Cyclic Scheduling of Continuous Parallel-Process Units with Decaying Performance

Vipul Jain and Ignacio E. Grossmann
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

The problem of scheduling multiple feeds on parallel units is addressed. The performance of each unit is assumed to decrease with time, therefore requiring planned maintenance to restore its performance. A mixed-integer nonlinear programming (MINLP) model presented uses the process information regarding the exponential decay in performance with time to find a cyclic schedule for feed processing. Optimization trade-off is between the performance of the units, and the maintenance costs and the loss in production due to shutdowns. Three main decisions for the optimal schedule are: a) assignment of various feeds to different units; b) processing time of each feed on different units; and c) number of subcycles of each feed on different units. Special mathematical properties of the MINLP model are exploited to solve the problem to global optimality with a branch-and-bound method. Using several examples advantages of the model are illustrated and performances of different solution algorithms are compared.

Introduction

Motivated by a number of practical industrial applications, planning and scheduling of continuous process plants has recently received increasing attention (Pinto and Grossmann, 1995; Sahinidis and Grossmann, 1991; and Schilling et al., 1994). In the past, most continuous process plants were designed for a particular feedstock and for manufacturing a specific product. An increased need for manufacturing flexibility requires that plants be able to process different feedstocks and produce multiple products. If the demand is stable, scheduling is performed by assuming constant-demand rates over an infinite time horizon in which plants operate in a cyclic fashion. On the other hand, if the demand exhibits an irregular pattern, scheduling must be performed on a short-term basis over a specified time horizon.

Scheduling problems can in general be modeled by using either a continuous-time representation or a discrete-time representation. A review of these modeling techniques can be found in Pinto and Grossmann (1998). An extensive review of batch scheduling has been presented by Reklaitis (1992). An algorithm for cyclic scheduling of batch plants based on discrete-time representation was proposed by Shah et al. (1993). A different model based on the Resource Task Network (RTN) process representation and continuous-time representation has been proposed by Schilling and Pantelides (1997). When the problem involves an infinite time horizon and multiproduct scheduling on a single continuous production line, it is known as the economic lot scheduling problem (Elmaghraby, 1978). If the demand rates are constant, the realistic solution for this case is a finite-length repeatable schedule. A survey of related work is given in Sahinidis and Grossmann (1991).

Cyclic scheduling of continuous plants has received limited attention. Most of the work has focused on scheduling driven by trade-offs between inventory and cleanup costs. Sahinidis and Grossmann (1991) considered continuous plants with parallel lines, but with only one stage. These authors developed a solution method based on generalized Benders decomposition to solve large mixed-integer nonlinear programs (MINLPs) that arise in this problem. A key feature in their solution method was the analytical solution of the dual subproblems. Pinto and Grossmann (1995) developed a model for continuous scheduling in multistage plants and with one line per stage. Inventories between the stages were modeled using min and max functions. These nondifferentiable functions were rigorously handled with mixed-integer constraints. The resulting MINLP was solved by combining the outer approximation algorithm with generalized Benders decomposition. The objective of both these models was to find the

Correspondence concerning this article should be addressed to I. E. Grossmann.

processing times and the sequence of the products to be manufactured to maximize the profit. The models in both these articles incorporate sequence-dependent cleanup times.

This article deals with a cyclic continuous scheduling and planning problem for multiple feedstocks that require the use of process models for scheduling decisions. The process model considered is the one of exponential decay in performance, which requires the shutdown and cleaning of the units. This type of scheduling problem in ethylene plants will be discussed, as well as the case of parallel reactors with catalyst deactivation. MINLP models for these problems are developed, and their mathematical properties and a branch-and-bound algorithm to solve these models are presented. Finally, several example problems are considered to illustrate the usefulness of the proposed model and the solution method.

Motivating Example

The ethylene production process is the source of motivation for this work. Ethylene is mainly produced by steam cracking (Matar and Hatch, 1994). The feedstocks for ethylene production vary from light paraffinic hydrocarbon gases to various petroleum fractions. The nature of the feed determines the operating conditions, the yield of ethylene, and the design of the furnace used for steam cracking. Design of the furnace is determined mainly by the physical state of the feed, that is, whether the feed is liquid or gas. The operating conditions and yield of ethylene, on the other hand, vary significantly with the average molecular weight of the feed. The ethylene yield can be as high as 80% for feeds like ethane and can be as low as 25% if the feed is gas oil. In the past most of the ethylene plants used only a single feed for ethylene production. Increased demand for ethylene has forced the plants to move toward multiple feeds. This shift was highlighted in a recent article by Lee and Aitani (1990).

The cracking process leads to the formation of coke on the inner surface of the furnace tubes. The rate of coke formation is dependent on the nature of the feed, and is particularly significant for the case of liquid feeds. Coke deposits inside the tube have two main effects:

1. It acts as insulation and offers higher heat-transfer resistance. If the coil temperature is kept constant, the temperature inside the tube decreases.

2. It reduces the effective area of cross section inside the tubes, leading to reduced residence time of feed in the furnace.

To account for these effects, the furnace can be operated in two different ways:

1. The conversion (dependent on temperature and residence time) is kept constant by reducing the feed rate and increasing the coil temperature with time. This results in reduced production (lower feed rate) of ethylene and increased utility cost. If the coil temperature is increased above a certain limit, the life of the furnace also decreases.

2. The feed rate and coil temperature are kept constant, with which the conversion decreases with time. This results in lower profit (lower conversion) and an increased percentage of byproducts.

Irrespective of the operating policy, the performance of the furnace decreases over time. Therefore, the furnace has to be shut down and cleaned to restart operation at a higher performance level. The time between the cleanings must be determined by considering the trade-off between cleanup costs and the average profitability. Similar scenarios exist in reactors where the catalyst undergoes deactivation over time. This article focuses on solving cyclic scheduling problems using process models that predict decaying performance.

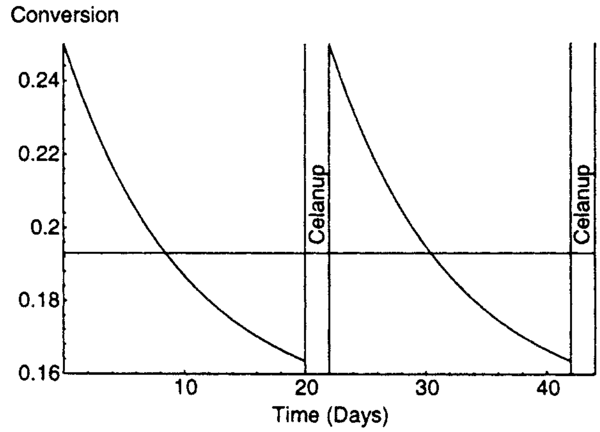

Figure 1. Variation in conversion with time (Case 1: infrequent cleaning).

First, let us consider a small quantitative example that highlights the importance of trade-offs and the need for optimization models for these problems. Consider steam cracking of a hydrocarbon fraction in a furnace. Let us assume that the feed rate and coil temperature are kept constant, as a result of which conversion to ethylene decreases over time. Also, assume that 2 days are required for the cleanup of the furnace. There are two major possible scenarios for operation: (1) frequent cleaning of furnace; and (2) infrequent cleaning of the furnace. If the conversion to ethylene decreases exponentially with time, the variation in conversion over time for these two cases is as shown in Figures 1 and 2, respectively.

In the first case (Figure 1) a cleanup operation is performed every 22 days (cycle time). If the cleanup cost is \$2,000, then the cleanup cost on a daily basis is \$2,000/22. In the second case (Figure 2) a cleanup operation is performed after every 11 days, so the cleanup cost is \$2,000/11. Clearly

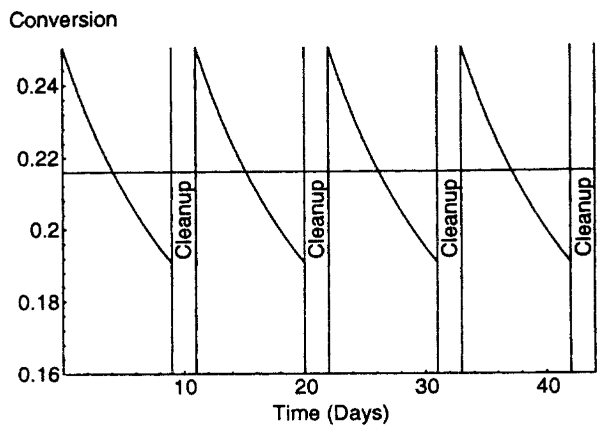

Figure 2. Variation in conversion with time (Case 2: frequent cleaning).

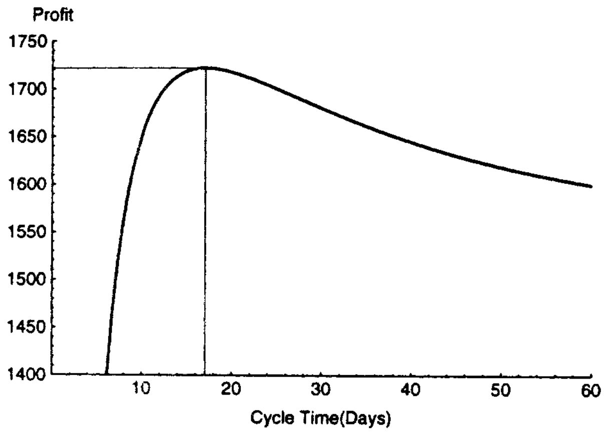

Figure 3. Profit as a function of cycle time.

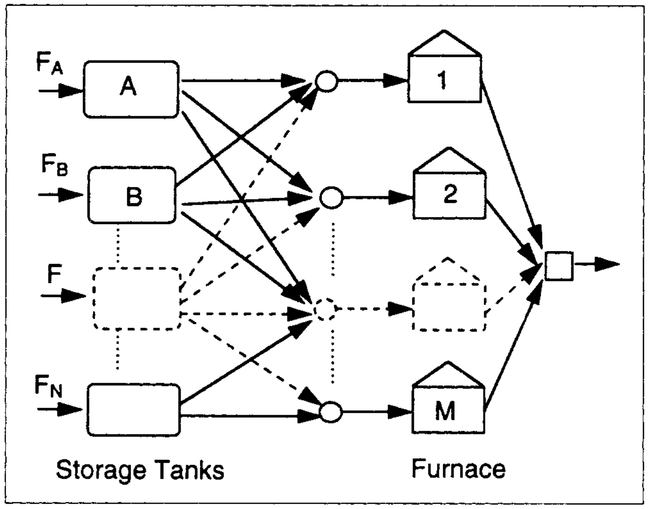

Figure 4. Ethylene scheduling problem.

the cleanup cost per day is lower in the first case. The same is true for loss in production time because of cleanup operation. On the other hand, the average conversion is higher in the second case (0.216 vs. 0.193) because of the more frequent maintenance of the furnace. This may result in higher income per unit time in the second case.

Clearly there is a trade-off between high conversion and high cleanup costs. The net profit is the difference between income and cleanup cost. Plotting net profit/time as a function of cycle time yields the curve shown in Figure 3. The net profit per unit time increases initially with the increase in cycle time because cleanup cost decreases, reaches a maximum, and then starts decreasing as average conversion goes down. To obtain the maximum profit ($1,721.6/d) the model proposed in this article predicts that this furnace should be cleaned about every 17 days, for a feed with the coking characteristics mentioned earlier.

The preceding analysis was only for a single feed and a single furnace. If there are multiple feeds and multiple furnaces, the trade-off is considerably more complex, as decisions regarding assignments of various feeds to various furnaces must also be made. In this case a repetitive schedule for all the feeds has to be determined, and the length of this schedule is called the "cycle time." A cycle for the previous case is equivalent to a subcycle in this generalized case of multiple feeds and multiple furnaces. The aim of this work is to model such a problem and to determine the schedules for feed processing by optimizing the performance. With this background we now state the problem considered in this article.

Problem Statement

For the sake of clarity in the presentation we will use the ethylene plant as a reference, even though the problem addressed in this article is generalizable to multiproduct continuous plants.

It is assumed that different feeds (i = A, B, C, ...) arrive continuously at an ethylene plant at a constant rate and are stored in different storage tanks (see Figure 4). These feedstocks are then processed sequentially or in parallel, depending on the number of furnaces ($l = 1, 2, \ldots, NF$) that are available in the plant. The rate of arrival of different feedstocks is a variable and is represented by F_i. This rate lies between a lower and an upper bound that are denoted by Flo_i and Fup_i, respectively. Each of the feedstocks can be processed in any of the furnaces, and the processing rate of feed i in furnace l is denoted by D_{il}. It is assumed that whenever there is a changeover, the furnace is cleaned, and the operating parameters set so that the furnace operates at the best possible conversion for that particular feed. The changeover time for feed i in furnace l is given by τ_{il}. It includes both the cleanup and setup times, and is assumed to be sequence independent. The setup and cleanup cost for the feed i in furnace l is given by the constant Cs_{il}. It is assumed that the operating conditions with which the conversion to ethylene decreases with time are kept constant in the furnace. (A similar treatment can be used for the case of constant conversion, as shown in Appendix A.) The conversion to ethylene is assumed to decrease exponentially with time (t) as in $c_{il} + a_{il}e^{-b_{il}t}$, where a_{il}, b_{il}, and c_{il} are given parameters for feed i and furnace l that are typically fitted with plant data. The problem is then to determine the feed assignment to furnaces, the number of subcycles for each feed with cleanups, and the total cycle time (T_{cycle}) in order to maximize the profit per unit time (for the case of a different cycle time for each furnace, see Appendix B).

The profit for the system described earlier can be determined as follows. The production rate of ethylene is the product of processing rate (D_{il}) and conversion. By integrating the production rate with time, the production as a function of time (or length of a subcycle T) is given by the equation

$$\int_0^T D_{il}(c_{il} + a_{il}e^{-b_{il}t})dt = D_{il}c_{il}T + \frac{D_{il}a_{il}(1-e^{-b_{il}T})}{b_{il}}. \quad (1)$$

The income is assumed to be directly proportional to production, and the proportionality constant is given by the price parameter P_i. This parameter takes into account the revenues obtained by selling the product and all the expenses incurred for producing the product, except the cleanup cost. Hence, the income is given by

$$P_i D_{il} c_{il} T + \frac{P_i D_{il} a_{il}}{b_{il}}(1 - e^{-b_{il}T}), \qquad (2)$$

which can also be expressed as

$$Cc_{il}T + Cp_{il}(1 - e^{-b_{il}T}), \qquad (3)$$

where $Cc_{il} = P_i D_{il} c_{il}$ and $Cp_{il} = (P_i D_{il} a_{il})/b_{il}$ are the new parameters.

If in each cycle of operation feed i has n_{il} subcycles in furnace l, the length of each subcycle (T) is given by t_{il}/n_{il}, where t_{il} is the total processing time of feed i in furnace l. The assumption that the length of the subcycles of feed i in furnace l are equal is shown to be true for the optimal solution in Appendix C. The income due to feed i in furnace l is obtained by substituting t_{il}/n_{il} for T in Eq. 3, and multiplying it by n_{il}. In this way, the income can be expressed as

$$Cc_{il}t_{il} + Cp_{il}n_{il}(1 - e^{-b_{il}t_{il}/n_{il}}). \qquad (4)$$

The net profit is the difference between income and cleanup cost. Hence, the net profit for feed i in furnace l is given by

$$NP_{il} = Cc_{il}t_{il} + Cp_{il}n_{il}(1 - e^{-b_{il}t_{il}/n_{il}}) - Cs_{il}n_{il}. \qquad (5)$$

Finally, the overall profit is determined by summing over all the feeds and all the furnaces:

$$\sum_i \sum_l NP_{il} = \sum_i \sum_l [Cc_{il}t_{il} + Cp_{il}n_{il}(1 - e^{-b_{il}t_{il}/n_{il}}) - Cs_{il}n_{il}]. \qquad (6)$$

It should be noted that the problem considered here is fairly general. Perhaps one of the perceived limitations is that it might be possible to operate the furnace without cleaning it at the time of changeover. In this case, however, it is very difficult to estimate the conversion for the new feedstock since its starting point with partial coking is not well defined. Furthermore, for practical reasons it is easier to perform a cleanup of the furnace when there is a changeover in the feedstock. Limitations of the model considered here include the assumptions that there is unlimited storage, there are no resource constraints, and the cost of inventory can be neglected.

MINLP Model

To develop the MINLP model it is useful to refer to the Gantt chart in Figure 5. The values of the following variables have to be determined

- t_{il} Total processing time of feed i in furnace l. If a feed is not assigned to a furnace, its value is zero (lower bound).
- n_{il} Number of subcycles of feed i in furnace l. Its value is zero if feed i is not assigned to furnace l. For numerical reasons (division by zero in Eq. 7) it will be assigned a very small lower bound of ϵ (e.g., $\epsilon = 0.0001$). It should be noted that n_{il} is an integer variable.
- F_i The rate of arrival of the feed i. Its value lies between a lower (Flo_i) and a upper (Fup_i) bound.
- Δt_{il} The total time devoted to feed i in furnace l. This includes both the processing time and the cleanup time.
- T_{cycle} The common cycle time for all the furnaces.

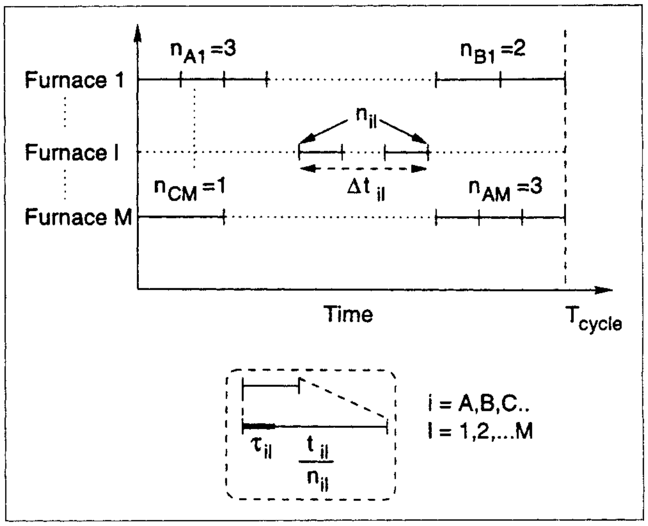

Figure 5. Gantt chart for a cyclic schedule.

Next, we develop the MINLP model for the general case of multiple furnaces and multiple feeds. This model is then reformulated for the special case of a single furnace and multiple feeds.

General case: Multiple furnace and multiple feeds

The problem at hand can be modeled as a mathematical programming problem that corresponds to an MINLP in which a nonlinear objective function has to be maximized subject to linear constraints. The objective function and constraints for the problem are as follows.

Objective Function. The objective of this problem is to maximize the profit. The expression derived in Eq. 6 can be used as an objective function. However, it first must be normalized on a per unit time basis to allow comparison with solutions with different cycle times. After dividing the net profit expression in Eq. 6 by the cycle time, the objective function is as follows:

$$\max z = \frac{\sum_{il}\left[Cc_{il}t_{il} + Cp_{il}n_{il}(1 - e^{-b_{il}t_{il}/n_{il}}) - Cs_{il}n_{il}\right]}{T_{\text{cycle}}}. \qquad (7)$$

Mass Balance. For each cycle of operation there should be no accumulation or depletion of feedstock in the system. Therefore, the amount of feed consumed should be equal to the amount of feed that came into the plant. The mass balance for each feed is then the following:

$$F_i T_{\text{cycle}} = \sum_l D_{il} t_{il} \quad \forall i.$$

Since both F_i and T_{cycle} are variables, this set of constraints involves a bilinear term on the lefthand side. However, the preceding constraints can be linearized by introducing a new variable S_i that corresponds to the extra amount of feedstock that must be processed over and above the minimum,

$$Flo_i T_{cycle} + S_i = \sum_l D_{il} t_{il} \quad \forall i \tag{8}$$

$$S_i \leq (Fup_i - Flo_i) T_{cycle} \quad \forall i. \tag{9}$$

Note that twice as many constraints are now required, but they are all linear.

Integrality Constraints for n_{il}. Binary variables are introduced to enforce that n_{il} is an integer:

$$n_{il} = \sum_{k=\epsilon,1,\ldots,K} k y_{ilk} \qquad \forall i,l \tag{10}$$

$$\sum_{k=\epsilon,1,\ldots,K} y_{ilk} = 1 \qquad \forall i,l \tag{11}$$

$$\epsilon = 0.0001. \tag{12}$$

Here y_{ilk} is a binary variable and ϵ is a constant with a value close to zero. If the number of subcycles of feed i in furnace l is equal to k, the binary variable y_{ilk} is one. It should be noted that k can either be ϵ or an integer greater than or equal to one. Therefore, even if the number of subcycles of feed i in furnace l is zero, n_{il} is not assigned a value of zero; instead, it is assigned a value ϵ. This, in conjunction with upper bound constraints (Eq. 15 below), prevents division by zero in the exponential term of the objective function without changing its numerical value. The parameter K is selected so as to be consistent with the upper bound on the number of subcycles n_{il}.

Timing Constraints. There are three constraints of this type:

• Equalities that relate total time allocated for feed i on furnace l to the corresponding processing time and clean up time,

$$\Delta t_{il} = n_{il} \tau_{il} + t_{il} \qquad \forall i,l. \tag{13}$$

• Inequalities that ensure that the total time allocated (processing time and cleanup time) on any furnace is not more than the cycle time of operation. Since the cycle time is in the denominator of the objective function, it can be easily shown that at least one of these constraints will be active:

$$\sum_i \Delta t_{il} \leq T_{cycle} \qquad \forall l. \tag{14}$$

• The following upper-bound constraints ensure that the processing time of feed i on furnace l is zero if the corresponding number of subcycles is zero:

$$t_{il} \leq U(1 - y_{\epsilon lk}) \qquad \forall i,l, \tag{15}$$

where U is a valid upper bound for the processing time.

Bounds. All the variables except n_{il} and T_{cycle} have a lower bound of zero. The variables n_{il} have a lower bound of • and a valid upper bound of K.

$$\epsilon \leq n_{il} \leq K, \quad S_i \geq 0, \quad \Delta t_{il} \geq 0, \quad t_{il} \geq 0, \quad \forall i,l$$
$$T_{cycle} > 0$$
$$y_{ilk} \in \{0,1\} \qquad \forall i,l,k. \tag{16}$$

Extra Constraints. These constraints do not affect the solution of the problem, but they can improve the computational time by tightening the MINLP model. These constraints are based on the idea that if a particular feed has a nonzero lower bound for the flow rate, then there is at least one subcycle for that feed. Hence,

$$\sum_l n_{il} \geq 1 \qquad \forall i \qquad (Flo_i > 0). \tag{17}$$

The MINLP problem (P1), with Eq. 7 as an objective function and Eqs. 8 to 17 as constraints, has the advantage that all the constraints are linear. This model is for the general case of multiple furnaces and multiple feeds. Before we consider the theoretical properties of this model, it is useful to consider the special case of a single furnace and multiple feeds.

Special case: Single furnace and multiple feeds

The case of a single furnace is important because in this case it is relatively easy to find a heuristic solution. Later in the article a comparison is made between the heuristic solution and the solution obtained using the proposed model. In this special case, since there is only one furnace, the index l can be dropped from all the variables. Furthermore, all the feeds will have at least one subcycle, each in a given cycle of operation (assuming a nonzero lower bound on feed rates). This then implies a lower bound of one for the variables n_i (index l has been dropped). Also, constraints (Eqs. 12 and 15) are no longer required. Hence, the simplified model (P2) for the case of single furnace and multiple feeds is as follows:

$$\text{P2:} \quad \max z = \frac{\Sigma_i \left[Cc_i t_i + Cp_i n_i (1 - e^{-b_i t_i / n_i}) - Cs_i n_i \right]}{T_{cycle}} \tag{18}$$

$$\text{s.t.} \quad Flo_i T_{cycle} + S_i = D_i t_i \qquad \forall i \tag{19}$$

$$S_i \leq (Fup_i - Flo_i) T_{cycle} \qquad \forall i \tag{20}$$

$$n_i = \sum_{k=1,\ldots,K} k y_{ik} \qquad \forall i \tag{21}$$

$$\sum_{k=1,\ldots,K} y_{ik} = 1 \qquad \forall i \tag{22}$$

$$\Delta t_i = n_i \tau_i + t_i \qquad \forall i \tag{23}$$

$$\sum_i \Delta t_i \leq T_{cycle} \tag{24}$$

$$1 \leq n_i \leq K, \quad S_i \geq 0, \quad \Delta t_i \geq 0, \quad t_i \geq 0, \quad \forall i$$
$$T_{cycle} > 0$$
$$y_{ik} \in \{0,1\}. \tag{25}$$

In this model (P2), Eqs. 18–20 correspond to Eqs. 7 to 9 of the general model. Similarly, Eqs. 21 and 22 correspond to Eqs. 10 and 11, and Eqs. 23 and 24 correspond to Eqs. 13 and 14, respectively. This simplified model also has linear constraints and a nonlinear objective function. Both these models ((P1) and (P2)) have special mathematical properties that can be exploited for finding a global optimum solution of the problem, as will be shown in the next section.

Mathematical Properties of the Model

All the constraints in models (P1) and (P2) developed earlier are linear. If the integrality constraints on the binary variables are relaxed, the solution space of the relaxed MINLP problems (RMINLPs) is a convex set. Let us denote this convex solution space by S. The objective function in these models is a nonlinear function with fractional and exponential terms. At first sight it is difficult to determine whether this function is concave over the solution space S to guarantee a unique global optimum. First, consider the numerator of the objective function (Eq. 18), which can be written as

$$\sum_{il} Cc_{il} t_{il} + \sum_{il} Cp_{il} n_{il} (1 - e^{-b_{il} t_{il}/n_{il}}) - \sum_{il} Cs_{il} n_{il}. \quad (26)$$

The preceding expression consists of three different terms. Clearly, the first and the third terms are linear, and the second term is nonlinear. It can be proved that this term is a concave function over the solution space S from the following theorem by Stubbs and Mehrotra (1996), which has been presented in a slightly modified form:

Theorem 1 (Stubbs and Mehrotra, 1996). Let $g(\bar{x})$ be a bounded convex (concave) function over a convex set C.

(a) If $(0 < \lambda \le K)$, then $h(\bar{x}, \lambda) = \lambda g(\bar{x}/\lambda)$ is a bounded convex (concave) function over $\tilde{C} = \{(\bar{x}, \lambda){:}\ \bar{x} \in C \wedge \lambda \in (0, K]\}$.

(b) If $(0 \le \lambda \le K)$ and $(\lambda = 0 \Rightarrow \bar{x} = 0)$, then

$$h(\bar{x}, \lambda) = \begin{cases} \lambda g(\bar{x}/\lambda){:} & \lambda \neq 0 \\ 0{:} & \lambda = 0 \end{cases}$$

is a bounded convex (concave) function over $\tilde{C} = \{(\bar{x}, \lambda){:}\ \bar{x} \in C \wedge \lambda \in [0, K]\}$.

Proof. See Appendix D.

Consider the concave function $g(t_{il}) = Cp_{il}\,(1 - e^{-b_{il} t_{il}})$ defined over S. From Theorem 1, the function

$$h(t_{il}, n_{il}) = n_{il} g(t_{il}/n_{il}) = Cp_{il} n_{il} (1 - e^{-b_{il} t_{il}/n_{il}})$$

is a concave function over S because $0 < n_{il} \le K$ (in both the models (P1) and (P2)). The summation of concave functions is also a concave function (Bazaraa et al., 1993), and therefore the second term in Eq. 26 is a concave function, which implies that the numerator of the objective function (Eq. 7 or Eq. 18) is a concave function over S. Next we state a theorem by Avriel (1976) that establishes the exact nature of the objective function.

Theorem 2 (Avriel, 1976). Let $f(\bar{x})$ be a real valued convex (concave) function and $g(\bar{x})$ be a positive linear function over a convex set C. Then $h(\bar{x}) = f(\bar{x})/g(\bar{x})$ is a quasi-convex (quasi-concave) function over C. Furthermore, if both $f(\bar{x})$ and $g(\bar{x})$ are differentiable, then $h(\bar{x})$ is a pseudoconvex $[(\bar{x}^2 - \bar{x}^1)^T \nabla_x h(\bar{x}^1) \ge 0 \Rightarrow h(\bar{x}^2) \ge h(\bar{x}^1)\ \forall \bar{x}^1,\ \bar{x}^2 \in C]$ (pseudoconcave $[(\bar{x}^2 - \bar{x}^1)^T \nabla_x h(\bar{x}^1) \le 0 \Rightarrow h(\bar{x}^2) \le h(\bar{x}^1)\ \forall \bar{x}^1,\ \bar{x}^2 \in C]$ function over C.

The numerator of the objective function is a concave function and the denominator (T_{cycle}) is a positive linear function ($T_{\text{cycle}} > 0$). Both of them are also differentiable. Therefore, the objective function is a pseudoconcave function over S. It should be noted that this result regarding the nature of the objective function is fairly general. It is true for any objective function that uses a concave function to predict the amount of ethylene produced in a subcycle (as in Eq. 1).

If the integrality constraints in the MINLP ((P1) or (P2)) are relaxed, each reduces to a nonlinear programming (NLP) problem (RMINLP). The RMINLP in this case has a pseudoconcave objective function that has to be maximized over linear constraints. The following theorem establishes the globality of the solution:

Theorem 3 (Kuhn-Tucker). Every local minimum (maximum) of a pseudoconvex (pseudoconcave) function over a convex set C is also a global minimum (maximum), and Kuhn-Tucker differential conditions are sufficient for global optimality.

Proof. See Bazaraa et al. (1993).

This theorem implies that for the RMINLPs that arise from (P1) and (P2), a solution that is a local optimum is guaranteed to be a global optimum. Next, we present a theorem that will be used as the basis for finding globally optimal solutions of models (P1) and (P2) by solving their corresponding RMINLPs.

Theorem 4. Given is an MINLP where a pseudoconvex (pseudoconcave) function has to be minimized (maximized) subject to a set of constraints. If the feasible solution space obtained after relaxing the integrality constraints on the binary variables is a convex set, the optimal solution of the corresponding RMINLP is a rigorous lower bound (upper bound) for the objective function.

Proof. The RMINLP has a pseudoconvex (pseudoconcave) function that has to be minimized (maximized) over a convex set. Theorem 3 implies that it has a global optimum. Since the feasible solution space of an RMINLP is a superset of the feasible solution space of an MINLP, the optimal solution of RMINLP provides a rigorous lower (upper) bound for the objective function.

Since the MINLP at hand is a maximization problem, the optimal solution for the corresponding RMINLP gives a rigorous upper bound for the objective function. This property can be exploited for obtaining a global optimum for the MINLP problems (P1) and (P2), as shown in the next section. It is also interesting to note that for the case of a fixed time horizon, the problem reduces to a convex MINLP problem.

Solution Algorithm

The problem of finding the maximum of a function is equivalent to finding the minimum of the negative of that function. For the sake of clarity, we assume that instead of maximizing profit, we are minimizing the negative of profit. In this case, the objective function is a pseudoconvex function, and the minimum has to be found over the linear constraint set.

Consider a general MINLP problem of the form,

$$\begin{aligned} \text{M1.} \quad & \min\ f(x,y) \\ \text{s.t.} \quad & g_j(x,y) \le 0 \quad \forall j \\ & x \in X, \quad y \in Y, \end{aligned}$$

where x and y are continuous and discrete variables, respectively, defined over their corresponding sets X and Y. There are a number of methods for solving MINLP problems, but most of them have some additional restrictions on the structure of the problems. These methods fall into two different categories:

- Two-level decomposition methods.
- Branch-and-bound type enumerative methods.

The various two-level decomposition methods are the Generalized Benders Decomposition method (Geoffrion, 1972), the Outer-Approximation method (Duran and Grossmann, 1986), the Generalized Outer Approximation method (Yuan et al., 1989), the Linear Programming/Nonlinear Programming (LP/NLP)-based branch and bound (Quesada and Grossmann, 1992), the Mixed Integer Quadratic Outer Approximation method (Fletcher and Leyffer, 1994), and the Extended Cutting Plane method (Westerlund and Petterson, 1995). In all these methods the key idea is that the nonlinear functions are approximated by supporting hyperplanes. For convex functions these hyperplanes yield a relaxation of the feasible space and underestimators for the objective function (see Figure 6). Therefore, these methods ensure the global optimum for convex MINLP problems. A summary of all these methods and the key differences between them has been presented by Grossmann and Kravanja (1997).

If the nonlinear functions are pseudoconvex, then the algorithms mentioned in the previous paragraph cannot guarantee the global optimum. This is because the linearizations of these functions are not supporting hyperplanes at some points. As a result, the linearizations might cut off a part of the feasible region, and may not underestimate the objective function (see Figure 7).

Recently, Westerlund et al. (1997) have proposed a method based on the Extended Cutting Plane method for a class of nonconvex problems. They have assumed that in the MINLP (M1) the objective function $[f(x,y)]$ is linear and the functions $[(g_j(x,y)]$ on the lefthand side of the constraints are pseudoconvex. One might think that the problem at hand can meet these criteria by reformulating it as follows:

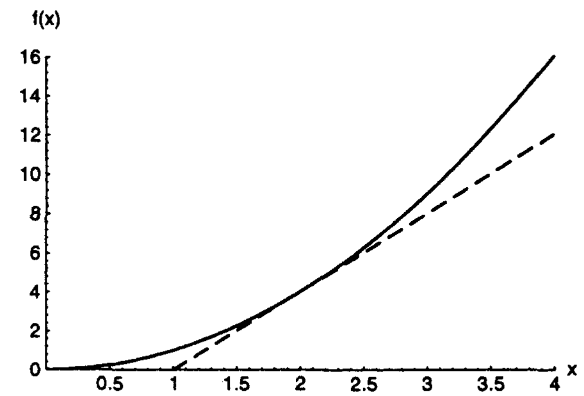

Figure 6. Linearization underestimates the convex function.

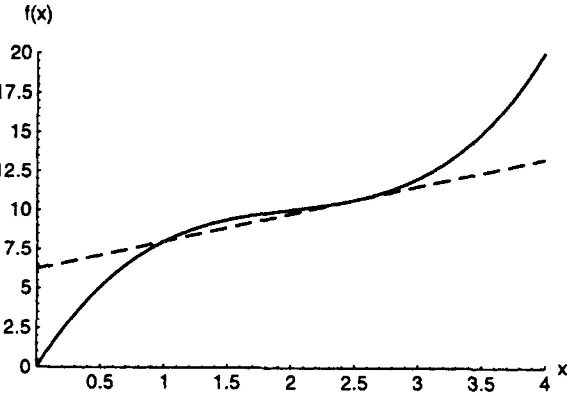

Figure 7. Linearization may not underestimate the pseudoconvex function.

$$\begin{aligned} \text{M2:} \quad & \min\ \alpha \\ \text{s.t.} \quad & g_j(x,y) \le 0 \quad \forall j \\ & f(x,y) - \alpha \le 0 \\ & x \in X, \quad y \in Y. \end{aligned}$$

However, it should be noted that linear functions (α) added to or subtracted from a pseudoconvex function $[f(x,y)]$ need not generally result in a pseudoconvex function. Hence, the reformulation (M2) will not satisfy the criteria for this method. For these reasons none of the known two-level decomposition methods will guarantee the global optimum for MINLP problems (P1) and (P2).

Another class of methods for solving MINLPs are the enumerative search techniques based on branch and bound. The NLP-based branch and bound for solving a convex MINLP (Gupta and Ravindran, 1985; Borchers and Mitchell, 1994; and Stubbs and Mehrotra, 1996) is similar to the branch-and-bound method for Mixed Integer Linear Program (MILP) (Nemhauser and Wolsey, 1988). The first step in this method consists of solving the corresponding RMINLP. If all the integer variables take integral values, the search is stopped. Otherwise, a tree search is performed in the space of the integer variables. The key requirement for this method is that the RMINLP must yield a rigorous lower bound for the corresponding MINLP. This method was originally proposed for convex MINLPs, but it can also be used for solving MINLPs where a corresponding RMINLP involves minimizing a pseudoconvex objective function over a convex set. This follows from the fact that for these MINLPs the corresponding RMINLP yields a rigorous lower bound (see Theorem 4). The NLP-based branch-and-bound (B&B) method will therefore guarantee the global optimum for MINLPs (P1) and (P2). This method is attractive only if NLP subproblems are relatively inexpensive to solve, or the lower bound obtained by solving the RMINLP is tight, as is the case for models (P1) and (P2). A brief description of the B&B algorithm is as follows.

The B&B method for solving an MINLP involves solving a series of NLP subproblems. Let us assume that the MINLP problem to be solved is a minimization problem. Furthermore, assume that the structure of the MINLP is such that the corresponding RMINLP gives a rigorous lower bound for the objective function. Let r^0 denote the RMINLP corresponding to the MINLP to be solved. Any feasible solution of the MINLP gives an upper bound (UB) on the optimal value of the objective function. Let us denote the set of NLP subproblems to be solved by R^*. The major steps in the algorithm are as follows:

Step 1 (Initialization). $\mathrm{UB} = \infty$, $R^* = \{r^0\}$.

Step 2 (Check if there are Any More Subproblems to Solve). If $R^* = \{\phi\}$, then go to Step 5, else go to Step 3.

Step 3 (Fathom a Node). Select an NLP r from the set R^*. The criterion for selecting an NLP is also called as the *node selection rule*. There are many different rules for selecting an NLP, and they play a key role in the efficiency of the B&B algorithm (Nemhauser and Wolsey, 1988).

Solve the NLP subproblem r and remove it from the set R^*. If the NLP is infeasible or the optimal value of the objective function (lower bound) is greater than the UB, go to step 2. If the optimal solution has integral values for all the integer variables, then update the UB and go to Step 2. Otherwise, go to Step 4.

Step 4 (Branch on a Variable). There is at least one integer variable that has a nonintegral value. Of all the variables that have been assigned nonintegral values, select one according to some prespecified *branching variable selection rule* (Nemhauser and Wolsey, 1988). Let us denote this integer variable by y_r. Generate two new NLP subproblems r^1 and r^2 and add them to the set R^*. The subproblem r^1 is generated by specifying the floor of the optimal value of y_r as the upper bound for y_r, and the subproblem r^2 is generated by specifying the ceiling of the optimal value of y_r as the lower bound of y_r. Go to Step 2.

Step 5 (Termination). If the $\mathrm{UB} = \infty$, then the MINLP does not have a feasible solution or it is unbounded. Otherwise, the UB is the optimal solution.

An advantage of using B&B in this fashion is that branching can be done in the space of general integer variables (n_{il}) rather than on the binary variables. However, special care has to be taken whenever the optimal value of n_{il} in any of the subproblems is equal to its lower bound of ϵ. In this situation n_{il} should be treated as having an integer value of 0, and no branching is required on this variable.

Computational Results

In this section we consider a number of different example problems of varying complexity. These example problems are solved using the NLP-based B&B method and the augmented-penalty outer-approximation method as implemented in DICOPT++ (Viswanathan and Grossmann, 1990). As pointed out in the previous section, the efficiency of the B&B algorithm depends on the rules used for "Node Selection" and "Branching Variable Selection" (Nemhauser and Wolsey, 1988). The aim of this article is not to study the effects of these rules on the efficiency of the B&B method, but to highlight the importance of the rigorous solutions obtained by using these methods. Two different node-selection rules have been considered in solving the problems using the B&B method, namely: (a) Breadth First Search (BFS) or First In First Out (FIFO), and (b) Depth First Search (DFS) or Last In First Out (LIFO). In both these cases branching is performed on the variable that is farthest away from an integer value. At each node, NLP subproblems were solved using MINOS (Murtagh and Saunders, 1985). The MILP master problems and NLP subproblems in the OA algorithm were solved using OSL (OSL, 1991) and MINOS (Murtagh and Saunders, 1985), respectively. The modeling system GAMS (Brooke et al., 1992) was used to implement the models and the solution algorithms.

Table 1. Values of Various Parameters for Example 1

	Feed i		
Parameters	A	B	C
τ_i (d)	2	3	3
D_i (ton/d)	1,300	1,000	1,100
a_i	0.20	0.18	0.19
b_i (1/d)	0.10	0.13	0.09
c_i	0.18	0.10	0.12
P_i ($/ton)	160	90	120
Cs_i ($)	100	90	80
Flo_i (ton/d)	350	300	300
Fup_i (ton/d)	650	600	600

Example 1. It is desired to determine a cyclic schedule to process three different feeds A, B, and C in a cracking furnace. The minimum flow rates of A, B, and C are 350, 300, and 300 ton/d, respectively. Various parameters associated with the process are listed in Table 1.

For comparison purposes, a simple heuristic approach will be used to schedule these three different feeds on the furnace. In this heuristic a cycle of operation involves one subcycle for every feed, and the processing time (length of subcycle) for each feed is determined by its minimum flow rate and relative profitability. The specific heuristic is as follows:

1. In each cycle of operation use exactly one subcycle for each feed.
2. Assume that on average the cleanup is performed every one and a half months. Therefore, the overall cycle time is 135 days.
3. Since it is least profitable to produce ethylene from feed B, only the minimum quantity of feed B is processed. The processing time of B in each cycle can be calculated using the mass balance:

$$t_B = \frac{F_B T_{\mathrm{cycle}}}{D_B}$$

$$t_B = \frac{(300)(135)}{1,000} = 40.5 \text{ days}.$$

4. It is less profitable to produce ethylene from feed C than from feed A. Therefore, the minimum amount of feed C is processed. The processing time for feed C can be calculated as follows:

$$t_C = \frac{(300)(135)}{1,100} = 36.82 \text{ days}.$$

5. The processing time available for feed A is then calculated as follows:

$t_A = 135 - 40.5 - 36.82 - (2+3+3) = 49.68$ days.

6. Calculate the flow rate required for feed A to meet its demand:

$$F_A = \frac{D_A T_A}{T_{\text{cycle}}}$$

$$F_A = \frac{(1{,}300)(49.68)}{135} = 478.4 \text{ ton/d}.$$

This flow rate lies between the upper and the lower bound. Hence, we have obtained a feasible schedule. Conversion curves for this type of schedule are shown in Figure 8. It should be noted that if the required flow rate of A is less than the lower bound, then a feasible schedule cannot be obtained for the assumed cycle time. In that case the whole procedure is repeated by assuming a longer cycle time. However, if the flow rate is greater than the upper bound, fix the flow rate of A to its upper bound and calculate its processing time. Using this processing time, calculate the processing time and the flow rate for feed C. Check if the required flow rate for feed C is within the bounds. If not, repeat the same procedure for feed C.

An optimal solution obtained by using model (P2) is shown in Figure 9. This model predicts a schedule with many more changeovers and shorter subcycles. The value of the objective function for the optimal solution is \$30,430/d, while for the heuristic solution it is \$26,763/d. Hence, the optimal value is 12.1% higher than the heuristic solution. This difference can be reduced if the processing times are optimized by keeping one subcycle for each feed. In this case profit is increased to \$29,279/d, which is still 3.8% lower than the optimal.

The resulting MINLP for this example had 12 binary variables, 26 continuous variables, and 20 constraints. It was solved to optimality using the B&B algorithm, requiring three nodes for both the DFS branching rule and the BFS branching rule. The tree search for the DFS case with lower and upper bounds at each node is shown in Figure 10. It should be noted that the lower and upper bounds have a negative sign because the negative of the profit was minimized. As can be seen in Figure 10, the initial lower bound of $-30{,}443.71$ is obtained by solving the relaxed MINLP. Since the number of subcycles for feed C is fractional (1.74), we branch on that variable. Adding the inequality $n_c \geq 2$ leads to an integer solution at node 2, and hence the upper bound of $-30{,}430.18$. Adding the inequality $n_c \leq 1$ at node 3 results in a fractional solution and a lower bound of $-30{,}294.98$. Since the lower bound exceeds the upper bound, node 3 is fathomed and the search is terminated, confirming that the solution with four subcycles for A, one subcycle for B, and two subcycles (see Figure 9) is the global optimum. It took 0.22 CPU seconds on a HP 9000/C110 workstation to perform this B&B search. It should also be noted that the relaxation of the problem is very tight (gap < 0.01%). This is because there are no upper-bound constraints in the model (P2). In this particular example DICOPT++ gave the same solution as the B&B method.

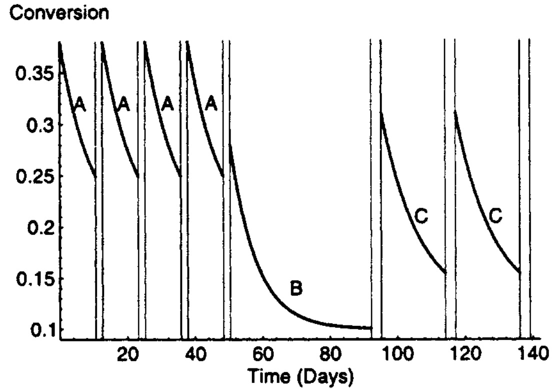

Figure 9. Optimal solution for Example 1.

It is interesting to note that the decision regarding the number of subcycles is also dependent on the relative coking characteristics of different feeds. Since it is more profitable to produce ethylene from feeds A and C rather than from feed B, multiple subcycles are used for feeds A and C and only one subcycle is used for feed B (Figure 9). In this particular example the cleanup times that are required for different feeds are very similar. In the next example we consider a case when there are multiple furnaces, and the cleanup times for the different feeds are significantly different.

Example 2. Consider an ethylene plant that has four different furnaces and processes seven different feeds (A, ..., G). The minimum flow rates for these seven feeds are 300, 400, 300, 500, 500, 100 and 600 ton/d. Various parameters for this problem are listed in Table 2.

The problem is solved using the model (P1). The resulting MINLP has 140 binary variables, 233 continuous variables,

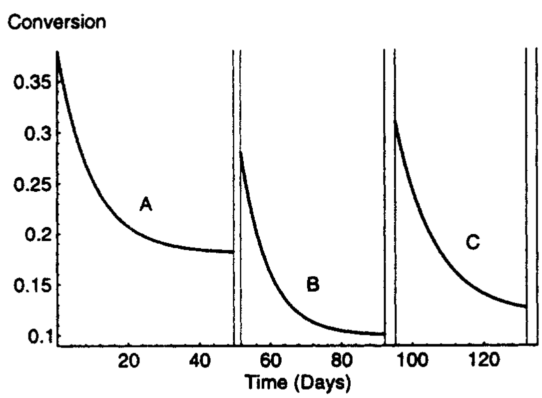

Figure 8. Heuristic solution for Example 1.

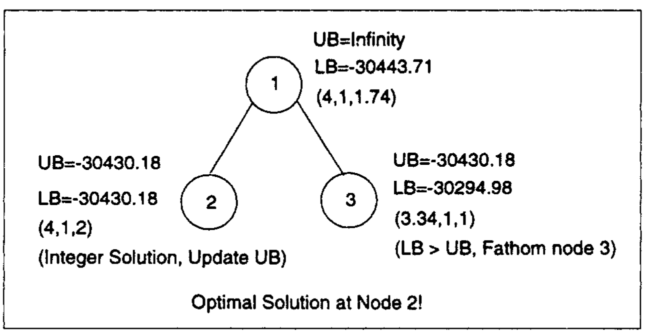

Figure 10. B&B tree search using the DFS rule for Example 1.

Table 2. Values of Various Parameters for Example 2

Parameter	Furnace l	Feed i A	B	C	D	E	F	G
τ_{il} (d)	1	2	3	3	3	1	2	3
	2	3	1	2	2	2	1	1
	3	1	3	1	1	2	1	2
	4	2	1	3	2	2	1	1
D_{il} (ton/d)	1	1,300	1,200	1,100	800	1,300	300	700
	2	1,100	1,050	1,000	1,000	1,200	400	600
	3	900	800	800	1,200	1,000	300	850
	4	1,200	1,000	800	700	1,200	400	600
a_{il}	1	0.3	0.4	0.35	0.32	0.29	0.35	0.31
	2	0.32	0.38	0.33	0.31	0.28	0.4	0.34
	3	0.31	0.35	0.36	0.36	0.29	0.37	0.31
	4	0.31	0.36	0.35	0.36	0.28	0.39	0.32
b_{il} (1/d)	1	0.10	0.20	0.10	0.20	0.23	0.34	0.20
	2	0.32	0.38	0.33	0.31	0.28	0.4	0.34
	3	0.31	0.35	0.36	0.36	0.29	0.37	0.31
	4	0.31	0.36	0.35	0.36	0.28	0.39	0.32
c_{il}	1	0.20	0.18	0.21	0.20	0.3	0.26	0.16
	2	0.21	0.19	0.23	0.25	0.31	0.27	0.17
	3	0.19	0.18	0.21	0.23	0.30	0.25	0.18
	4	0.20	0.19	0.21	0.24	0.31	0.26	0.17
P_{il} (\$/ton)	1	123	105	110	123	105	110	120
	2	114	132	129	114	132	129	113
	3	110	122	120	110	122	120	117
	4	120	125	129	115	115	128	115
Cs_{il} (\$)	1	100	90	80	75	90	93	78
	2	80	85	75	90	94	78	70
	3	90	90	90	85	93	92	75
	4	80	90	85	80	92	85	72
Flo_i (ton/d)	—	300	400	300	500	500	100	600
Fup_i (ton/d)	—	600	700	600	800	800	400	900

and 138 constraints. The conversion of various feeds on different furnaces for the optimal schedule is shown in Figures 11 to 14. The solution of this problem required 85 nodes and 49.03 CPU seconds using the B&B algorithm with the DFS rule. This problem could not be solved in reasonable time (2,000 nodes) by B&B with the BFS rule. It is interesting to note that for this example DICOPT++ gave a slightly suboptimal solution (\$165,382 vs. \$165,395).

Three main decisions are made by solving the model: the assignment of various feeds to different furnaces; the number of subcycles of each feed on the assigned furnace; and the length of these subcycles. These decisions are made by con-

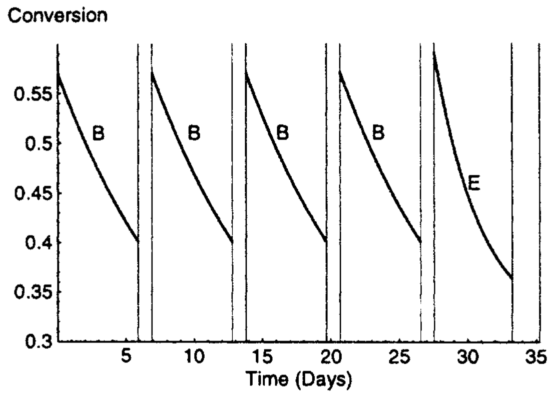

Figure 12. Optimal solution for furnace 2 in Example 2.

sidering the relative profitability, relative cleanup time and cost, and the coking characteristics of different feeds on each furnace. For example, in furnace 1 (Figure 11) the cleanup time for feeds A and C is larger than for feed D. Therefore, two subcycles for feed E are used as compared to feeds A and C, which have only a subcycle each. It should be noted that the same feed can be assigned to more than one furnace, and the number of subcycles of the same feed on different furnaces can be different. For example, two subcycles are required for feed E in the first furnace and only one subcycle is required in the second and fourth furnaces (see Figures 12 and 14). On the whole, the trade-offs are complex and interdependent. It is therefore difficult to make these decisions intuitively. While obtaining a heuristic solution for the case of single furnace is not that difficult, it can lead to suboptimal solutions. Furthermore, for the case of multiple furnaces, finding a good feasible solution is nontrivial. The proposed model can clearly overcome these difficulties.

Other results

Besides the two example problems presented earlier, a number of other example problems were solved and the results are summarized in Table 3. This table highlights some interesting results regarding the computational aspects of the problem. The relaxation of the problem is very tight for all the examples considered. In particular, the relative gap is extremely small for the single-furnace case; it is less than 1% for the three single-furnace examples considered. The B&B

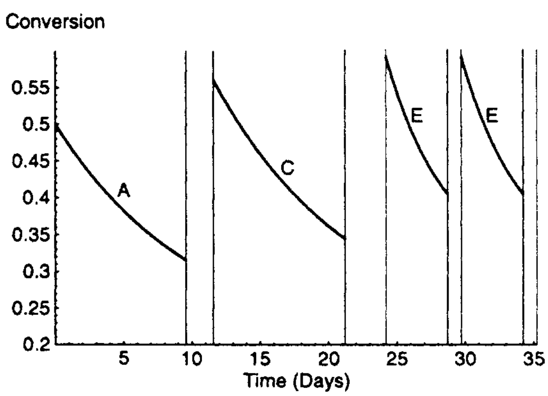

Figure 11. Optimal solution for furnace 1 in Example 2.

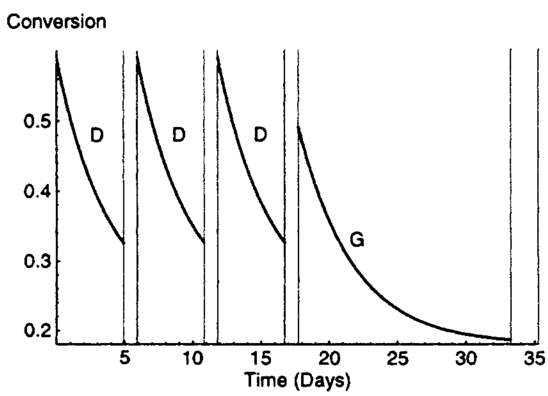

Figure 13. Optimal solution for furnace 3 in Example 2.

Table 3. Computational Results

Feeds	Furnaces	Relaxation**	Binary Vars. Cont. Vars. Constraints	Algorithm	Nodes (B&B) or Major Iterations (OA)[†]	CPU* (s)	Profit ($/day)
3	1	30,443.71	12	B&B (BFS)	3	0.17	30,430.18
			26	B&B (DFS)	3	0.22	30,430.18
			20	DICOPT++	5	0.57	30,430.18
5	1	24,928.23	20	B&B (BFS)	7	0.26	24,928.21
			42	B&B (DFS)	5	0.18	24,928.21
			32	DICOPT++	7	0.92	24,928.21
7	1	28,167.02	28	B&B (BFS)	25	1.54	28,160.02
			58	B&B (DFS)	13	1.070	28,160.02
			44	DICOPT++	31	10.89	28,160.02
3	3	149,287.50	45	B&B (BFS)	117	20.40	147,729.05
			77	B&B (DFS)	41	7.35	147,729.05
			46	DICOPT++	29	71.12	142,975.91
6	3	120,083.58	90	B&B (BFS)	—	—	—
			152	B&B (DFS)	81	24.81	121,555.90
			94	DICOPT++	36	269.39	121,554.40
4	4	186,286.49	80	B&B (BFS)	—	—	—
			134	B&B (DFS)	143	47.68	184,361.30
			81	DICOPT++	26	61.31	184,361.30
7	4	165,988.60	140	B&B (BFS)	—	—	—
			233	B&B (DFS)	85	49.03	165,398.70
			138	DICOPT++	45	1,326.80	165,382.27

*On a HP 9000/C110 workstation.
**Profit for RMINLP.
[†]Termination criteria for DICOPT++: crossover of bounds.

method with the DFS rule performed more efficiently than DICOPT++ in most of the cases. Furthermore, in some cases DICOPT++ gave a suboptimal solution. The efficiency of B&B is a result of a very tight formulation that yields a very good relaxation. It should be noted that the computational time reported is the time it took to solve the NLPs and does not include the overhead of branching. Also, larger example problems could not be solved by B&B with the BFS rule in a reasonable amount of time (2,000 nodes).

Conclusions

A special class of problems dealing with the scheduling of multiple feeds on parallel units has been considered in this article. A unique feature of this problem is that the performance of a processing unit decreases with time, and therefore it has to be shut down for maintenance after regular intervals. Application in ethylene plants was the major motivation for this work. Process models were used to predict the exponential decay in conversion to ethylene with time. It was shown that the trade-off for the optimal solution is a function of the average performance, the maintenance cost, and the production loss because of the shutdown.

An MINLP model has been developed to determine the cyclic schedules for these types of problems. It was shown that for this MINLP the objective function is pseudoconcave and the constraints are linear. This property was exploited to solve the problem of global optimality using the NLP-based B&B method. It was demonstrated that the schedules obtained using this model can lead to a considerable increase in profits.

The proposed MINLP model yields a very tight formulation, thereby providing very good bounds at each node of the B&B tree. As a result of this, the B&B method proved to be very effective for solving the example problems. It was also shown that the outer-approximation method may result in suboptimal solutions for this problem, although the actual differences were small in most cases. Finally, the results for problems with up to seven feeds and four furnaces showed that the proposed models have the capability of efficiently finding globally optimal schedules.

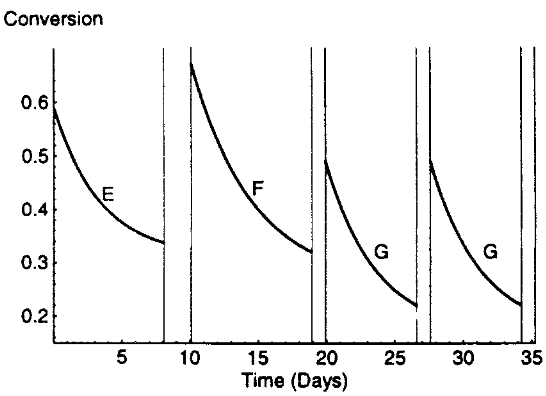

Figure 14. Optimal solution for furnace 4 in Example 2.

Acknowledgments

Financial support from the Computer Aided Process Design Consortium at Carnegie Mellon University is gratefully acknowledged. The authors also thank Dr. Bhieng Tjoa of Mitsubishi Chemical Corp. for useful discussions on the workings of ethylene plants.

Literature Cited

Avriel, M., *Nonlinear Programming: Analysis and Methods*, Prentice Hall, Englewood Cliffs, NJ (1976).

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd ed., Wiley, New York (1993).

Borchers, B., and J. E. Mitchell, "An Improved Branch and Bound Method for Mixed Integer Nonlinear Programming," *Comput. Oper. Res.*, **21**, 359 (1994).

Brooke, A., D. Kendrick, and A. Meeraus, *GAMS: A Users Guide*, Scientific Press, Palo Alto, CA (1992).

Duran, M. A., and I. E. Grossmann, "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs," *Math. Program.*, **36**, 307 (1986).

Elmaghraby, S. E., "The Economic Lot Scheduling Problem (ELSP): Review and Extensions," *Manag. Sci.*, **24**, 587 (1978).

Fletcher, R., and S. Leyffer, "Solving Mixed Integer Nonlinear Programs by Outer Approximation," *Math Program.*, **66**, 327 (1994).

Geoffrion, A. M., "Generalized Benders Decomposition," *J. Optim. Theory Appl.*, **10**(4), 237 (1972).

Grossmann, I. E., and Z. Kravanja, "Mixed-Integer Nonlinear Programming: A Survey of Algorithms and Applications," *Large-Scale Optimization with Applications*, L. T. Biegler, T. F. Coleman, A. R. Conn, and F. N. Santosa, eds., Vol. 93, *IMA Volumes in Mathematics and Its Applications*, Springer-Verlag, New York (1997).

Gupta, O. P., and V. Ravindran, "Branch and Bound Experiments in Convex Nonlinear Integer Programming," *Manag. Sci.*, **31**, 1533 (1985).

Lee, A. K. K., and A. M. Aitani, "Saudi Ethylene Plants Move Toward More Feed Flexibility," *Oil & Gas J. (Special Issue)* (1990).

Matar, S., and L. F. Hatch, *Chemistry of Petrochemical Process*, Gulf Pub., Houston (1994).

Murtagh, B. A., and M. A. Sanders, *MINOS User's Guide*, Systems Optimization Laboratory, Dept. of Operations Research, Stanford Univ., Stanford, CA (1985).

Nemhauser, G. L., and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York (1988).

OSL, *Optimization Subroutine Library Guide and Reference Release 2*, IBM, Kingston, NY (1991).

Pinto, J. M., and I. E. Grossmann, "Optimal Cyclic Scheduling of Multistage Continuous Multiproduct Plants," *Comput. Chem. Eng.*, **18**(9), 797 (1994).

Pinto, J. M., and I. E. Grossmann, "Assignment and Sequencing Models for the Scheduling of Chemical Process," *Ann. Oper. Res.* (1998).

Quesada, I., and I. E. Grossmann, "An LP/NLP Based Branch-and-Bound Algorithm for Convex MINLP Optimization Problems," *Comput. Chem. Eng.*, **19**, 937 (1992).

Reklaitis, G. V., "Overview of Scheduling and Planning of Batch Process Operations," NATO Advanced Study Institute—Batch Process System Engineering, Antalya, Turkey (1992).

Sahinidis, N. V., and I. E. Grossmann, "MINLP Model for Cyclic Multiproduct Scheduling on Continuous Parallel Lines," *Comput. Chem. Eng.*, **15**, 85 (1991).

Schilling, G., and C. C. Pantelides, "Optimal Periodic Scheduling of Multipurpose Plants in Continuous Time Domain," *Comput. Chem. Eng.*, **21**(Suppl.), S1191 (1997).

Schilling, G., Y. E. Pineau, C. C. Pantelides, and N. Shah, "Optimal Scheduling of Multipurpose Continuous Plants," AIChE Meeting, San Francisco, CA (1994).

Shah, N., C. C. Pantelidis, and R. W. H. Sargent, "Optimal Periodic Scheduling of Multipurpose Batch Plants," *Ann. Oper. Res.*, **42**, 193 (1993).

Stubbs, R. A., and S. Mehrotra, "A Branch-and-Cut Method for 0-1 Mixed Convex Programming," Tech. Rep. 96-01, Dept. of IE/MS, Northwestern Univ., Evanston, IL (1996).

Viswanathan, J., and I. E. Grossmann, "A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization," *Comput. Chem. Eng.*, **14**, 769 (1990).

Westerlund, T., and F. Petterson, "An Extended Cutting Plane Method for Solving Convex MINLP Problems," *Comput. Chem. Eng.*, **19**(Suppl.), S131 (1995).

Westerlund, T., H. Skrifvars, I. Harjunkoski, and R. Pörn, "An Extended Cutting Plane Method for a Class of Non-Convex MINLP Problems," *Comput. Chem. Eng.* (1997).

Yuan, X., L. Pibouleau, and S. Domenech, "Experiments in Process Synthesis via Mixed Integer Programming," *Chem. Eng. Process.*, **25**, 99 (1989).

Appendix A: Model for the Case of Constant Conversion

In this case instead of maintaining constant operating conditions, the processing rate of the feedstock is decreased and the coil temperature in the furnace is increased over time. This results in constant conversion of the feedstock to ethylene. Let us assume that processing rate decreases exponentially with time ($D_{il} = C_{il} + A_{il} e^{-B_{il} t}$), and that the energy cost increases linearly with time ($p_{il} + q_{il} t$). These assumptions are simple approximations and can be relaxed by using comprehensive models. This case can be easily modeled using the concepts presented in the third and fourth sections. The key differences over the MINLP model (M1) are as follows:

1. Since the processing rate is no longer a constant, the amount of feedstock processed in a furnace in each cycle is no longer a linear function of processing time. Hence, it should be calculated by integrating the processing rate over time (processing time in a subcycle), and then multiplying by the number of subcycles. Therefore, the amount of feedstock i processed in furnace l is given by

$$
\begin{aligned}
& n_{il} \int_0^{t_{il}/n_{il}} D_{il}\, dt \\
&= n_{il} \int_0^{t_{il}/n_{il}} \left(C_{il} + A_{il} e^{-B_{il} t} \right) dt \\
&= C_{il} t_{il} + \frac{A_{il} n_{il}}{B_{il}} \left(1 - e^{-B_{il} t_{il}/n_{il}}\right).
\end{aligned}
$$

The mass balance equation (Eq. 8) has to be modified to

$$
Flo_i T_{\text{cycle}} + S_i = \sum_l \left[C_{il} t_{il} + \frac{A_{il} n_{il}}{B_{il}} \left(1 - e^{-B_{il} t_{il}/n_{il}}\right) \right] \quad \forall i. \tag{A1}
$$

All other constraints in model (M1) are the same for this case.

2. Since the process model is no longer the same, the objective function has to be rederived. In this case, the conversion to ethylene X_{il} is constant. The production of ethylene from feed i in furnace l is given by

$$
C_{il} X_{il} t_{il} + \frac{A_{il} X_{il} n_{il}}{B_{il}} \left(1 - e^{-B_{il} t_{il}/n_{il}}\right).
$$

The inflow is given by

$$
P_{il} C_{il} X_{il} t_{il} + \frac{P_{il} A_{il} X_{il}}{B_{il}} n_{il} \left(1 - e^{-B_{il} t_{il}/n_{il}}\right).
$$

This expression can be rewritten in the simplified form as

$$
Cc_{il} t_{il} + Cp_{il} n_{il} \left(1 - e^{-B_{il} t_{il}/n_{il}}\right), \tag{A2}
$$

where $Cc_{il} = P_{il}C_{il}X_{il}$ and $Cp_{il} = P_{il}A_{il}X_{il}/B_{il}$. This expression is exactly the same as the one derived in Eq. 4, except that it does not take into account the increasing utility cost. It must be accounted for separately, and for feed i in furnace l it can be calculated as

$$n_{il}\int_0^{t_{il}/n_{il}}(p_{il} + q_{il}t)\,dt \tag{A3}$$

$$= n_{il}\left[p_{il}\frac{t_{il}}{n_{il}} + q_{il}\left(\frac{t_{il}}{n_{il}}\right)^2\right]. \tag{A4}$$

Therefore, the objective function for this case is,

$$\max z = \frac{\Sigma_{il}\left[Cc_{il}t_{il} + Cp_{il}n_{il}(1 - e^{-B_{il}t_{il}/n_{il}})\right]}{T_{\text{cycle}}} - \frac{\Sigma_{il}n_{il}\left[p_{il}\frac{t_{il}}{n_{il}} + q_{il}\left(\frac{t_{il}}{n_{il}}\right)^2\right]}{T_{\text{cycle}}} - \frac{\Sigma_{il}n_{il}Cs_{il}}{T_{\text{cycle}}}. \tag{A5}$$

The MINLP model (P3) for this case includes Eq. A5 as the objective function and Eqs. A1, 9, 10, 11, 12, 13, 14, 15, 16, and 17 as constraints. Theorems 1, 2, and 3 can be used to prove that the objective function is pseudoconcave, and that the righthand side of Eq. A1 is a concave function.

Appendix B: Model for the Case of an Unequal Cycle for Each Furnace

In this case, mass balance constraints (Eqs. 8 and 9) have to be modified to

$$F_i n_{il} T_{\text{cycle}_l} = D_{il}t_{il} \qquad \forall i,l \tag{B1}$$

$$\sum_i F_{in_{il}} = F_i \qquad \forall i. \tag{B2}$$

Here $F_{in_{il}}$ is the amount of feed i required per unit time for furnace l, and T_{cycle_l} is the cycle time of furnace l. Furthermore, the timing constraint (Eq. 14) for this case is as follows:

$$\sum_i \Delta t_{il} = T_{\text{cycle}_l} \qquad \forall l. \tag{B3}$$

Finally, the objective function is

$$\max \ z = \Sigma_l \frac{\Sigma_i\left[Cc_{il}t_{il} + Cp_{il}n_{il}(1 - e^{-b_{il}t_{il}/n_{il}}) - Cs_{il}n_{il}\right]}{T_{\text{cycle}_l}}. \tag{B4}$$

The MINLP model (P4) for this case includes Eq. B4 as objective function and Eqs. B1, B2, B3, 10, 11, 12, 14, 15, 16, and 17 as constraints. It should be noted that constraints are nonlinear and have nonconvexities in the form of bilinear terms. Furthermore, the objective function (Eq. B4) may not be pseudoconcave because the sum of pseudoconcave functions is not necessarily a pseudoconcave function.

Appendix C: On the Relation of Timings for Subcycles

For the derivation of the MINLP model it was implicitly assumed that the lengths of the subcycles for every feed on each furnace are equal. In this section, we present an analytical proof that shows that in an optimal schedule this must hold true.

Consider an optimal cyclic schedule for feed processing. Assume that the optimal schedule has n_{il} subcycles of feed i on furnace l and that the lengths of subcycles are different. Let us denote the length of subcycle j ($j = 1, \ldots, n_{il}$) by t_{ijl}. The summation of t_{ijl} over j gives the processing time of feed i on furnace l (t_{il}). For this case the objective function can be written as

$$z = \sum_i \sum_l \frac{\Sigma_{j=1}^{n_{il}}Cc_{il}t_{ijl} + \Sigma_{j=1}^{n_{il}}Cp_{il}(1 - e^{-b_{il}t_{ijl}}) - Cs_{il}n_{il}}{T_{\text{cycle}}}. \tag{C1}$$

We prove below that in an optimal solution, the length of subcycles for a particular feed on a particular furnace are equal, that is, $t_{ijl} = t_{ij'l}$ ($\forall j \neq j'$).

First, fix T_{cycle}, t_{il}, and n_{il} to the optimal values and allow t_{ijl} to be a variable. As a result, irrespective of the value of t_{ijl} all constraints of the optimization problem are satisfied except for the following additional constraint:

$$t_{il} = \sum_{j=1}^{n_{il}} t_{ijl} \qquad \forall i,l.$$

However, we can make the following substitution in the objective function.

$$t_{in_{il}l} = t_{il} - \sum_{j=1}^{n_{il}-1} t_{ijl} \qquad \forall i,l.$$

Hence, the objective function can be rewritten as

$$z = \sum_i \sum_l \frac{Cc_{il}t_{il} + \Sigma_{j=1}^{n_{il}-1}Cp_{il}(1 - e^{-b_{il}t_{ijl}}) + Cp_{il}[1 - e^{-b_{il}(t_{il} - \Sigma_{j=1}^{n_{il}-1}t_{ijl})}] - Cs_{il}n_{il}}{T_{\text{cycle}}},$$

with which the optimization problem can be treated as an unconstrained optimization problem. The necessary and sufficient condition for optimality, given the convexity of z for fixed T_{cycle}, is attained by setting the gradient of the objective function equal to zero,

$$\frac{\partial z}{\partial t_{i1l}} = \frac{\partial z}{\partial t_{i2l}} = \cdots = \frac{\partial z}{\partial t_{i(n_{il}-1)l}} = 0 \qquad \forall i, l$$

$$\Rightarrow \frac{Cp_{il} b_{il} [e^{-b_{il} t_{ij'l}} - e^{-b_{il}(t_{il} - \Sigma_{j=1}^{n_{il}-1} t_{ijl})}]}{T_{\text{cycle}}} = 0$$

$$\forall i, l, j' = 1, \ldots, (n_{il}-1)$$

$$\Rightarrow e^{-b_{il} t_{ij'l}} = e^{-b_{il}(t_{il} - \Sigma_{j=1}^{n_{il}-1} t_{ijl})}$$

$$\forall i, l, j' = 1, \ldots, (n_{il}-1)$$

$$\Rightarrow t_{ij'l} = t_{il} - \sum_{j=1}^{n_{il}-1} t_{ijl} \qquad \forall i, l, j' = 1, \ldots, (n_{il}-1)$$

$$\Rightarrow t_{i1l} + \cdots + 2t_{ij'l} + \cdots + t_{i(n_{il}-1)l} = t_{il}$$

$$\forall i, l, j' = 1, \ldots, (n_{il}-1)$$

$$\Rightarrow t_{ij'l} = t_{in_{il}l} \qquad \forall i, l, j' = 1, \ldots, (n_{il}-1).$$

Hence, for optimality, the length of the subcycles for every feed on each furnace must be equal.

Appendix D: Proof of Theorem 1

(a) Let (v^1, λ^1) and (v^2, λ^2) be any two points in the set $\tilde{C}$, and $0 \le \alpha \le 1$. First we assume that $\lambda^1 > 0$ and $\lambda^2 > 0$. We have

$$h(\alpha v^1 + (1-\alpha)v^2, \quad \alpha\lambda^1 + (1-\alpha)\lambda^2)$$

$$= (\alpha\lambda^1 + (1-\alpha)\lambda^2) g\left(\frac{\alpha v^1 + (1-\alpha)v^2}{\alpha\lambda^1 + (1-\alpha)\lambda^2}\right)$$

$$= (\alpha\lambda^1 + (1-\alpha)\lambda^2) g\left(\frac{\alpha\lambda^1}{\alpha\lambda^1 + (1-\alpha)\lambda^2}\frac{v^1}{\lambda^1} + \frac{(1-\alpha)\lambda^2}{\alpha\lambda^1 + (1-\alpha)\lambda^2}\frac{v^2}{\lambda^2}\right)$$

$$\le (\alpha\lambda^1 + (1-\alpha)\lambda^2)\left[\frac{\alpha\lambda^1}{\alpha\lambda^1 + (1-\alpha)\lambda^2} g\left(\frac{v^1}{\lambda^1}\right) + \frac{(1-\alpha)\lambda^2}{\alpha\lambda^1 + (1-\alpha)\lambda^2} g\left(\frac{v^2}{\lambda^2}\right)\right]$$

$$= \alpha\lambda^1 g\left(\frac{v^1}{\lambda^1}\right) + (1-\alpha)\lambda^2 g\left(\frac{v^2}{\lambda^2}\right)$$

$$= \alpha h(v^1, \lambda^1) + (1-\alpha) h(v^2, \lambda^2).$$

The inequality in the preceding equation follows from the convexity of $g(\tilde{x})$. This proves the convexity of function $h(\tilde{x}, \lambda)$ for case (a).

(b) Now if $\lambda^1 = \lambda^2 = 0$, then there is nothing to be proved [since v^1 and v^2 must be zero and $h(0,0) = 0$]. Therefore, without loss of generality assume that $\lambda^1 = 0$ and $\lambda^2 > 0$. Then,

$$h[\alpha v^1 + (1-\alpha)v^2, \alpha\lambda^1 + (1-\alpha)\lambda^2]$$

$$= (1-\alpha)\lambda^2 g\left(\frac{v^2}{\lambda^2}\right)$$

$$= \alpha h(0,0) + (1-\alpha) h(v^2, \lambda^2).$$

Hence $h(\tilde{x}, \lambda)$ is a convex function in case (b) also. The boundedness of $h(\tilde{x}, \lambda)$ follows immediately because λ and $g(\tilde{x})$ are bounded over the feasible set. The proof is similar for the case when $g(\tilde{x})$ is concave.

Manuscript received Dec. 8, 1997, and revision received Apr. 13, 1998.