive action. There are a number of papers in the literature that focus on this kind of methodology.[1–8]

Preventive scheduling on the other hand generates scheduling policies before uncertainty occurs. Detailed classification of preventive scheduling includes: stochastic scheduling, robust optimization method, fuzzy programming method, and sensitivity analysis and parametric programming method.

Stochastic scheduling is the most commonly used approach in the literature for preventive scheduling,[9–14] in which the

Correspondence concerning this article should be addressed to M. G. Ierapetritou at marianth@soemail.rutgers.edu.

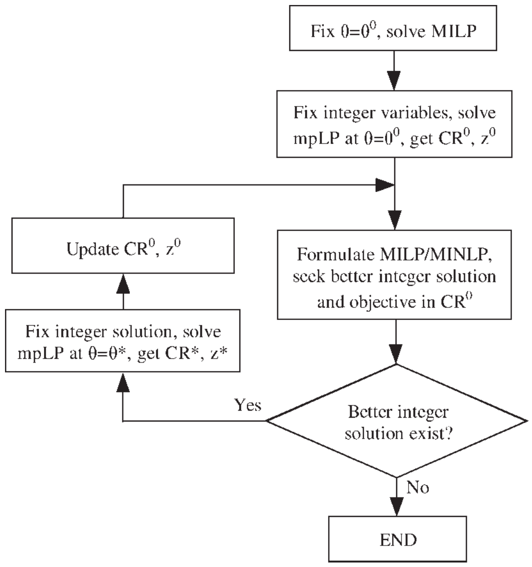

Figure 1. Flow chart of multiparametric programming for one critical region.

original deterministic scheduling model is transformed into stochastic model treating the uncertainties as stochastic variables. Within the stochastic programming models we can distinguish the following categories: two-stage/multistage stochastic programming and chance constraint programming-based approach. Robust scheduling aims at building the preventive schedule to minimize the effects of disruptions on the performance measure.[15–18] It tries to ensure that the predictive and realized schedules do not differ drastically while maintaining a high level of schedule performance. Fuzzy programming also addresses optimization problems under uncertainty and is applied in uncertain scheduling.[19–21] It can be used in the situation when probabilistic information is not available. Fuzzy set theory and interval arithmetic are used to describe the imprecision and uncertainties in process parameters. An alternative way in preventive scheduling is using MILP sensitivity analysis and parametric programming. These methods are important as they can offer significant analytical results to problems related to uncertainty. Sensitivity analysis is used to determine how a given model output depends upon the input parameters.[22] Parametric programming serves as an analytic tool by mapping the uncertainties in the optimization problem to optimal solution alternatives. From this point of view, parametric programming provides the exact mathematical solution of the optimization problem under uncertainty.

In the literature, multiparametric linear programming (mpLP) and multiparametric quadratic programming (mpQP) problem are well studied due to the relatively smaller problem complexity.[23–26] General multiparametric nonlinear programming (mpNLP) problem is not well addressed because the exact solution of mpNLP is very complex.[27] On the other hand, existing multiparametric mixed integer programming methods are based on the solution of mpLP or mpQP subproblems.[28,29] For the multiparametric mixed integer quadratic programming (mpMIQP), there is still not an efficient method for solving the general problem. Dua et al.[30] proposed a methodology to address this problem for the special case derived from optimal control problem.

In the literature, the multiparametric programming method has been mainly applied in online optimization, process control, and process synthesis.[31,32] All these problems are of relatively small scale. There are not many records on the application of parametric programming in process scheduling problem. To our knowledge, only Ryu and Pistikopoulos[33,34] has reported the application of parametric programming to a zero-waiting scheduling problem and Pistikopoulos and coworkers[35–37] have applied parametric programming for the solution of process planning problem.

Formulating the scheduling problem under uncertainty as a multiparametric programming problem gives rise to mpMILP, mpMIQP, or mpMINLP problem depending on the type of uncertain parameters, which is actually not the multiparametric mixed integer linear programming (mpMILP) problem studied mostly in the literature. The other important characteristic for multiparametric programming in scheduling formulation is that the problem is generally large scale, because the deterministic formulation of process scheduling problem involves a large number of constraints and integer variables.

In this article, we proposed a framework to solve the mpMILP and mpMIQP problems generated from uncertain process scheduling problem. The framework is based on the idea of decomposing the original problem into a series of smaller problems. The parametric solution of each subproblem provides the solution around a given parameter value. The article's structure is as follows: in Problem Formulation section, the problem formulation is given and the solution characteristic of the multiparametric programming problem derived from scheduling formulation is analyzed; in Proposed Framework section, we explain in detail the framework we

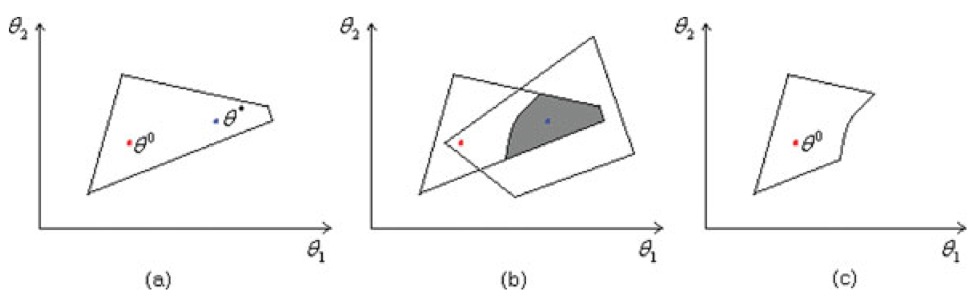

Figure 2. Critical region updating process.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

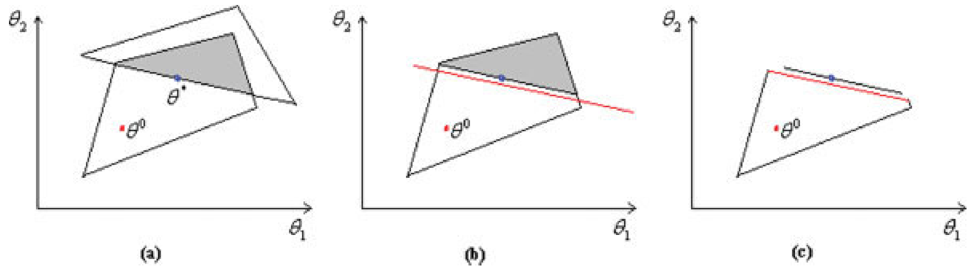

Figure 3. Boundary contraction illustration.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

developed, and several examples are illustrated in Example section to illustrate the complexity and effectiveness of the proposed method; finally, the work is concluded in Conclusion section.

Problem Formulation

The mathematical model used for batch process scheduling in this article follows the main idea of the continuous time formulation proposed by Ierapetritou and Floudas.[38] The general model involves the following objective and constraints:

$$\max \sum_{s,n} \text{price}_s d_{s,n} \tag{1}$$

$$\text{s.t.} \qquad \sum_{i \in I_j} wv_{i,j,n} \leq 1 \qquad \forall i \in I \tag{2}$$

$$st_{s,n} = st_{s,n-1} - d_{s,n} - \sum_{i \in I_s} \rho^P_{s,i} \sum_{j \in J_i} b_{i,j,n} + \sum_{i \in I_s} \rho^c_{s,i} \sum_{j \in J_i} b_{i,j,n-1} \quad \forall s \in S, \forall n \in N \tag{3}$$

$$st_{s,n} \leq st_s^{\max} \quad \forall s \in S, \forall n \in N \tag{4}$$

$$v_{i,j}^{\min} wv_{i,j,n} \leq b_{i,j,n} \leq v_{i,j}^{\max} wv_{i,j,n} \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{5}$$

$$\sum_n d_{s,n} \geq r_s \quad \forall s \in S \tag{6}$$

$$Tf_{i,j,n} = Ts_{i,j,n} + \alpha_{i,j} wv_{i,j,n} + \beta_{i,j} b_{i,j,n} \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{7}$$

$$Ts_{i,j,n+1} \geq Tf_{i,j,n} - H(1 - wv_{i,j,n}) \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{8}$$

$$Ts_{i,j,n+1} \geq Tf_{i',j,n} - H(1 - wv_{i',j,n}) \quad \forall i, i' \in I_j, \forall j \in J, \forall n \in N \tag{9}$$

$$Ts_{i,j,n+1} \geq Tf_{i',j',n+1} - H(1 - wv_{i',j',n}) \quad \forall i, i' \in I_j, i \neq i' \ \forall j, j' \in J, \forall n \in N \tag{10}$$

$$Ts_{i,j,n+1} \geq Ts_{i,j,n} \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{11}$$

$$Tf_{i,j,n+1} \geq Tf_{i,j,n} \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{12}$$

$$Ts_{i,j,n} \leq H \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{13}$$

$$Tf_{i,j,n} \leq H \quad \forall i \in I, \forall j \in J_i, \forall n \in N \tag{14}$$

In the above formulation, the objective function is the profit (different performance measures can be used like makespan); allocation constraints (2) state that only one of the tasks can be performed in each unit at an event point (n); constraints (3) represent the material balances for each state (s) expressing that at each event point (n) the amount $st_{s,n}$ is equal to that at event point ($n - 1$), adjusted by any amounts produced and consumed between event points ($n - 1$) and (n), and delivered to the market at event point (n); the storage and capacity limitations of production units are expressed by constraints (4) and (5); constraints (6) are written to satisfy the demands of final products; and constraints (7) to (14) represent time limitations due to task duration and sequence requirements in the same or different production units. Detailed description of the symbols in the above formulation is provided in the notation section of the article. It should be noticed that minimum product demand and minimum processing time in the uncertain range can be used to identify an appropriate event point number before the multiparametric solution process to avoid the loss of solution optimality.

Although the above scheduling formulation has been used, the proposed methodology in this article is not limited to this one. It can also be applied onto other scheduling formulations. For ease in the presentation, the scheduling model (1)–(14) (or other scheduling formulations) can be compactly written as the following mixed integer linear programming (MILP) problem:

$$\begin{aligned} \min \quad & z = cx \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned} \tag{P1}$$

Table 1. Data for Example 1

Unit	Capacity	Suitability	Processing Time	State	Storage Capacity	Initial Amount	Price
Unit 1	100	Task 1	4.5	State 1	Unlimited	Unlimited	0
Unit 2	75	Task 2	3.0	State 2	100	0.0	0
Unit 3	50	Task 3	1.5	State 3	100	0.0	0.7
				State 4	Unlimited	0.0	1.0

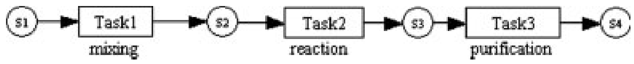

Figure 4. State Task Network (STN) of Example 1.

where the decision variable vector x is composed by both continuous and binary variables. When uncertainty is incorporated into the scheduling model, problem (P1) becomes:

$$\begin{aligned} \min \quad & z = (c + \theta^T D)x \\ \text{s.t.} \quad & (A + F\theta)x \geq b + E\theta \\ & x \geq 0 \\ & \theta_j^l \leq \theta_j \leq \theta_j^u, \quad j = 1,..,m \end{aligned} \tag{P2}$$

where θ_j represent uncertain parameter. According to the type of the uncertainty, the uncertain parameter can appear in the objective coefficients (e.g. price of product), right hand side (RHS) of the constraints (e.g. demand of product), left hand side (LHS) matrix of the constraints (e.g. processing time, recipe).

In the view of mathematical programming, problem (P2) represents a multiparametric programming problem. The complete solution of a multiparametric programming problem is composed by the complete set of critical regions and optimal objective functions described with respect to uncertain parameters. The critical region is defined as the range of parameter values where the same solution remains optimal. The direct objective of this article is to provide an efficient way to solve (P2) which is derived from uncertain process scheduling problem.

Among the mixed integer multiparametric programming problem, mpMILP and mpMIQP are mostly studied in the literature as stated in the introduction. mpMILP problem is generally defined as follows:

$$\begin{aligned} \min \quad & z = cx + D\theta \\ \text{s.t.} \quad & Ax \geq b + E\theta \\ & x \geq 0 \\ & \theta_j^l \leq \theta_j \leq \theta_j^u, \quad j = 1,..,m \end{aligned} \tag{P3}$$

A general mpMIQP problem has the following form, comprising by a quadratic objective function:

$$\begin{aligned} \min \quad & z = x'Hx + (c + \theta^T D)x + F\theta \\ \text{s.t.} \quad & Ax \geq b + E\theta \\ & x \geq 0 \\ & \theta_j^l \leq \theta_j \leq \theta_j^u, \quad j = 1,..,m \end{aligned} \tag{P4}$$

Table 2. Demand Uncertainty for Example 1

Parameter	Value	Variation Range
Demand of S3	θ_1	$0 \leq \theta_1 \leq 50$
Demand of S4	$50 + \theta_2$	$-50 \leq \theta_2 \leq 50$

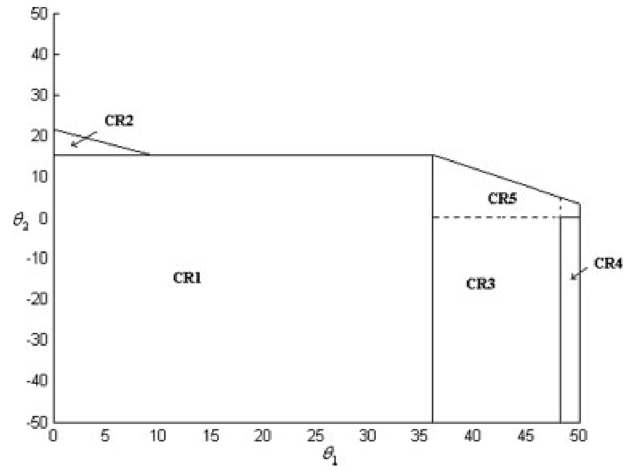

Figure 5. Critical region of Example 1 with demand uncertainty

Considering the different type of uncertainties that can appear in the scheduling problems, we can distinguish the following cases:

Case 1: When only RHS uncertainties are involved, problem (P2) gives rise to a special mpMILP problem shown above as problem (P3).

Case 2: When uncertainty is involved in the parameters of the objective function, or simultaneously on the RHS and objective function, problem (P2) becomes a special case of mpMIQP described by problem (P4).

Case 3: When LHS uncertainty is involved but only appears as the coefficient of binary variable (RHS and objective uncertainty can also be involved), the LHS uncertainty $x_i\theta_j$ can be transformed into RHS uncertainty through the following reformulation:

$$y_{ij} = x_i\theta_j \tag{15}$$

$$x_i\theta_j^l \leq y_{ij} \leq x_i\theta_j^u \tag{16}$$

$$\theta_j - \theta_j^u(1 - x_i) \leq y_{ij} \leq \theta_j - \theta_j^l(1 - x_i) \tag{17}$$

where y_{ij} is new variables which are introduced to linearize the bilinear term between binary variable (x_i) and uncertain parameter (θ_j) using Al-Khayyal linearization.[39]

Case 4: When LHS uncertain parameter appears as coefficient of continuous variables, problem (P2) becomes an mpMINLP problem.

Table 3. Solution of Example 1 with Demand Uncertainty

Schedule (Appendix A)	Parametric Objective	Critical Region
A.1	-90.46	CR1
A.2	$-72.097 + 0.029\theta_2$	CR2
A.3	$-96.14 + 0.158\theta_1$	CR3, CR5
A.4	-88.55	CR4

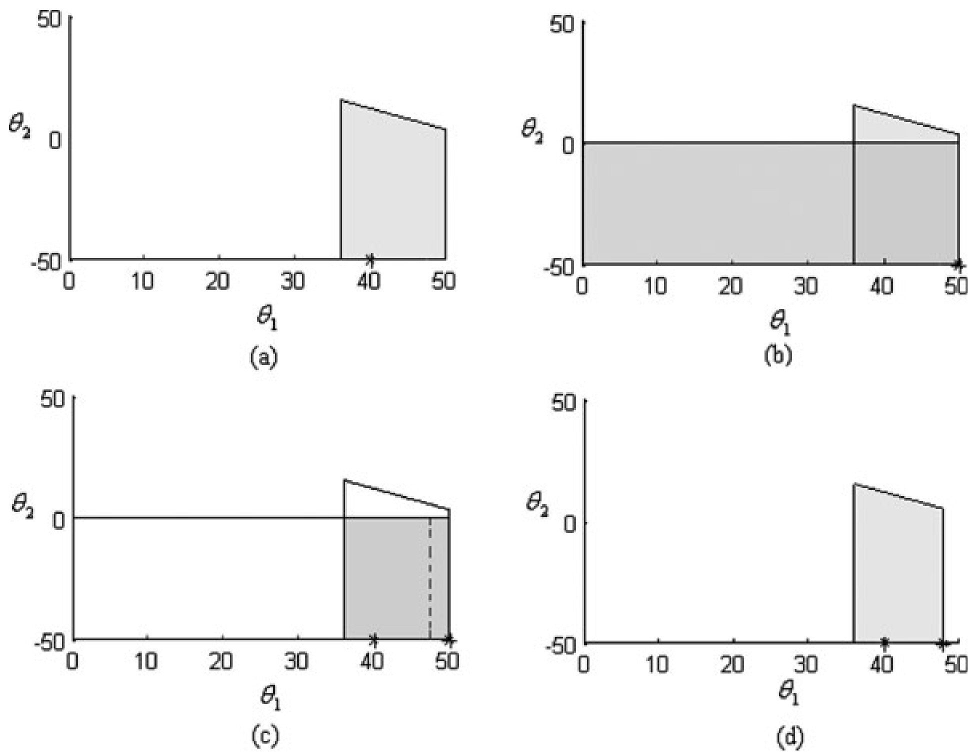

Figure 6. Process of solving one critical region.

Before presenting the proposed methodology for solving problem (P2), let's first look at some of the characteristics of the parametric solution. In an extreme case, the solution of (P2) can be retrieved by enumerating all the feasible integer solutions, solving the corresponding mpLP problem (by fixing those integer variables), and comparing all of the objective functions in every intersection of the critical regions of all the mpLP problems. So the characteristics of the parametric solution of (P2) can be analyzed based on the parametric solution of the mpLP problem.

For a standard linear programming problem:

$$\begin{aligned} \min \quad & z = cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{P5}$$

where x contain only continuous variables. In the following formula, we use subscript B to describe the column index corresponding to a basis variable and N to describe the column index corresponding to a nonbasis variable. For example, A_B is the matrix composed by the columns corresponding to the basic variables, and A_N is the matrix corresponding to the nonbasic variables. Then the following optimality conditions describe the optimal solution:

$$\text{Optimality conditions}: \quad A_B^{-1}b \geq 0 \tag{18}$$

$$c_N - c_B A_B^{-1} A_N \geq 0 \tag{19}$$

$$\text{Optimal objective}: \quad z^* = c_B A_B^{-1} b \tag{20}$$

When b,c involve uncertainty, they can be substituted by $b + E\theta, c + \theta^T D$ respectively and (P5) becomes an mpLP problem. The optimality conditions (18) and (19) form the critical region of an mpLP problem, and (20) gives the optimal objective function, as follows:

$$-A_B^{-1}E\theta \leq A_B^{-1}b \tag{21}$$

$$(D_B A_B^{-1} A_N - D_N)^T \theta \leq (c_N - c_B A_B^{-1} A_N)^T \tag{22}$$

$$z^*(\theta) = c_B A_B^{-1} b + (D_B A_B^{-1} b)^T \theta + (c_B A_B^{-1} E)\theta + \theta^T D_B A_B^{-1} E\theta \tag{23}$$

Examining the form of constraints (21)–(22), it can be concluded that when only RHS and objective uncertainty exists, the critical region of mpLP is formed by linear inequalities. As stated previously, the solution of multiparametric-mixed integer problem (P2) can be retrieved by comparing all of the objective functions in every intersection of the critical regions

Table 4. Price Uncertainty for Example 1

Parameter	Value	Variation Range
Price of S3	$0.7 + \theta_1$	$-0.5 \leq \theta_1 \leq 0.5$
Price of S4	$1 + \theta_2$	$-0.5 \leq \theta_2 \leq 0.5$

Table 5. Solution of Example 1 with Price Uncertainty

Schedule (Appendix B)	Parametric Objective	Critical Region
B.1	$-90.46 - 36.06\theta_1 - 65.22\theta_2$	CR1
B.2	$-71.47 - 71.47\theta_2$	CR2
B.3	$-88.55 - 55.07\theta_1 - 50\theta_2$	CR3, CR4

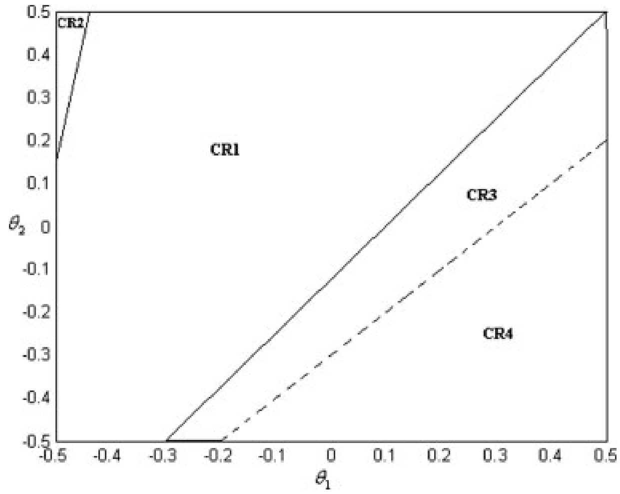

Figure 7. Critical region of Example 1 with price uncertainty.

of all the mpLP problems. The comparison between two linear objective functions will only generate linear inequalities, and once nonlinear (quadratic, fractional) objective function is involved in the comparison, the resulted inequality can be either linear or nonlinear inequality. Consequently, the following solution characteristics of (P2) can be concluded:

(i) When only RHS or only objective uncertainty exists, the optimal parametric objective function of the corresponding mpMILP or mpMIQP problem is linear function, the critical region is formed by linear inequalities;

(ii) When RHS and objective uncertainty both exist, the optimal parametric objective function of the corresponding mpMIQP problem is quadratic function, and the critical region is formed by linear/quadratic inequalities;

(iii) When LHS uncertainty appears as coefficient of continuous variable, the objective function of the corresponding mpMINLP is fractional polynomial function of uncertain parameters, and the critical region is formed by fractional polynomial inequalities.

Considering the solution complexity of general mpNLP and mpMINLP problem, in this article, we study the consideration of all the above cases except the case where LHS uncertainty appears in the coefficients of the continuous variables. In the scheduling formulation (1)–(14), there might be LHS uncertainties in production recipe $\rho^P_{s,i}$ and $\rho^c_{s,i}$ of constraints (3) and in the coefficients $\beta_{i,j}$ of processing time of constraints (7). To address the processing time uncertainty using the proposed method, we reformulate the duration constraints so that the LHS uncertain parameters only appear as coefficients of integer variables and can be transformed into RHS uncertainty. The original duration constraint (7) is reformulated as follows:

$$Tf_{i,j,n} = Ts_{i,j,n} + \alpha_{i,j} wv_{i,j,n} + \beta_{i,j} b_{i,j,n} + \theta wv_{i,j,n} \qquad (24)$$

Table 6. Price and Demand Uncertainty for Example 1

Parameter	Value	Variation Range
Price of S3	$0.7 + \theta_1 - 0.01\theta_2$	$-0.5 \le \theta_1 \le 0.5$
Demand of S4	$50 + \theta_2 - 20\theta_1$	$-50 \le \theta_2 \le 50$

Table 7. Solution of Example 1 with Demand and Price Uncertainty

Schedule (Appendix C)	Parametric Objective	Critical Region
C.1	$-90.46 - 36.06\theta_1 + 0.36\theta_2$	CR1, CR9
C.2	-71.47	CR2, CR3
C.3	$-88.55 - 55.07\theta_1 + 0.55\theta_2$	CR4
C.4	$-88.55 - 49.07\theta_1 + 0.25\theta_2 - 20\theta_1^2 + 1.2\theta_1\theta_2 - 0.01\theta_2^2$	CR5
C.5	$-73.55 - 105.07\theta_1 + 1.55\theta_2$	CR6
C.6	$-72.1 - 32.15\theta_1 + 0.34\theta_2 - 29.4\theta_1^2 + 1.76\theta_1\theta_2 - 0.015\theta_2^2$	CR7
C.7	$-87.66 - 50.13\theta_1 + 0.35\theta_2 - 23.32\theta_1^2 + 1.4\theta_1\theta_2 - 0.012\theta_2^2$	CR8

In the new formulation, $\alpha_{i,j}$ and $\beta_{i,j}$ is calculated based on nominal processing time, and uncertainty is modeled through the last term $\theta wv_{i,j,n}$.

In the process of comparing all the objective functions in every intersection of the critical regions of all the mpLP problems, the original critical region of an mpLP problem might be separated or cut-off partly, thus the final critical region of an mpMILP or mpMIQP might be nonconvex. The proposed method in this article addresses this case by separating the nonconvex critical region into several parts, and those separated critical regions might have an overlapped area.

Proposed Framework

The proposed multiparametric programming framework is based on the idea of decomposing the parameter space and solve for a specific point in the parameter space which can be either an infeasible point or a point that belongs to a critical region of the solution space. So the basic idea is to find the critical region that covers any given point in the parameter space and finally combine these separate parametric solutions. In other words, the following problem is studied: What

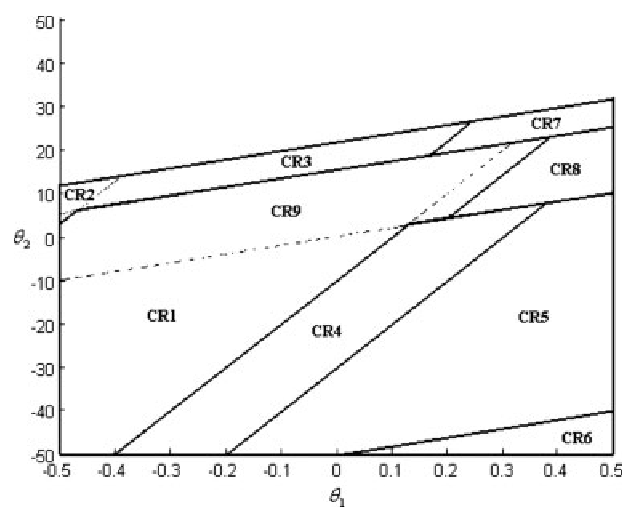

Figure 8. Critical region of Example 1 with price and demand uncertainty.

Table 8. Demand, Price and Processing Time Uncertainty for Example 1

Parameter	Value	Variation Range
Price of S4	$1 + \theta_1$	$-0.5 \le \theta_1 \le 0.5$
Demand of S4	$50 - 20\theta_1$	
Mixing time	$4.5 + \theta_2$	$-0.5 \le \theta_2 \le 0.5$

is the critical region that covers the point corresponding to given uncertain parameter values, what is the current optimal objective and current optimal solution (schedule)?

The framework of determining one critical region is shown in Figure 1 and involves the following steps:

Step 1. Select a nominal parameter value θ^0 to be studied and solve the deterministic MILP problem by fixing the uncertain parameters at θ^0 to get an integer solution.

Step 2. Formulate an mpLP problem by fixing the integer variables at the values obtained from Step 1. Solve the parametric mpLP problem. Note that a complete mpLP solution is not necessary and only the critical region that covers θ^0 is needed, which is denoted as CR^0, the corresponding optimal objective function is z^0. The parametric solution is obtained using the optimality conditions (21)–(23).

Step 3. Formulate the following MILP/MINLP problem, which aims at seeking an integer solution with a better objective function in the current critical region:

$$\begin{aligned} \max \quad & err = z^0 - (c + \theta^T D)x \\ \text{s.t.} \quad & (A + F\theta)x \ge b + E\theta \\ & x \ge 0 \\ & \sum_{i \in I^m} x_i^m - \sum_{i \in T^m} x_i^m \le |I^m| - 1, \quad m = 1, \ldots, M,\ x_i \text{ integer} \\ & err \le \varepsilon \\ & \theta \in CR^0 \end{aligned} \tag{P6}$$

where $I^m = \{i | x_i^m = 1\}$, $T^m = \{i | x_i^m = 0\}$, $|I^m|$ is the number of elements in I^m, M is the number of iterations, CR^0 is the current critical region that covers θ^0. Problem (P6) includes all the original constraints of (P2) and following additional constraints: (i) integer cuts to exclude those integer solutions that have been studied; (ii) a parametric cut to seek the integer solution that provides a better objective function, (iii) a restriction of the solution space to current critical region CR^0. The above formulation doesn't aim to get the global optimal solution that generate the biggest difference, since we want to stop the optimization process of seeking better solution once we get a better solution which generate smaller objective value, so we adopt a goal programming formulation which use the small positive parameter ε. So, in the parametric cut: $err \le \varepsilon$, ε is a small positive number to ensure that only a better but not the best objective is found. In this article, the ε value is set as 1 for all the examples which is small compared with the general profit objective. If (P6) is infeasible or $err^* \le 0$, the solution procedure stops, otherwise, store the integer solution and parameter solution θ^*, and go to Step 4.

Step 4. Formulate a new mpLP problem by fixing the integer variables at the solution of problem (P6). Solve the mpLP to get the optimal objective function z^* and the critical region CR^* that covers θ^*.

Step 5. Update the critical region that covers θ^0, then go to Step 3. This step involves the comparison of the objectives in the critical regions. By comparing the two objective functions in the intersection of the two critical regions: $CR^0 \cap CR^*$, we can get the region CR^{EX} where the new objective z^* is better. This can be achieved by solving the following redundancy test problem:

$$\begin{aligned} \max \quad & err = z^* - z^0 \\ \text{s.t} \quad & \theta \in CR^0 \cap CR^* \end{aligned} \tag{P7}$$

If $err^* \le 0$, it means that $z^* \le z^0$ is redundant in $CR^0 \cap CR^*$ (z^* is always better for a minimize problem) and $CR^{EX} = CR^0 \cap CR^*$, whereas if $err^* > 0$, it means that $z^* \le z^0$

Table 9. Solution of Example 1 with Demand, Price and Processing Time Uncertainty

Schedule (Appendix D)	Parametric Objective	Critical Region ($-0.5 \le \theta_i \le 0.5$, $i = 1,2$)
D.1	$-88.55 - 44\theta_1 + 25.16\theta_2 + 20\theta_1^2$	$CR_1 = \begin{cases} \theta_1 \le -0.3 \\ -2.63\theta_1 + 6.06\theta_2 \le -1 \end{cases}$
D.2	$-87.66 - 46.33\theta_1 + 30.52\theta_2 + 20\theta_1^2$	$CR_2 = \begin{cases} -2.63\theta_1 + 6.06\theta_2 \le -1 \\ -1.533\theta_1 + \theta_2 \le 1.17 \\ \theta_1 \le -0.184 \end{cases}$
D.3	$-86.27 - 42.4\theta_1 + 38.99\theta_2 + 46.08\theta_1\theta_2$	$CR_3 = \begin{cases} -2.63\theta_1 + 6.06\theta_2 \le -1 \\ -0.3 \le \theta_1 \le -0.184 \end{cases}$
D.4	$-90.46 - 65.22\theta_1 + 32.92\theta_2 + 13.04\theta_1\theta_2$	$CR_4 = \begin{cases} \theta_1 \ge -0.184 \\ -15.22\theta_1 + 16.46\theta_2 + 13.04\theta_1\theta_2 \le 6.2 \\ -15.22\theta_1 + 7.75\theta_2 + 13.04\theta_1\theta_2 \le 1.9 \end{cases}$
D.5	$-88.55 - 50\theta_1 + 25.16\theta_2$	$CR_5 = \{-0.184 \le \theta_1 \le -0.01$ $CR_6 = \begin{cases} \theta_1 \ge 0 \\ 15.22\theta_1 - 7.75\theta_2 - 13.04\theta_1\theta_2 \le -1.9 \end{cases}$
D.6	$-72.1 - 50.58\theta_1 + 25.16\theta_2 + 20\theta_1^2$	$CR_7 = \begin{cases} -\theta_1 + 1.223\theta_2 \le 1.074 \\ 1.533\theta_1 - \theta_2 \le -1.17 \end{cases}$

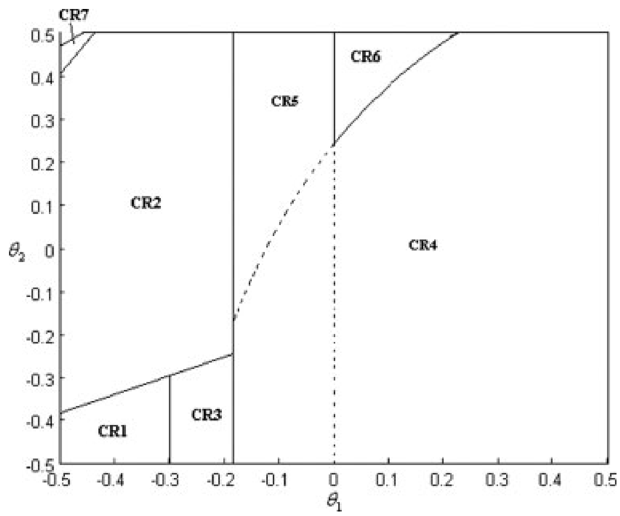

Figure 9. Critical region of Example 1 with demand, price and processing time uncertainty.

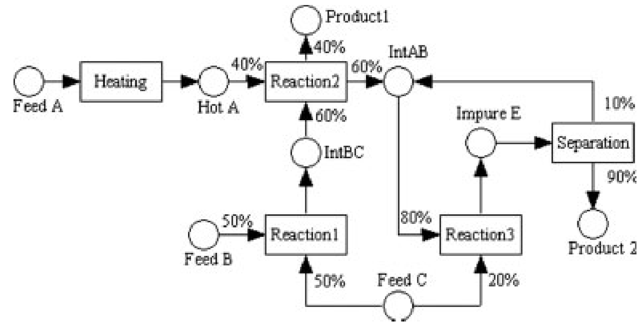

Figure 10. STN of Example 2.

is not redundant and $CR^{EX} = CR^0 \cap CR^* \cap (z^* \leq z^0)$. It should be noticed that although this redundancy test formulation is the same as the one proposed by Acevedo and Pistikopoulos,[28] nonlinear optimization rather than linear programming is needed here because the objective function might be quadratic function.

Then the region CR^{EX} is excluded from the original critical region CR^0 that covers θ^0, the exclusion is achieved by testing and selecting one constraint in CR^{EX} that θ^0 violates and adding a corresponding constraint into CR^0. To illustrate this procedure, if the selected constraint in CR^{EX} has this form: $\sum_{i,j} a_{ij}\theta_i\theta_j + \sum_i b_i\theta_i + c \leq 0$, the constraint added into CR^0 is the following: $-\sum_{i,j} a_{ij}\theta_i\theta_j - \sum_i b_i\theta_i - c \leq 0$. This process is illustrated in Figure 2, where the shadowed area representing CR^{EX} is excluded from CR^0.

Note that the original objective function z^0 should be better than z^* at θ^0 because $z^0(\theta^0)$ is the optimal objective of the MILP problem at Step 1. However since the MILP problem in Step 1 might not be solved to optimality for large scale problem (especially in scheduling applications), the objective function and integer solution of the final critical region that covers θ^0 can be updated if CR^* covers θ^0 and better objective at θ^0 is found.

There are several issues that need to be discussed regarding the proposed framework.

First, problem (P6) is a MILP/MINLP problem, which is solved to find a solution that has better objective function value at one point in the current critical region. The parametric cut $err \leq \varepsilon$ is used to stop the optimization when a better but not the best objective is found, so the computational effort is decreased which will be illustrated in Example 2. Only at the final iteration, a global optimal solution is needed to prove that no better objective exists. Thus the algorithm is more efficient in the critical region updating process comparing with the existing method which requires global optimal solution in every critical region updating step.[30] As the size of the problem becomes big for large scale scheduling problem, a resource limit has to be set for problem (P6) because it is difficult to solve to optimality for large MILP/MINLP problem, this is brought by the NP-hard basic of the problem.

Second, because the solution θ^* of (P6) might be at the boundaries of CR^0, a boundary contraction operation is used to avoid repeating the same exclusion. When θ^* is on the boundary of CR^0, the intersection $CR^0 \cap CR^*$ might only be the boundary, thus the exclusion operation will not actually update the critical region CR^0 and the new solution θ^* of (P6) will be still on this boundary. So, to avoid this case, every time that the new constraint is added, it is slightly perturbed to ensure that point θ^* is excluded (Figure 3), which can be achieved by slightly changing the constant term of the constraints. For example, a linear constraint introduced can be formulated as following:

$$\sum_i b_i\theta_i - c + \sigma \leq 0, \quad \sigma = 10^{-3}\frac{\sum_i b_i\theta_i^0 + c}{\sqrt{\sum_i b_i^2}} \tag{25}$$

Table 10. Data for Example 2

Unit	Capacity	Suitability	Processing Time	State	Storage Capacity	Initial Amount	Price
Heater	100	Heating	10	Feed A	Unlimited	Unlimited	0.0
Reactor 1	50	Reaction 1,2,3	2.0, 2.0, 1.0	Feed B	Unlimited	Unlimited	0.0
Reactor 2	80	Reaction 1,2,3	2.0, 2.0, 1.0	Feed C	Unlimited	Unlimited	0.0
Still	200	Separation	1 for product 2, 2 for IntAB	Hot A	100	0.0	0.0
				IntAB	200	0.0	0.0
				IntBC	150	0.0	0.0
				Impure	200	0.0	0.0
				Product 1	Unlimited	0.0	10.0
				Product 2	Unlimited	0.0	10.0

Table 11. Demand Uncertainty for Example 2

Parameter	Value	Variation Range
Demand of product 1	θ_1	$50 \leq \theta_1 \leq 200$
Demand of product 2	θ_2	$50 \leq \theta_2 \leq 250$

where σ is a small positive number determined to move the constraint just 0.1% of the distance between point θ^0 and the constraint border. Thus it ensures that the constraint does not move too much and also θ^0 does not violate this new constraint. This change on the constraint is only used to avoid repeating the same exclusion, and is corrected back to the original value after the iterative process finishes so that the correct constraints are not affected.

Third, the proposed framework only determines one critical region around one point. To get a complete map of the original problem, the rest of the parameter space should be explored. In the current framework, an evenly distributed testing method is used because it is difficult to find an exact geometric method to cover all the uncertain space due to of the nonconvex nature of the critical regions. This involves the generation of a number of evenly distributed points to cover the uncertainty space which are then tested to determine whether they are already covered by the identified critical regions, if not, the proposed procedure is used. This method is effective since it only requires function value evaluations. Moreover, in reality it is often not necessary to get a complete map of the parametric information. The number of points selected is based on the dimension of the uncertain space and the range: a higher dimension space and a relative bigger range needs more points to be tested so that the space is covered as much as possible.

Finally, in the solution of the problem, the resulted critical regions might have overlapped area because they belong to a larger nonconvex critical region as stated in Problem Formulation section. In the next section three examples are solved to illustrate the steps of the proposed procedure.

Examples

The proposed method has been implemented in GAMS and MATLAB, where MATLAB is used to formulate the standard form of problem based on GAMS file, to control the flow and to calculate the optimality conditions. GAMS is used to solve MILP/MINLP/NLP problems, where CPLEX 10.1 is used to solve MILP, BARON is used to solve NLP in the comparison of Step 5 and SBB is used to solve the MINLP problem. In this section, several scheduling problems are solved using the proposed method. These examples illustrate the characteristics and the computational complexity of the multiparametric programming for the solution of scheduling under uncertainty. Example 1 is a relatively smaller problem, which is used to illustrate the different uncertainty cases. Example 2 is used to illustrate the capability to address relatively larger scale problems. Example 3 is used to illustrate the ability in addressing a number of uncertain parameters. All these examples are solved in a Pentium PC (2.8 GHz, 1G RAM) running in Windows XP operation system.

Example 1

This example process involves three processing stages, namely mixing, reaction, and separation, which are processed in 3 units respectively.[38] The state-task-network (STN) representation of this example is shown in Figure 4 and the data is shown in Table 1. Products include S3 and S4 (the purified product). For the deterministic formulation with five event points, there are 236 constraints, 45 integer variables, and 86 continuous variables. Solving the deterministic MILP problem normally requires around 0.25-CPU s.

Demand Uncertainty. Assuming only demand uncertainty for the two products gives rise to RHS uncertainty in problem (P2) and an mpMILP problem is solved. The demands are defined in Table 2, where the variation ranges of the uncertain parameters are given in a boundary form. The corresponding multiparametric programming problem (P2) is solved with testing 400 evenly distributed points using the proposed method and the solution is shown in Figure 5 and Table 3 (note that in all the examples of this article the objective is set as minimum negative profit). The total time consumed is 140.6-CPU s.

As stated in Problem Formulation section, all the parametric objectives are linear and the critical regions are formed by linear constraints in this case. Among the critical regions (Figure 5), CR3 and CR5 have an overlapping area, because they have same objective function and integer solution (schedule) and actually belong to the same larger nonconvex critical region. This is also a characteristic of the solution for mpMILP, which is different from mpLP which always has a convex critical region.

To illustrate the framework, we explain in detail the process of solving for one critical region that cover the test point $(\theta_1,\theta_2) = (40.05, -49.9)$. First, we fix the parameters'

Table 12. Solution Process for Example 2 at $(\theta_1, \theta_2) = (165,50)$

Iteration in Step 3	Time Elapsed (s)	Point (θ_1^*, θ_2^*)	Objective z^*	New Constraint
1st MILP	2.63	(50, 50)	-3495.7	$\theta_1 \geq 143.87$
2nd MILP	1.72	(144, 50)	-3570	$\theta_1 \geq 150$
3rd MILP	133.63	(150, 50)	-3482.5	$\theta_1 \geq 159.25$
4th MILP	40.16	(159, 50)	-3515	$\theta_1 \geq 162.5$
5th MILP	1.25	(163, 50)	$-14340 + 66.8\theta_1$	$\theta_1 \geq 162.6$
6th MILP	1.2	(163, 50)	$-32310 + 177.2\theta_1$	$\theta_1 \geq 162.7$
7th MILP	17.11	(163, 50)	$-13590 + 62\theta_1$	$\theta_1 \geq 163.06$
8th MILP	>5000		$err^* \leq 0$, no better point exists, end	

Table 13. Solution Process for Example 2 at $(\theta_1, \theta_2) = (50,50)$

Iteration in Step 3	Time Elapsed (s)	Point (θ_1^*, θ_2^*)	Objective z^*	Operation
1st MILP	167.88	(50, 50)	−3703.2	z^0 is updated
2nd MILP	83.98	(50, 50)	−3720.16	z^0 is updated
3rd MILP	343.45	(50, 50)	−3726.26	z^0 is updated
4th MILP	1.42	(50, 50)	−3737.1	z^0 is updated
5th MILP	>5000		$err^* \leq 0$, no better point exists, end	

value at (40.05, −49.9), after solving the corresponding MILP problem of the scheduling formulation, we get an integer solution, which is then fixed and a LP problem is solved. Then mpLP problem is solved around the point (40.05, −49.9), and the resulted critical region for the mpLP problem is as Figure 6a, and the corresponding optimal objective function is $-96.14 + 0.158\theta_1$. Following the steps in our framework, in Step 3 a MILP is solved to seek any integer solution and parameter value in current critical region that can generate better objective. The solution of Step 3 (with the parameter value of $\varepsilon = 1$) is that at $(\theta_1, \theta_2) = (50, -50)$, an integer solution exists with better objective function −88.55. Then this integer solution is fixed, and a resulted mpLP problem is solved again. The resulted critical region is shown in Figure 6b. In the next step, a comparison is made to update the critical region that contains the point $(\theta_1, \theta_2) = (40.05, -49.9)$. The comparison result is that the intersection region has to be divided into two parts with additional objective comparison constraint: $-96.14 + 0.158\theta_1 \geq -88.55$, that is $\theta_1 \geq 48.2$. So the part of the original critical region that should be excluded is the right hand side part of Figure 6c, then one constraint is selected form the constraints of this excluded region to ensure that (40.05, −49.9) does not satisfy the constraint. Only $\theta_1 \geq 48.2$ satisfy this requirement, so it is selected, and correspondingly, $\theta_1 \leq 48.2 + \sigma$ is the new constraint of the updated critical region, where σ = 0.052 is used to exclude the points on this boundary. Following the framework, the next step is to seek for a better integer solution in the updated region, but an optimal objective of the MILP problem in Step 3 is 0.00, which means that there is no better objective further. Thus the procedure is ended. The final critical region that cover the point $(\theta_1, \theta_2) = (40.05, -49.9)$ is as Figure 6d, and the optimal objective function is $-96.14 + 0.158\theta_1$.

Table 14. Parametric Solution of Example 2

Schedule (Appendix E)	Parametric Objective	Critical Region
E.1	−3737.1	$CR_1 = \begin{cases} 50 \leq \theta_1 \leq 147.3 \\ 50 \leq \theta_2 \leq 226.4 \end{cases}$
E.2	−3693.3	$CR_2 = \begin{cases} 148.2 \leq \theta_1 \leq 153.3 \\ 50 \leq \theta_2 \leq 216 \end{cases}$
E.3	−3593	$CR_3 = \begin{cases} 154.12 \leq \theta_1 \leq 157.03 \\ 50 \leq \theta_2 \leq 201 \end{cases}$
E.4	−3515	$CR_4 = \begin{cases} 158.15 \leq \theta_1 \leq 162.5 \\ 50 \leq \theta_2 \leq 189 \end{cases}$
E.5	−3480	$CR_5 = \begin{cases} 163.06 \leq \theta_1 \leq 168 \\ 50 \leq \theta_2 \leq 180 \end{cases}$
E.6	−3173.3	$CR_6 = \begin{cases} 168.01 \leq \theta_1 \leq 173.3 \\ 50 \leq \theta_2 \leq 144 \end{cases}$
E.7	−2700	$CR_7 = \begin{cases} 173.9 \leq \theta_1 \leq 180 \\ 50 \leq \theta_2 \leq 90 \end{cases}$

Price Uncertainty. In this case, we consider the price uncertainty as shown in Table 4. Thus the uncertainty is included in the objective function and a specific mpMIQP problem is solved. The solution of the multiparametric programming problem is shown in Table 5 and Figure 7, where the solution still has linear objectives and critical regions formed by linear inequalities. This verified the conclusion in Problem Formulation section. The number of testing points used is 400. Total time consumed is 114-CPU s.

Demand and Price Uncertainty. In this case both demand and price uncertainties are considered as shown in Table 6. Thus the corresponding multiparametric programming problem contains both RHS and objective uncertainties and is also mpMIQP problem. This problem is solved within 250.86-CPU s. As analyzed in Problem Formulation section, the optimal objective functions as shown in Table 7 contain quadratic function of uncertain parameters. Note that the critical regions (Figure 8) don't involve quadratic constraints here because all of the objective comparison constraints are proved to be redundant in the redundancy test in Step 5, so they are not involved in the constraints. Still, we can see that overlapped critical regions exist and they form larger nonconvex regions (e.g., CR1 and CR9, CR2 and CR3). The

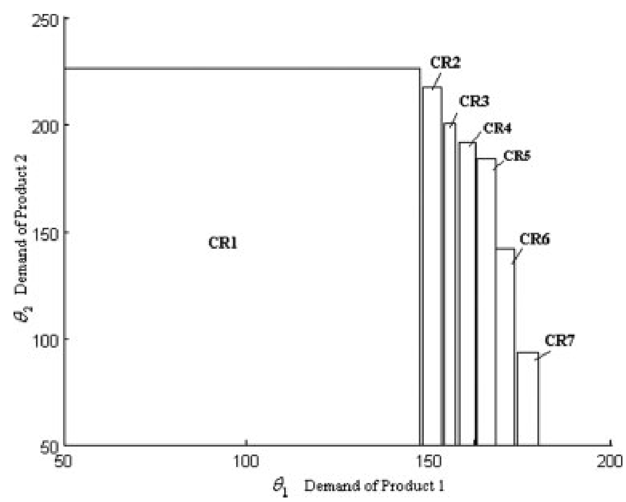

Figure 11. Critical region of Example 2.

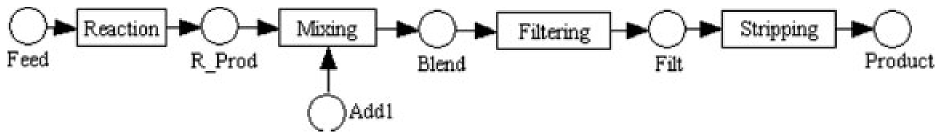

Figure 12. STN of Example 3.

number of testing points used in this case is 600 and time consumed is 250.9-CPU s.

Demand, Price, and Processing Time Uncertainty. In this case, three different uncertain parameters are involved as shown in Table 8. The uncertain mixing time parameter are formulated using new duration constraints (24) so that it is transformed into RHS uncertainty, thus an mpMIQP problem is formulated and solved here. Total 400 testing points is used and the elapsed time is 323.7-CPU s.

As shown in the solution (Table 9, Figure 9), the critical region and parametric objective function both involve quadratic terms. This also verifies the conclusion in Problem Formulation section and proves the effectiveness of the proposed method in solving mpMILP and mpMIQP problems. In all the different uncertainty cases addressed for Example 1, the computation time is no more than 300-CPU s, where 40 evenly distributed points are generated in every dimension of the parameter space, which is proved to be enough to cover all the critical regions. During the solution process, once a point is checked and is found to be covered by any critical region, it is fathomed. So although 1600 points are tested, only limited points are solved to find critical region, e.g., only 5, 4, 9, 7 points are solved for the above four cases, respectively.

The next example illustrates the computational complexity for relatively larger scale problem.

Example 2

This example involves the production of two products using three raw materials. The STN representation of this example is shown in Figure 10 and the data is shown in Table 10.[40] When 10 event points are used, the deterministic MILP formulation (1)–(14) include 1316 constraints, 200 integer variables, and 421 continuous variables. In this example, we study the effect of demand uncertainty on scheduling, thus an mpMILP problem is solved. The uncertain parameters are described in Table 11.

Compared with Example 1, the size of this problem is about five times bigger. Correspondingly, the computational time for the original MILP problem increases. For example, if we solve the parametric solution around point $(\theta_1,\theta_2) = (165,50)$, a MILP problem is first solved by fixing the parameter values at the nominal values. Using GAMS/CPLEX 10.1 the solution of this problem requires 43.1-CPU s to find an integer solution with relative optimality gap of 0.1. Next, the integer solution of the MILP is fixed and an mpLP problem is formulated. Using the optimality conditions, the following parametric information is achieved:

$$z^0 = -3480,\ CR^0 = \begin{cases} 50 \le \theta_1 \le 168 \\ 50 \le \theta_2 \le 180 \end{cases}$$

In the next iterations, because only RHS uncertainty is involved, additional MILP and mpLP problems need to be solved. For this point, seven additional MILP problems (P6) are solved to identify a point that has better objective and then update the critical region. From the details of the solution process shown in Table 12, we can see that the formulation (P6) is efficient in seeking better solution. The only computational intensive step is the last iteration which aims to prove that no better solution exists. Since this is very hard problem to solve, a resource limit of 5000-CPU s is placed at the last iteration. In every iteration step, the critical region is updated by introducing new constraint. Finally, the critical region covering point $(\theta_1,\theta_2) = (165,50)$ is updated as:

$$CR = \begin{cases} 163.06 \le \theta_1 \le 168 \\ 50 \le \theta_2 \le 180 \end{cases}.$$

The solution of the scheduling problem even the deterministic case is a very hard problem to solve.[41] So, when the size of the MILP problem becomes large, a relative gap is considered as a termination criterion (e.g., 0.1 as discussed in this example previously). However, to ensure that the correct parametric solution is obtained, the objective function and integer solution of the final critical region that covers θ^0 can be updated if CR^* covers θ^0 and better objective at θ^0 is found.

For example, when solving the parametric solution around $(\theta_1,\theta_2) = (50,50)$, the MILP problem in Step 1 requires 29.7-CPU s, the parametric objective in Step 2 is $z^0 = -3695.74$. In the next iterations of the solution process, the parametric objective and critical region is updated iteratively as shown in Table 13. Thus the final parametric objective is −3737.1 and not −3695.74 as originally determined. Of course, if we can solve the MILP in Step 1 to optimality, we can get the parametric objective −3737.1 in Step 2 and no objective updating operation will be required which can be achieved in smaller problems such as Example 1.

Table 15. Data for Example 3

Unit	Capacity	Suitability	Processing Time	State	Storage Capacity	Initial Amount	Price
Mixer 1	20	Mixing	4.0	Feed	Unlimited	Unlimited	0.0
Mixer 2	20	Mixing	4.0	Add 1	Unlimited	Unlimited	0.0
Reactor	20	Reaction	26.0	R-prod	100	0.0	0.0
Filter	20	Filtering	6.0	Blend	100	0.0	0.0
Strip tank 1	20	Stripping	8.0	Filt	100	0.0	0.0
Strip tank 2	20	Stripping	8.0	Prod	Unlimited	0.0	2.0

Table 16. Uncertainty Description of Example 3

Parameter	Value	Variation Range
Reaction time	$T_1 = 26 + \theta_1$	$-5 \le \theta_1 \le 5$
Mixing time	$T_2 = 4 + \theta_2$	$-1 \le \theta_2 \le 1$
Filtering time	$T_3 = 6 + \theta_3$	$-2 \le \theta_3 \le 2$
Stripping time	$T_4 = 8 + \theta_4$	$-3 \le \theta_4 \le 3$
Demand of product	θ_5	$0 \le \theta_5 \le 100$

The final parametric solution of this problem is shown in Table 14 and the critical regions are shown in Figure 11. It can be noticed that although additional demand of Product 1 is satisfied the profit is decreased. This happens because to satisfy the additional demand of Product 1, the production of Product 2 is reduced leading to reduction of the total profit. When the demand of Product 1 exceed 180, the problem becomes infeasible which means that within the time horizon of 16 h the capacity of the plant cannot meet this demand.

Example 3

This example involves the production of one product through four processing tasks: reaction, mixing, filtering, and stripping.[38] Figure 12 illustrates the STN representation of this plant network and Table 15 gives the data. Considering a scheduling horizon of 76 h, the deterministic MILP formulation with seven event points has 691 constraints, 168 integer variables, 311 continuous variables.

We consider five uncertain parameters in this example, which include all the processing times and the product demand as defined in Table 16. Using the same method as for Example 1 to address the duration constraints, processing time uncertainties are modeled as coefficients of binary variables, and they are transformed into RHS uncertainty through linearization before solving the mpMILP using the proposed framework.

After generating a number of evenly distributed points in the parameter space, a series of mpMILP problems are solved around every point which is not covered by any identified critical region. Part of the final parametric solution is shown in Table 17. Because of the higher dimension of the parameter space, the final solution involves a lot of critical regions as shown. CR1, CR2, CR4, and CR5 actually belong to the same larger critical region. Not all the critical regions are shown in this table to simplify the presentation of the results.

As shown in Table 18 that illustrates the computational requirements of the solution procedure, the computational

Table 17. Parametric Solution of Example 3

Schedule (Appendix F)	Objective	Critical Region (In the given range)
F.1	-80	$CR_1 = \begin{cases} \theta_5 \le 40 \\ -2\theta_1 - \theta_2 - 3.33\theta_3 + 1.33\theta_4 \le 14.22 \\ 2\theta_1 + \theta_2 + 1.43\theta_3 + 1.29\theta_4 \le -16.76 \end{cases}$
		$CR_2 = \begin{cases} \theta_5 \le 40 \\ -4\theta_1 - 2\theta_2 - \theta_3 - 3\theta_4 \le 32 \\ 2\theta_1 + \theta_2 + \theta_3 + \theta_4 \le -14.67 \\ 3\theta_1 + \theta_2 + \theta_3 \le -17.33 \end{cases}$
		$CR_4 = \begin{cases} \theta_5 \le 40 \\ -2\theta_1 - \theta_2 - 1.82\theta_3 - \theta_4 \le 16.24 \\ 2\theta_1 + \theta_2 + 1.71\theta_3 + 1.14\theta_4 \le -16.38 \end{cases}$
		$CR_5 = \begin{cases} \theta_5 \le 40 \\ 2\theta_1 + \theta_2 + \theta_3 + \theta_4 \le -13.33 \end{cases}$
F.2	$-58 + 3\theta_1 + 1.5\theta_2 + 1.5\theta_3 + 1.5\theta_4$	$CR_3 = \begin{cases} -2\theta_1 - \theta_2 - \theta_3 - \theta_4 \le 14.67 \\ 2\theta_1 + \theta_2 + \theta_3 + \theta_4 + 1.33\theta_5 \le 38.67 \\ 4.86\theta_1 + \theta_2 + \theta_3 - 1.86\theta_4 \le -22.29 \end{cases}$
		...
F.3	$-58.95 + 3.16\theta_1 + 1.58\theta_2 + 1.58\theta_3 + 1.58\theta_4$	$CR_6 = \begin{cases} 2\theta_1 + \theta_2 + \theta_3 + \theta_4 \le -13.33 \\ -2\theta_1 - \theta_2 + 5.33\theta_3 - 7.33\theta_4 \le 30.22 \\ 2.38\theta_1 - \theta_2 - \theta_3 - \theta_4 \le -2.26 \\ 2\theta_1 + \theta_2 + 10.5\theta_3 - 8.5\theta_4 \le 24.67 \\ 2\theta_1 + \theta_2 + \theta_3 + \theta_4 + 1.27\theta_5 \le 37.33 \\ \theta_1 \le -2 \end{cases}$
		...
F.4	$-58.55 + 3.16\theta_1 + 1.58\theta_2 + 1.58\theta_3 + 0.79\theta_4$	$CR_{21} = \begin{cases} 2\theta_1 + \theta_2 + 5.25\theta_4 \le 13.63 \\ 2\theta_1 + \theta_2 + \theta_3 + \theta_4 + 1.27\theta_5 \le 37.33 \\ 2\theta_2 + \theta_4 \le 1.5 \\ \theta_2 \le 0.5 \\ \theta_1 \le 2 \end{cases}$
		...

Table 18. Solution Process for Example 3 at $\theta^0 = (-4.99, -0.998, 0.004, 2.41, 0.1)$

	Time	Result
MILP in step 1	0.33 s	$z^0 = -58.95 + 3.16\theta_1 + 1.58\theta_2 + 1.58\theta_3 + 1.58\theta_4$
1st MILP in step 3	1.66 s	$\theta^* = (-2, 0.5, -0.5, 0.5, 0)$, $z^* = -59.34 + 3.16\theta_1 + 1.58\theta_2 + 0.79\theta_4$, new constraint introduced: $\theta_1 \leq -2$
2nd MILP in step 3	0.2 s	Infeasible problem, no better point exists, end

time for obtaining the parametric solution around a given parameter value does not depend on the number of uncertain parameters, but depends on the size of the deterministic problem. For example, solving one critical region in Example 3 normally costs 3-CPU s, but solving a critical region in Example 2 costs several thousand seconds (main computational effort is in the last iteration, as stated previously).

However, to solve a complete solution of the multiparametric programming problem, the computational time increases as the number of uncertain parameters increases because more points have to be generated and tested in the parameters space to cover all the critical regions. In Example 3, when we evenly select 10 values for every parameter in their range, 100,000 points are generated and tested. The whole solution process requires 5470-CPU s. On the other hand, a higher dimension parameter space will involve more critical regions, as Example 1 and 2 have only several critical regions, but Example 3 has more than 20 critical regions, thus more subproblems need to be solved.

Conclusion

In this article, a multiparametric programming framework is proposed to solve the process scheduling problem under uncertainty. This method provides an exact and systematic way to analyze the uncertainty in process scheduling problem and the parametric information achieved can be used in several different ways: the result of multiparametric programming for the scheduling problem can be used to get schedule instantly through looking up in the final solution table without resolving the problem; and also the solution provides a basis for reactive scheduling in the sense that the decision maker can rapidly find a new schedule with the realization of uncertainty.

To increase the efficiency of attaining the exact solution map of the corresponding mpMILP and mpMIQP problem, the proposed framework uses a decomposition method which solves for the parametric information around a certain parameter value, and not seeking for a complete map of solution at one time. This method can give the decision maker useful information about uncertainty effects fast. Another advantage of the proposed methodology is that it can be easily parallelized by decomposing the original parameter space into smaller regions that can be solved in parallel thus decreasing the computational complexity of the algorithm.

The proposed method is also efficient in the critical region updating process because the MILP/MINLP problem is solved to seek just better but not best solution in a given critical region. In other words, the proposed formulation is used to seek a feasible solution but not a global optimal solution, thus the computation efficiency is increased. The only big computation effort is in the final iteration of the proposed framework which needs to prove that no better solution exists, so a global optimal solution of the MILP/MINLP is needed.

The consideration of a general form of LHS uncertainty as coefficient of continuous variables is still a challenge. Because for this case the exact parametric solution is complex, different efficient methods have to be developed to address the underlying complexity of this problem.

Finally, as shown in Example 3 and for more realistic problems involving a lot of uncertain parameters, a large number of critical regions exist. For these cases we are investigating a hybrid scheme that will reduce the number of uncertain parameters that need to be considered in a parametric framework.

Acknowledgments

The authors gratefully acknowledge financial support from the National Science Foundation under Grants CTS 0625515, and 0224745.

Notation

$i \in I$ = tasks
I_s = tasks which produce or consume state (s)
I_j = tasks which can be performed in unit (j)
$j \in J$ = units
J_i = units which are suitable for performing task (i)
$n \in N$ = event points representing the beginning of a task
$s \in S$ = states
price_s = price of state (s)
$d_{s,n}$ = amount of state (s) delivered to the market at event point (n)
$wv_{i,j,n}$ = binary, whether or not task (i) in unit (j) start at event point (n)
$st_{s,n}$ = continuous, amount of state (s) at event point (n)
$\rho^P_{s,i}$, $\rho^C_{s,i}$ = proportion of state (s) produced, consumed by task (i), respectively
$b_{i,j,n}$ = continuous, amount of material undertaking task (i) in unit (j) at event point (n)
$st_s^{\max}$ = available maximum storage capacity for state (s)
$v_{i,j}^{\min}$, $v_{i,j}^{\max}$ = minimum amount, maximum capacity of unit (j) when processing task (i)
r_s = market demand for state (s) at the end of the time horizon
$Tf_{i,j,n}$ = continuous, time at which task (i) finishes in unit (j) while it starts at event point (n)
$Ts_{i,j,n}$ = continuous, time at which task (i) starts in unit (j) at event point (n)
$\alpha_{i,j}$, $\beta_{i,j}$ = constant, variable term of processing time of task (i) in unit (j) respectively
H = time horizon

Literature Cited

1. Huercio A, Espuña A, Puigjaner L. Incorporating on-line scheduling strategies in integrated batch production control. *Comput Chem Eng*. 1995;19:S609–S615.
2. Sanmartí E, Huercio A, Espuña A, Puigjaner L. A combined scheduling/reactive scheduling strategy to minimize the effect of process operations uncertainty in batch plants. *Comput Chem Eng*. 1996; 20:1263–1268.
3. Honkomp SJ, Mockus L, Reklaitis GV. A framework for schedule evaluation with processing uncertainty. *Comput Chem Eng*. 1999;23: 595–609.
4. Vin JP, Ierapetritou MG. A new approach for efficient rescheduling of multiproduct batch plants. *Ind Eng Chem Res*. 2000;39:4228–4238.

5. Ruiz D, Cantón J, Nougués JM, Espuña A, Puigjaner L. On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants. *Comput Chem Eng*. 2001;25:829–837.
6. Méndez CA, Cerdá J. Dynamic scheduling in multiproduct batch plants. *Comput Chem Eng*. 2003;27:1247–1259.
7. Mendez CA, Cerdá J. An MILP framework for batch reactive scheduling with limited discrete resources. *Comput Chem Eng*. 2004;28:1059–1068.
8. Janak SL, Floudas CA. Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling. *Ind Eng Chem Res*. 2006;45:8253–8269.
9. Ierapetritou MG, Pistikopoulos EN. Global optimization for stochastic planning, scheduling and design problems. In: Grossmann IE, editor. *Global Optimization in Engineering Design*. Dordrecht, Netherlands: Kluwer, 1996:231–287.
10. Bonfill A, Bagajewicz M, Espuna A, Puigjaner L. Risk management in the scheduling of batch plants under uncertain market demand. *Ind Eng Chem Res*. 2004;43:741–750.
11. Bonfill A, Espuna A, Puigjaner L. Addressing robustness in scheduling batch processes with uncertain operation times. *Ind Eng Chem Res*. 2005;44:1524–1534.
12. Balasubramanian J, Grossmann IE. A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Comput Chem Eng*. 2002;26:41–57.
13. Orçun S, Altinel K, Hortaçsu Ö. Scheduling of batch processes with operational uncertainties. *Comput Chem Eng*. 1996;20:S1191–S1196.
14. Petkov SB, Maranas CD. Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty. *Ind Eng Chem Res*. 1997;36:4864–4881.
15. Vin JP, Ierapetritou MG. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Ind Eng Chem Res*. 2001;40:4543–4554.
16. Lin X, Janak SL, Floudas CA. A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. *Comput Chem Eng*. 2004;28:1069–1085.
17. Janak SL, Lin X, Floudas CA. A new robust optimization approach for scheduling under uncertainty: II. Uncertainty with known probability distribution. *Comput Chem Eng*. 2007;31:171–195.
18. Jia Z, Ierapetritou MG. Generate pareto optimal solutions of scheduling problems using normal boundary intersection technique. *Comput Chem Eng*. 2007;31:266–280.
19. Balasubramanian J, Grossmann IE. Scheduling optimization under uncertainty—an alternative approach. *Comput Chem Eng*. 2003;27:469–490.
20. Petrovic D, Duenas A. A fuzzy logic based production scheduling/rescheduling in the presence of uncertain disruptions. *Fuzzy Set Syst*. 2006;157:2273–2285.
21. Wang J. A fuzzy robust scheduling approach for product development projects. *Eur J Oper Res*. 2004;152:180–194.
22. Jia Z, Ierapetritou MG. Short-term scheduling under uncertainty using MILP sensitivity analysis. *Ind Eng Chem Res*. 2004;43:3782–3791.
23. Borrelli E, Bemporad A, Morari M. A geometric algorithm for multi-parametric linear programming. *J Optim Theory Appl*. 2003;118:515–540.
24. Seron M, De Dona JA, Goodwin GC. Global analytical model predictive control with input constraints. *39th IEEE Conference on Decision and Control*, Sydney, Australia. 2000.
25. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. *Automatica*. 2002;38:3–20.
26. Johansen TA. On multi-parametric nonlinear programming and explicit nonlinear model predictive control. *41th IEEE Conference on Decision and Control*, Las Vegas, NV. 2002.
27. Acevedo J, Salgueiro M. An efficient algorithm for convex multiparametric nonlinear programming problems. *Ind Eng Chem Res*. 2003;42:5883–5890.
28. Acevedo J, Pistikopoulos EN. A multiparametric programming approach for linear process engineering problems under uncertainty. *Ind Eng Chem Res*. 1997;36:717–728.
29. Dua V, Pistikopoulos EN. Algorithms for the solution of multiparametric mixed-integer nonlinear optimization problems. *Ind Eng Chem Res*. 1999;38:3976–3987.
30. Dua V, Bozinis NA, Pistikopoulos EN. A multiparametric programming approach for mixed-integer quadratic engineering problems. *Comput Chem Eng*. 2002;26:715–733.
31. Pistikopoulos EN, Georgiadis MC, Dua V. *Multiparametric Programming, Process Systems Engineering,* Vol. 1. Weinheim, Germany: Wiley-VCH, 2007.
32. Pistikopoulos EN, Georgiadis MC, Dua V. *Multiparametric Model-Based Control, Process Systems Engineering*, Vol. 2. Weinheim, Germany: Wiley-VCH, 2007.
33. Ryu JH. *Design and Operation of Enterprise-Wide Process Networks Under Uncertainty,* Ph.D. Thesis. London: University of London, 2003.
34. Ryu JH, Pistikopoulos EN. A novel approach to scheduling of zero-wait batch processes under processing time variations. *Comput Chem Eng*. 2007;31:101–106.
35. Ryu J, Pistikopoulos EN. Solving scheduling problems under uncertainty using parametric programming. In: Stephanopoulos G, Lee JH, Yoon ES, editors. *Proceedings of the 6th IFAC Symposium on Dynamics and Control of Process Systems*. New York: Pergamon. 2001:225–230.
36. Pistikopoulos EN, Dua V. Planning under uncertainty: a parametric programming approach. *3rd International Conference on Foundations of Computer-Aided Process Operations*, Snowbird, UT. 1998.
37. Hugo A, Pistikopoulos S. Long-range process planning under uncertainty via parametric programming. *Eur Symp Comput Aided Process Eng*. 2005;15:127–132.
38. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling: I. Multipurpose batch processes. *Ind Eng Chem Res*. 1998;37:4341–4359.
39. Al-Khayyal FA, Falk JE. Jointly constrained biconvex programming. *Math Oper Res*. 1983;8:273–286.
40. Kondili E, Pantelides CC, Sargent RWH. A general algorithm for short-term scheduling of batch operations. I. MILP Formulation. *Comput Chem Eng*. 1993;17:211–227.
41. Floudas CA, Lin X. Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Ann Oper Res*. 2005;139:131–162.

Appendix A

Schedules for Example 1 with only demand uncertainty (*x*: hours, *y*: equipment).

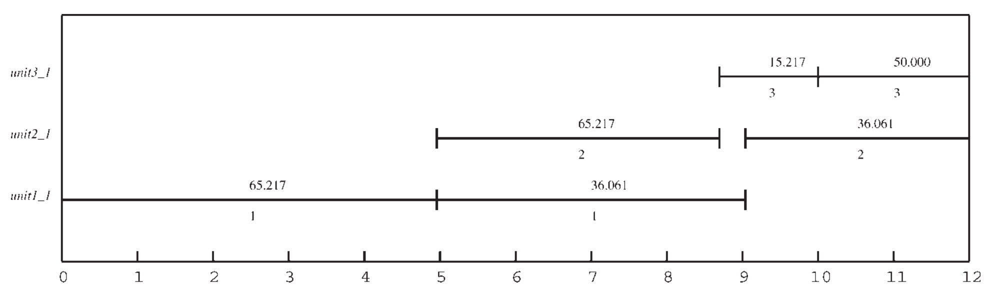

Figure A1. Schedule in CR1.

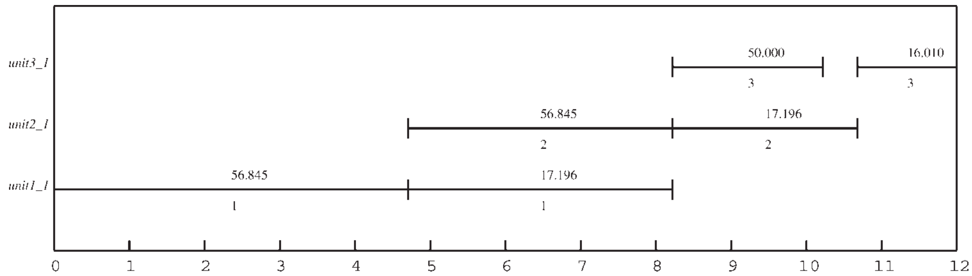

Figure A2. Schedule in CR2.

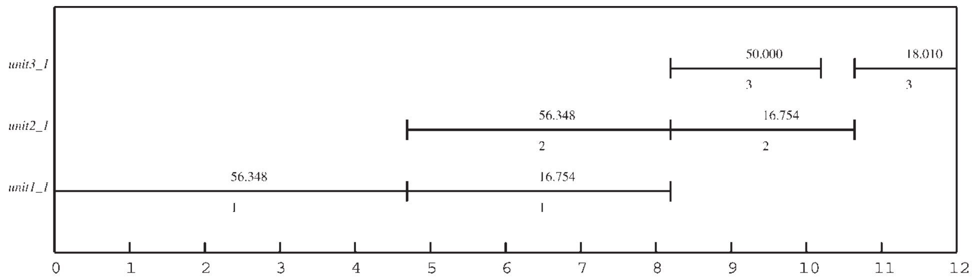

Figure A3. Schedule in CR3, CR5.

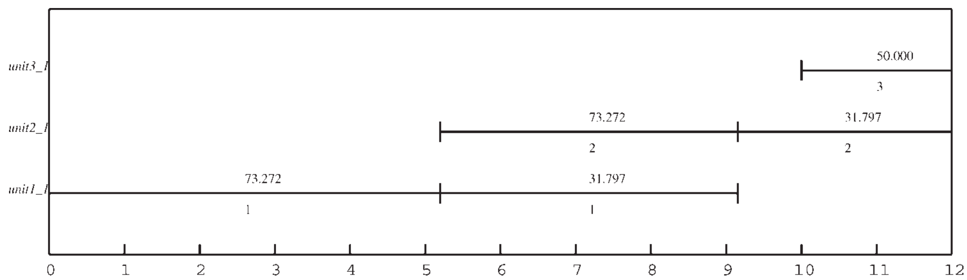

Figure A4. Schedule in CR4.

Appendix B

Schedules for Example 1 with only price uncertainty.

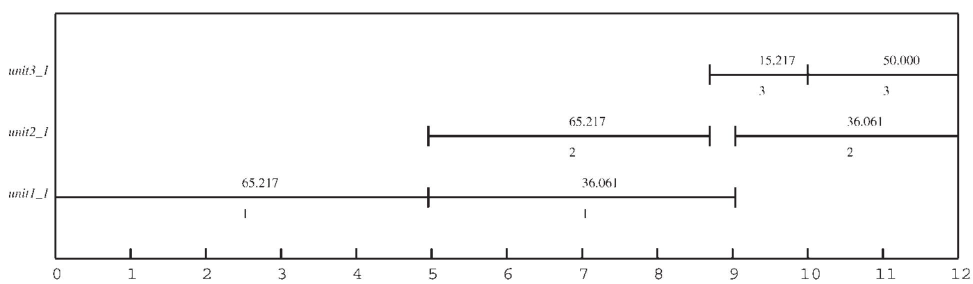

Figure B1. Schedule in CR1.

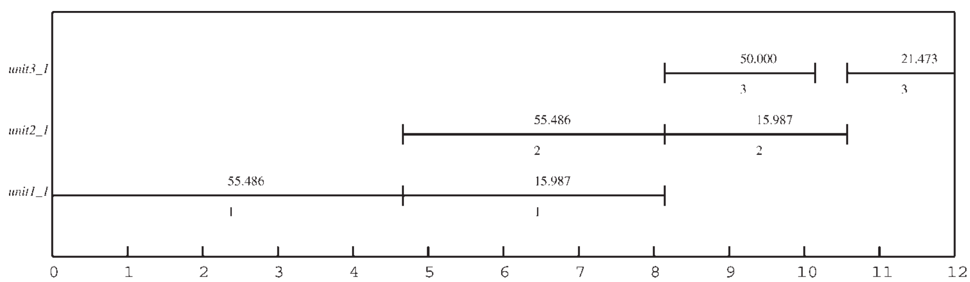

Figure B2. Schedule in CR2.

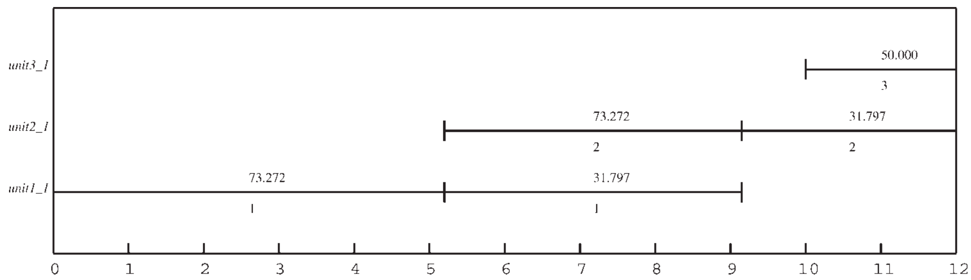

Figure B3. Schedule in CR3 and CR4.

Appendix C

Schedules for Example 1 with demand and price uncertainty.

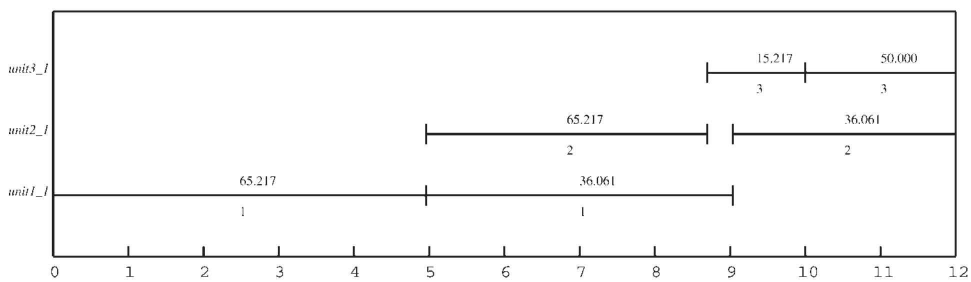

Figure C1. Schedule in CR1, CR9.

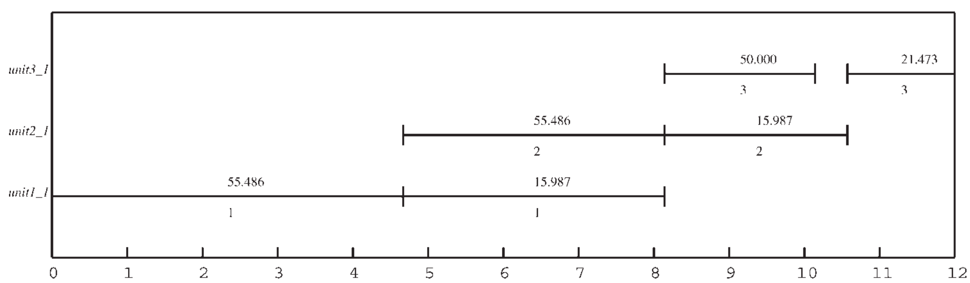

Figure C2. Schedule in CR2, CR3.

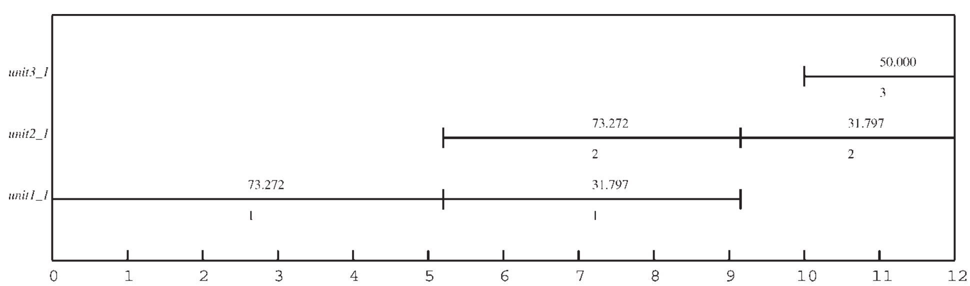

Figure C3. Schedule in CR4.

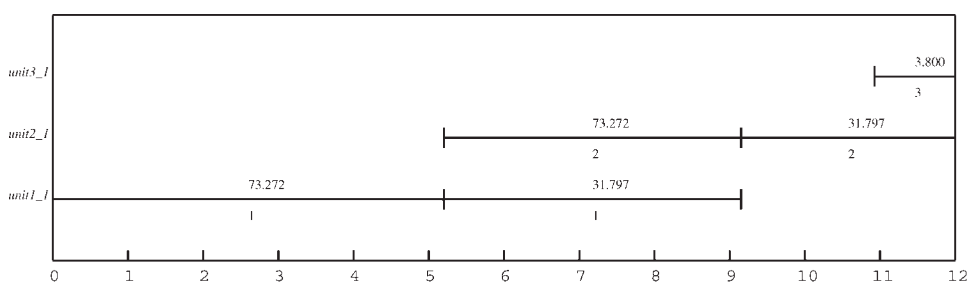

Figure C4. Schedule in CR5.

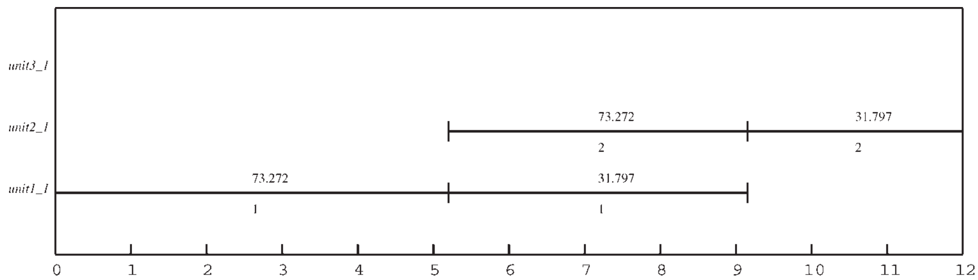

Figure C5. Schedule in CR6.

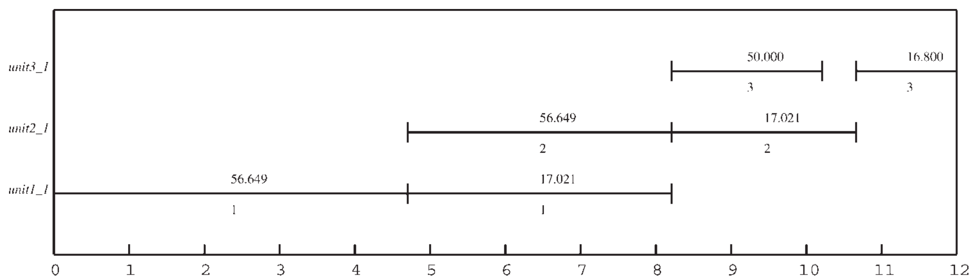

Figure C6. Schedule in CR7.

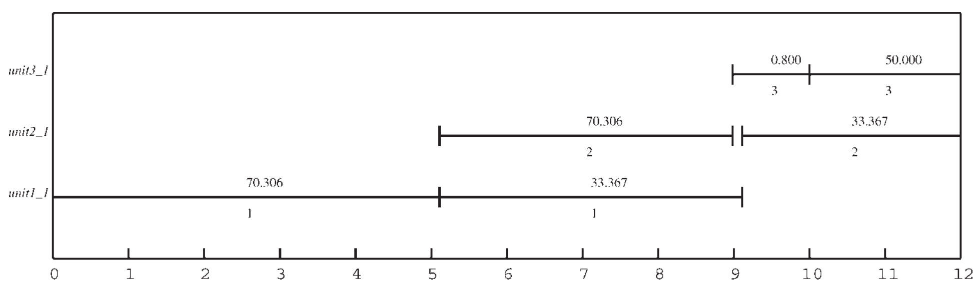

Figure C7. Schedule in CR8.

Appendix D

Schedules for Example 1 with demand, price and processing time uncertainty.

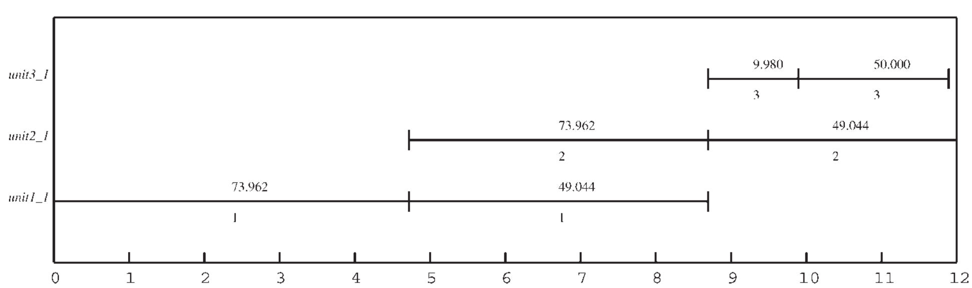

Figure D1. Schedule in CR1.

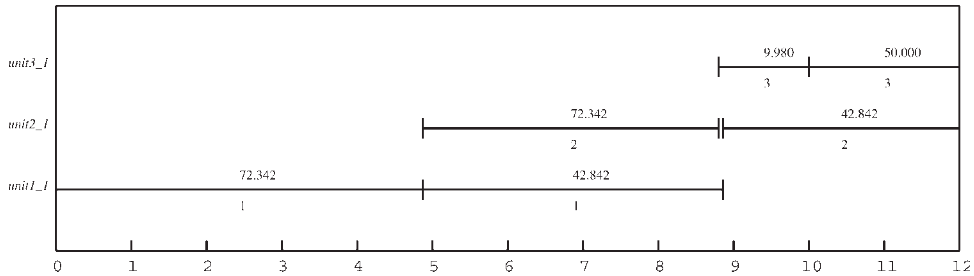

Figure D2. Schedule in CR2.

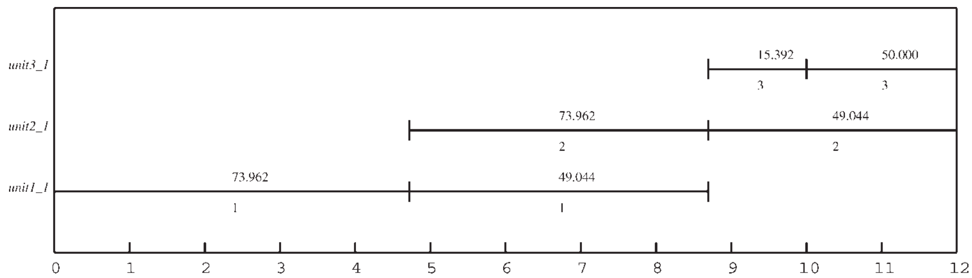

Figure D3. Schedule in CR3.

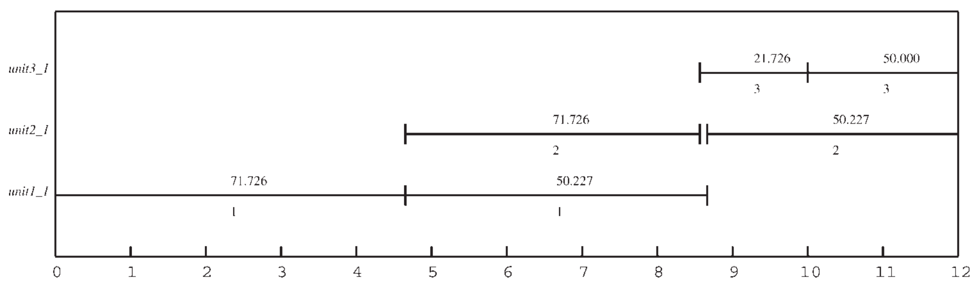

Figure D4. Schedule in CR4, CR5.

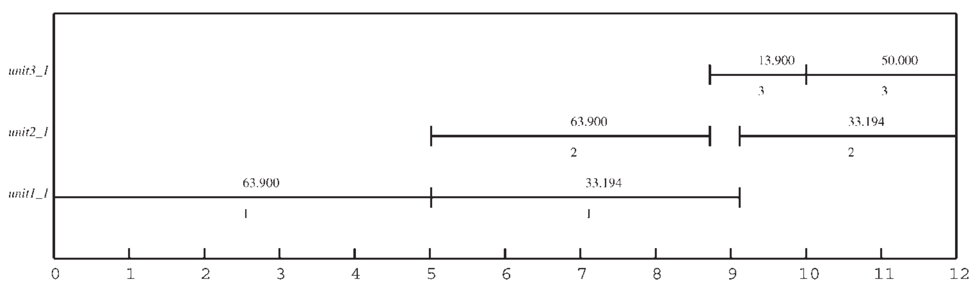

Figure D5. Schedule in CR6.

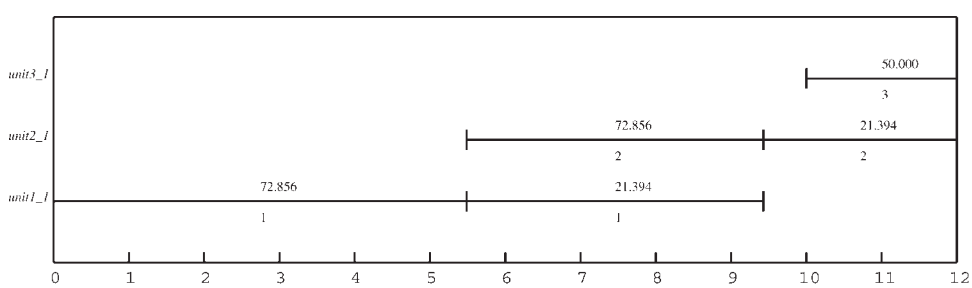

Figure D6. Schedule in CR7.

Appendix E

Schedules for Example 2.

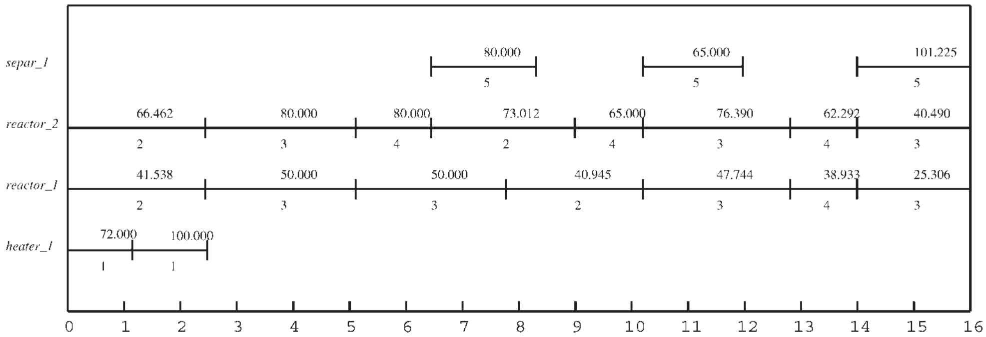

Figure E1. Schedule in CR1.

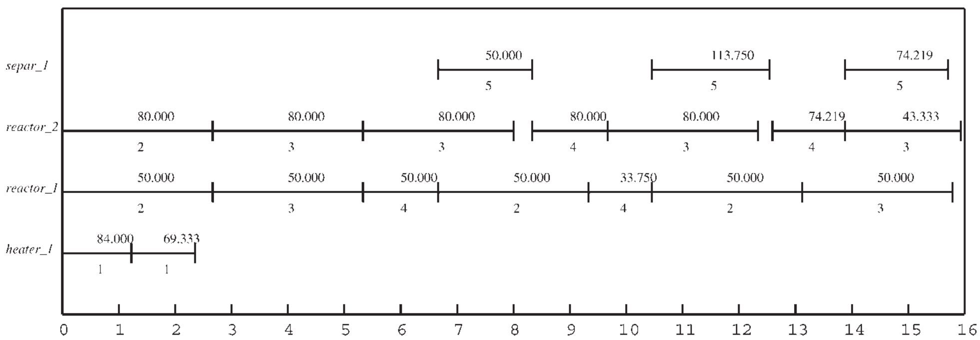

Figure E2. Schedule in CR2.

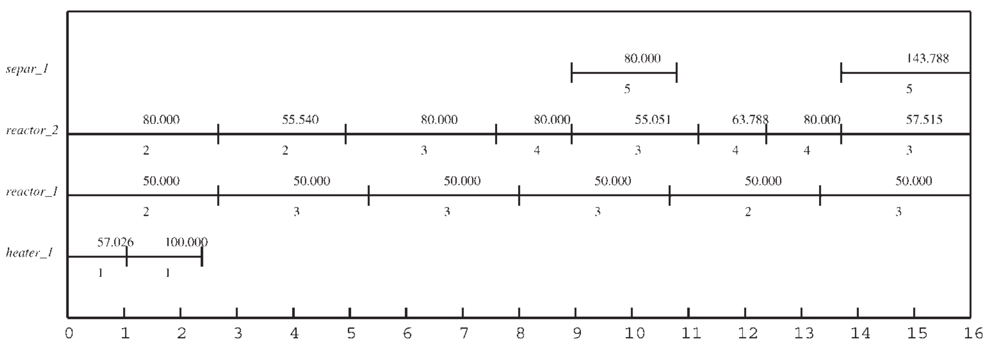

Figure E3. Schedule in CR3.

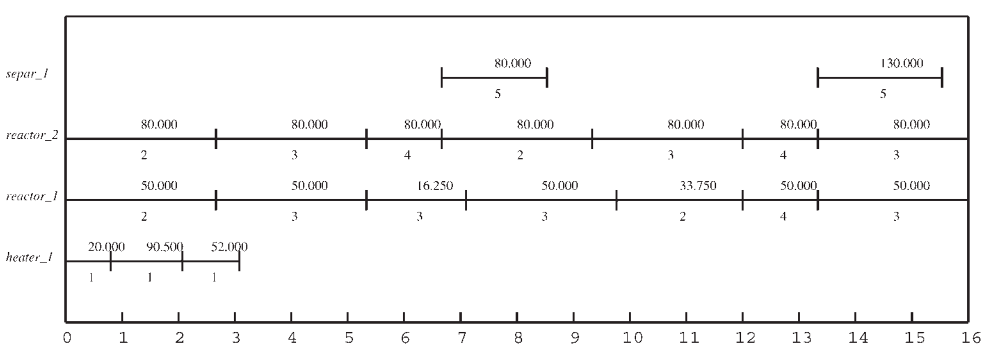

Figure E4. Schedule in CR4.

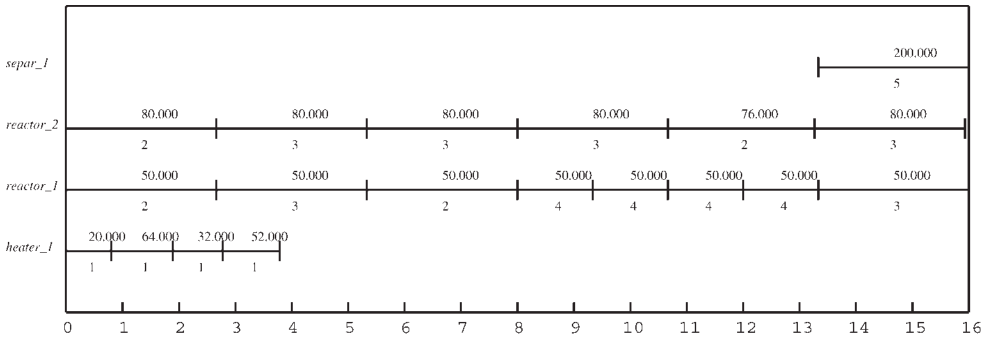

Figure E5. Schedule in CR5.

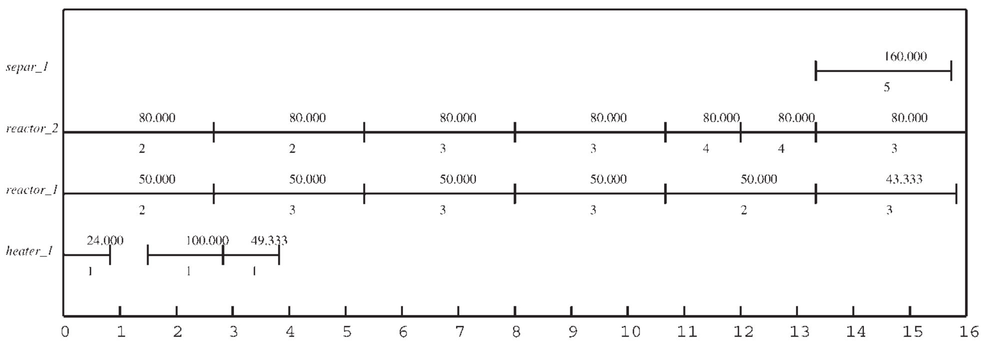

Figure E6. Schedule in CR6.

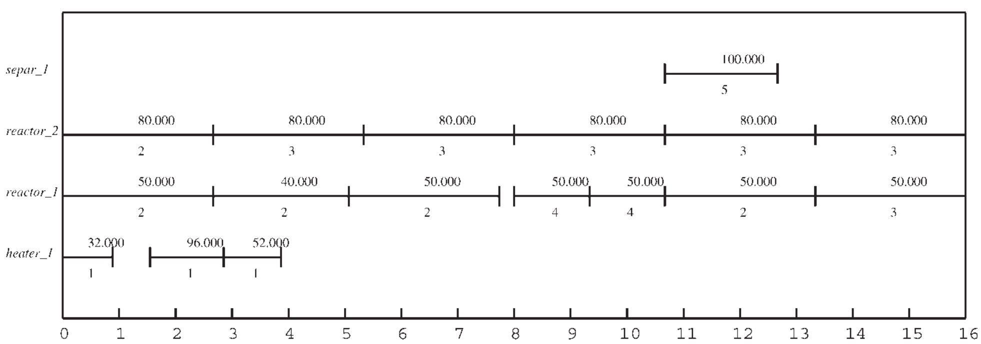

Figure E7. Schedule in CR7.

Appendix F

Schedules for Example 1 with only price uncertainty.

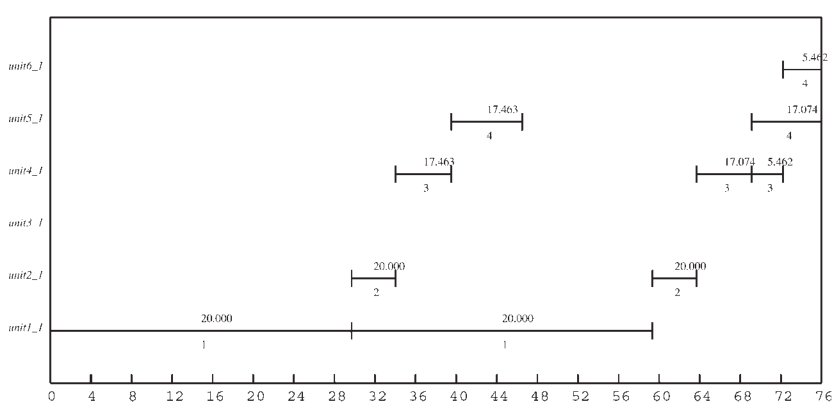

Figure F1. Schedule in CR1, CR2, CR4, CR5.

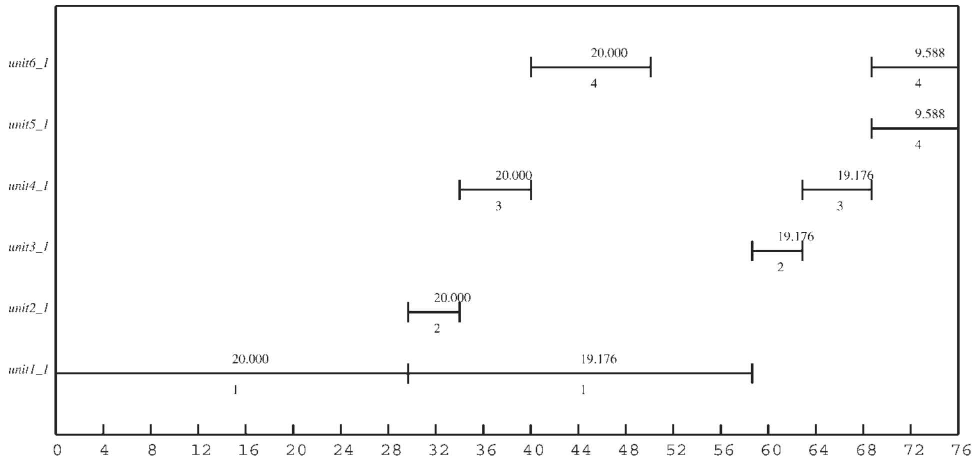

Figure F2. Schedule in CR3.

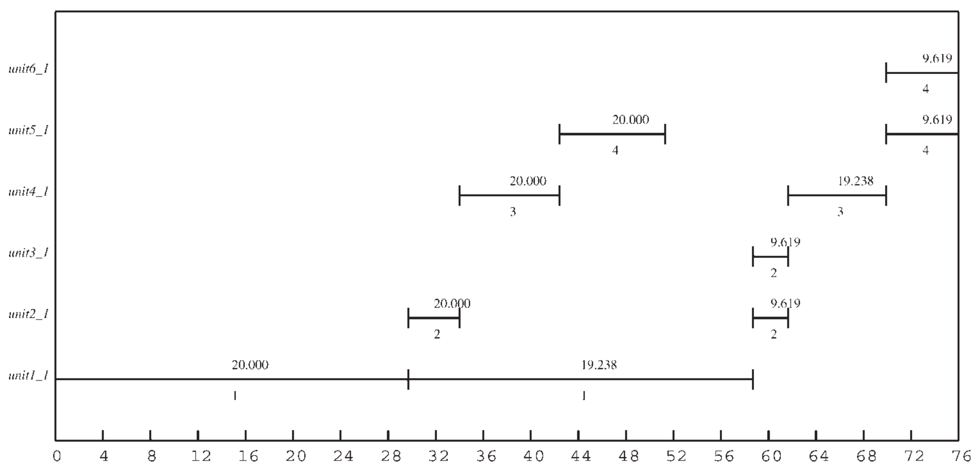

Figure F3. Schedule in CR6.

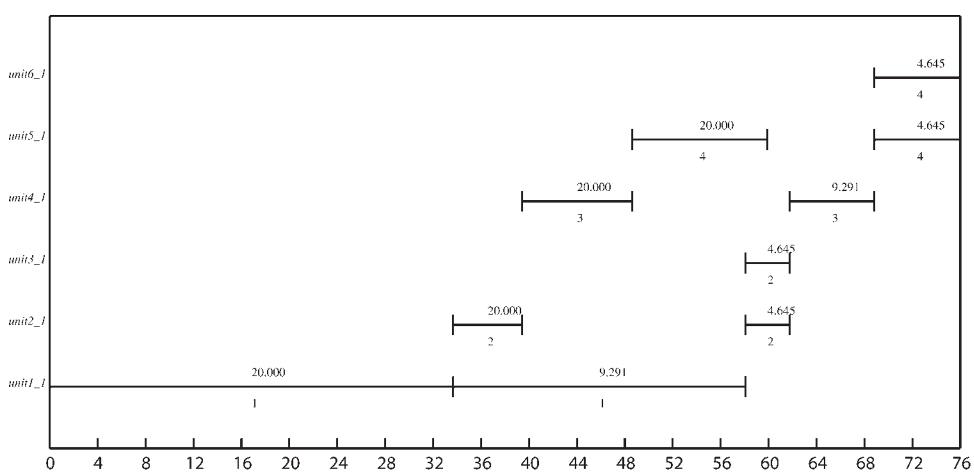

Figure F4. Schedule in CR21.

Manuscript received June 4, 2007, and revision received Sept. 19, 2007.