AIChE

A Pattern Matching Method for Large-Scale Multipurpose Process Scheduling

Yaohua He
Dept. of Industrial and Systems Engineering/Office of the Provost (PVO), National University of Singapore, Singapore 117576

Dept. of Chemical and Biomolecular Engineering, Hong Kong University of Science and Technology, Hong Kong, P. R. China

Chi-Wai Hui
Dept. of Chemical and Biomolecular Engineering, Hong Kong University of Science and Technology, Hong Kong, P. R. China

DOI 10.1002/aic.12302
Published online June 22, 2010 in Wiley Online Library (wileyonlinelibrary.com).

This article proposes a novel pattern matching method for the large-scale multipurpose process scheduling with variable or constant processing times. For the commonly used mathematical programming models, large-scale scheduling with long-time horizons implies a large number of binary variables and time sequence constraints, which makes the models intractable. Hence, decomposition and cyclic scheduling are often applied to such scheduling. In this work, a long-time horizon of scheduling is divided into two phases. Phase one is duplicated from a pattern schedule constructed according to the principle that crucial units work continuously, in parallel and/or with full load as possible, exclusive of time-consuming optimization. Phase two involves a small-size subproblem that can be optimized easily by a heuristic method. The computational effort of the proposed method does not increase with the problem size. The pattern schedule can be not only used for production/profit maximization but also for makespan estimation and minimization.
AIChE J, 57: 671–694, 2011
Keywords: pattern matching, scheduling, multipurpose batch plants, decomposition, heuristic, search tree

Introduction

Process industry shows considerable complexity compared to discrete parts manufacturing.[1] For instance, the complexity of scheduling batch processes is determined by such aspects as batch size constraints, shared intermediates, flexible proportions of input and output goods, carrying out processes without interruption, sequence- and usage-dependent cleaning operations, no-wait production for certain types of products, and so on. Consequently, most of the scheduling approaches developed for discrete parts manufacturing are hardly applicable.[2]

Research on batch and continuous process scheduling has received great attention from academia and industry in the past two decades. Although significant advances have been made, there are still a number of major challenges and questions that remain unresolved. One of the challenges is the large-scale industrial applications.[3] From the mathematical perspective, most scheduling problems found in industrial environments can be regarded as very large-scale combinatorial and complex optimization problems, which rarely can be solved to optimality within a reasonable amount of

Correspondence concerning this article should be addressed to Y. He at YaohuaHe2009@gmail.com.

computational time. Such a combinatorial explosiveness has to do with the increased number of products to be processed, the long sequence of processing stages, the multiple units available for each task and the length of the scheduling horizon (the case of the present work) to be considered. The complexity arises from a wide range of operational constraints that often need to be taken into account in real world problems.

General mathematical formulations for process scheduling are either based on the state task network (STN)[4] or resource task network (RTN)[5] process representations. STN- or RTN-based formulations can be applied to any network-represented processes from batch to continuous, consisting of fixed or variable duration tasks. The STN- or RTN-based MP models account for complex process network configurations (batch splitting/mixing and recycle streams), variable batch sizes, utility requirements, and various storage policies (NIS/FIS/UIS). The major difference between STN and RTN is that the STN treats equipment resources implicitly, while the RTN treats them explicitly. The objective functions are often the maximization of production/profit over a fixed time horizon, or the minimization of makespan with fixed product demand.

According to the time representation, the STN/RTN-based mathematical programming (MP) methods can be classified into discrete-time and continuous-time models. Earlier formulations use a discrete-time representation and may involve several thousands of binary variables to handle a sufficiently fine discretization that closely matches the exact problem data. Hence, when solving large-size problems, problem data are usually rounded to maintain problem tractability. In the discrete-time formulation proposed by Kondili et al.[4] and refined by Shah et al.,[6] the processing times of tasks are assumed to be constant, and the fixed time horizon was divided into intervals of known duration equal to the greatest common factor of the processing times of all tasks.

The assumption of constant processing times is not always realistic, and the length of the intervals can be so small that it either leads to a prohibitive number of intervals, resulting in the model unsolvable, or it requires approximations that might compromise the feasibility and optimality of the solution. This led several authors[7–14] to develop continuous-time STN/RTN formulations, where variable processing times were assumed and the time horizon was divided into time intervals of unequal and unknown duration. In continuous-time formulations, one needs to specify the number of time points that compose the time grids, depending on whether the formulation uses a single time grid,[15,16] or one for each equipment resource.[12] Although more realistic, the continuous-time models are hard to solve mainly because of their poor LP relaxation, which is due to the big-M time matching constraints.

With regard to the objective functions, it was noticed that both discrete-time and continuous-time STN formulations behave moderately well when the objective function is the maximization of production/profit over a fixed time horizon. For the minimization of the makespan, discrete STN formulations had never been used before Maravelias and Grossmann[15] did so, and continuous STN formulations behave poorly, as shown by Maravelias and Grossmann.[13]

The diverse continuous-time models can be classified into three types: slot-based, global event–based, and unit specific event–based formulations. Shaik et al.[17] compared and evaluated the performance of six such existing models, based on their own implementations using several benchmark problems from the literature, considering two different objective functions, profit maximization, and makespan minimization. It is notable that all such models only solved small-size problems within short-time horizons ($<$16 h) to optimality. However, for the large-size problems within longer time horizons, due to the large number of time grids involved, the model sizes are often too huge to get feasible solutions.

To solve the large-size problems, one choice is to apply decomposition approaches for near-optimal solutions. Decomposition has been long recognized as a fundamental technique in large-scale optimization. For the short-term scheduling problem, Bassett et al.[18] presented a number of time-based decomposition approaches based on a discrete-time formulation. As Wu and Ierapetritou[19] found that the mixed integer linear programming (MILP) model proposed by Ierapetritou and Floudas[9] was difficult to solve the problems with long-time horizons, they proposed several decomposition approaches.

Another choice is to use cyclic (or periodic) scheduling methods. Cyclic scheduling has been developed to make the operation decisions easier and profitable. It constructs an operation schedule to be executed repeatedly. In addition to the advantage of easy management and control of the plant, mathematically the problem is limited to a smaller time horizon and thus can be solved more efficiently. Shah et al.[6] modified the formulation of Kondili et al.[4] and extended it to the periodic scheduling of batch plants using a discrete time representation. Schilling and Pantelides[20] presented a periodic scheduling formulation which is based on their earlier work on continuous-time representation. Because of the difficulty in linearizing the nonlinear function, a branch-and-bound algorithm that branches on both discrete and continuous variables was proposed. Castro et al.[21] modified their early RTN-based formulation to fit periodic scheduling requirement for an industrial application. Then, Wu and Ierapetritou[22] extended the work by Ierapetritou and Floudas[9] based on the STN representation and developed the cyclic scheduling formulation with the advantage of using few binary variables.

The idea of cyclic scheduling usually overlooks the start-up and finishing phases of a schedule. However, to obtain a feasible schedule for real use, especially when long-time horizons are considered, which make it a challenge to achieve such a solution, detailed operation of start-up and finishing phases should be considered. Wu and Ierapetritou[22] proposed a cyclic scheduling approach to yield a complete schedule with a detailed account for start-up and finishing phases.

In fact, cyclic scheduling is still a decomposition technique. What should be noticed is that in cyclic scheduling the objective function may turn into a nonlinear equation which results in a mixed integer nonlinear programming (MINLP) model and increases the difficulty to solve the problem. Decomposition techniques are problem dependent and no underlying general theory has evolved.[23] The drawback of decomposition is that the global optimum may never be found despite the possible near-optimal solutions. So decomposition is a compromise under the current technological condition.

From the literature review, it is noted that no matter how long the time horizons are, MP (MILP or MINLP) models are needed for an optimum or a near-optimal solution. In such models, the binary variables and time sequence constraints increase with the extension of the scheduling time horizon. Consequently, an intractable problem may arise due to the heavy computational burden required. Although the decomposition techniques and cyclic scheduling methods divide the overall problem into several subproblems hence reduce the computational time in more than one order of magnitude, the total computational time still increases with the overall problem size. Furthermore, in cyclic scheduling, to achieve an optimal cycle time and a maximum average profit/production over the cycle time, one has to establish an MINLP model that is more difficult to be solved than an MILP model. To avoid these defects with the traditional mathematical methods, this article proposes a novel pattern matching method for the large-scale problems, especially for those with long-time horizons.

In another work by He and Hui[24] developing a genetic algorithm to evolve the task sequences in scheduling a multipurpose batch plant with constant processing times, we found that some equipment resources (or units) are crucial factors to minimize makespan with the fixed product demand or to maximize the production/profit within a fixed time horizon. These crucial units should work with full load and continuously as possible. Given a longer time horizon to find good task sequences to be executed on the crucial units, periodicity is discovered with the benchmark batch plant. Inspired by this example, it seems that periodicity may generally exist in other multipurpose batch plants. The periodicity can be utilized for long-horizon scheduling.

In this work to solve the long-horizon scheduling problems, a pattern schedule is first established based on the periodicity discovered through analyzing the recipe of the plant, and then the long time horizon of an overall problem is cut into two phases. The subschedule in phase one is copied from the pattern schedule. The subschedule in phase two is obtained by solving a small-size subproblem. Such a novel decomposition method avoids tedious long-time computation as MP models do, and can be easily applied to real-world plants. Case study shows that the proposed method performs better than the three-phase method[22] based on classical cyclic scheduling.

General Solution Strategy

Solution framework

The purpose of this work is to come up with a practical scheduling method for the large-scale multipurpose process scheduling that is intractable for the existing exact models. Even with decomposition or cyclic scheduling techniques, such scheduling suffers subopimality and still very long computational time; if start-up and finshing phases are considered, the solution process becomes more time-consuming and instance-dependent. The novel decomposition method proposed here is more efficient and easier for real-world applications, because it does not require a commercial solver necessary for MP models, or complicated programming as needed in metaheuristic methods.

The proposed approach is a two-phase mehod where the overall time horizon for scheduling is divided into two phases. The schedule of the first phase can be copied from a pattern schedule, which results in a great reduce of computational time. The schedule of the second phase can be obtained by solving a small-size subproblem via the existing MILP models or the proposed heuristic method in this work. Figure 1 shows the decomposition of a time horizon (denoted by TH) of a schedule. The solution framework for the large-scale scheduling can be described as follows:

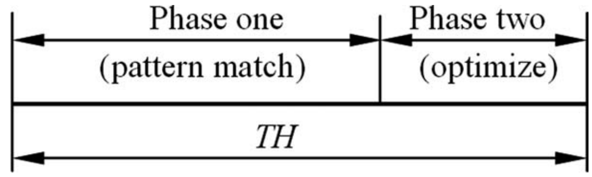

Figure 1. Decomposition of a long-time horizon.

(1) Construct a pattern schedule corresponding to the scheduling objectives. To establish the pattern schedule, the natural cycle, instead of the exact cycle as usual, is utilized in this work.

(2) Select a proper cut point between the two phases. This cut point makes the second phase have a enough length for a good solution and involve a small-size subproblem to be solved easily.

(3) To avoid a large number of binary vaiables and time sequence constraints, we use a heuristic method to solve the small-size subproblems.

Master/slave task sequences and bottleneck units

In a multipurpose batch plant, multiple products are produced by using a number of shared production units that constrain the plant operation. For the general resource-constrained scheduling of multipurpose batch plants, we could first outline a number of task sequences, each of which is performed by one or more units. Some task sequences play a crucial role in maximizing production within a given period of time or minimizing makespan with a fixed product demand, thus defined as master task sequences. Other task sequences are not so significant as to affect the objective function value, hence called slave task sequences.

Another important issue is to identify the bottleneck units (or crucial units, key units), which usually have to work continuously, in parallel and/or with full load as possible. Through analyzing the recipe of the process expressed by STN, RTN or recipe diagram (RD[14]), it should be easy for a scheduler to find out the bottleneck units by using the following general rules: (1) units in the master task sequences; (2) units which are shared to execute more than two different tasks; (3) units which carry out a single task multiple times, e.g. to produce an intermediate that is consumed by most other tasks; and (4) units used to consume the material in a tank with limited capacity. Whether a unit is crucial or not depends on whether the unit has enough time or capability to handle all the tasks in the natural cycle of a pattern schedule.

Natural periodicity and pattern scheduling

The periodicity of production is referred to as the property of an operation cycle which is repeated during the production process. If a batch plant is operated under conditions of relatively stable product demands over long periods of time, often covering several months, "it is often profitable to establish a regular periodic operating schedule in which the same sequence of operations is carried out repeatedly. In particular, the use of an operating cycle of relatively short duration (the 'cycle time'), typically of the order of days, greatly simplifies the operation and control of the plant. The goal of periodic (or 'cyclic') scheduling is to optimize the utilization of resources over a cycle by ignoring the start-up and shut-down phases of the production. In general, the optimal cycle time is also to be determined".[20] Therefore, in general, cyclic scheduling is to identify a unit schedule with an optimal cycle that is desired to be relatively short or of the order of days.

In the current literature, a rigorous scheduling cycle involves the same sequence of operations, and the identical initial and end intermediates (i.e., states of intermediate tanks). Under this rigorous concept, the existing mathematical methods often claimed an optimal cycle length with the accuracy of the order of seconds. But we have to argue: Does such an exact cycle length really "simplify the operation and control of the plant"? Furthermore, in the real industry, the optimal cycle length may be often intractable with the existing models.

As the product demands over a period of several months is relatively stable, why should the plant be operated by using a "factitious" cyclic schedule? It does not guarantee global optimality due to subproblems solved by decomposition. Can a "natural" cycle be repeated to satisfy the optimization purpose? In fact, such natural periodicity exists in the batch plants. A natural cycle can be calculated from the process recipe according to mass balance.

There are two ways to find the natural periodicity. One way is the manual assignment trying to let the task sequences have the maximum production/profit. When the relevant intermediate tank(s) return to initial material state (e.g., become empty again), a natural cycle emerges. Apparently, such a cycle can be performed repeatedly with the approximately maximum production/profit. The other way is the common factor calculation, where as the intermediate materials, produced by one or more tasks, have completely consumed by other tasks, a natural cycle also comes out.

It is notable that the natural periodicity depends on the prevailing plant conditions. For instance, the capacities of intermediate tanks influence the periodicity (see Example 1), the recycle stream is another impact factor. Besides, the length of the processing times is certainly affects the classification of master/slave task sequences and consequently the natural cycle length.

With the natural periodity determined, a pattern schedule can be easily constructed through manual operation or automatical computation, covering a time peroid as long as desired to match the scheduling time horizons. Such a pattern schedule, once established, can be repeatedly used for cutting and copying the first phase of a long-horizon schedule. The general aspects to construct a pattern schedule for the multipurpose process can be summarized as follows:

(1) Formulate the intermediate state consumption and replenishment (SCR) equations according to the mass balance coefficients given in the STN diagram. States S_{k+1} are calculated from the proceeding slot k (such as Eq. 2–4 for Example 1). Simultaneously consider the vessels' capacity constraints.

(2) Identify the master task sequences and the bottleneck processing units through simple calculations in the analysis of the natural cycle, where $S_{N+1} = S_1$, (some) intermediates return to initial states through a natural cycle with N slots.

(3) Arrange the batches into sequences which constitute a pattern schedule. The batches are assigned at slot k by checking the current states S_k, where problem knowledge are applied.

(4) Heuristics to make full use of capability of the bottleneck units.

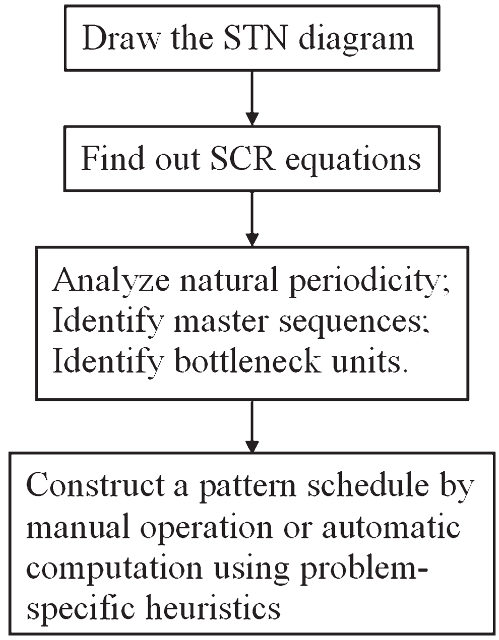

Figure 2. Steps for a pattern schedule.

Figure 2, based on the above general points, shows the basic steps to establish a pattern schedule. And the solution procedure for each subsequent example follows this flowchart.

General heuristics for task assignment

With the key units identified, heuristic rules associated with the objective function are to be used for appropriate task assignment. Such rules are case specific, but general heuristics can be summarized as follows: (1) Certain units work with full load as possible; (2) certain units perform tasks as continuously as possible; (3) some units must work in parallel; (4) some tasks must be carried out in advance to prepare enough intermediate materials to be consumed by the following tasks; and (5) some tasks must be performed in time to let the relevant tanks have enough space to store the materials produced by other tasks. Anyway, the precondition to assign a batch of a task is that the relevant material states (and utilities) are ready for this batch.

Decomposition based on a pattern schedule

With the pattern schedule available, we have already understood quite some problem knowledge, and we can

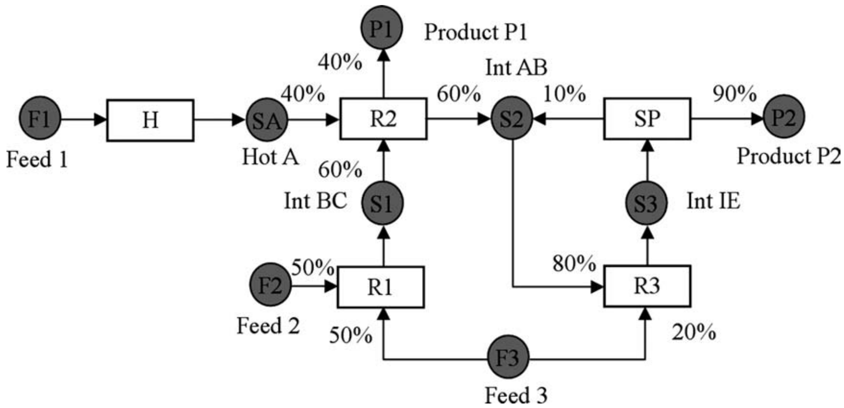

Figure 3. State-task network of Example 1.

estimate the total production/profit within the given TH. The purpose of phase 2 is to further maximize the total production/profit, so phase 2 should have enough slots for the arrangement of appropriate end batches. So the selection of the cut point should not be subjective or arbitrary. The proper number of batches in phase 2 is certainly determined by our problem knowledge but most of which is accumulated in our mind when constructing the pattern schedule.

In dividing the long time horizon, the following factors should be considered: (1) phase 1 is usually much longer so as to take the advantage of the pattern schedule; (2) phase 1 ends with full batches of tasks for final products, unless to consider (3); (3) phase 2 involves enough slots to have at least a feasible solution to wrap up the intermediates, or to ensure further optimization; and (4) phase 2 includes a small number of time slots to make the sub-problem be solved easily. With these considerations, it is easy to determine the cut point, not necessarily to use trial-and-error.

A Motivating Example

Figure 3 shows the STN representation of Example 1, a typical batch plant with splitting, merging and recycling streams, where two final products are produced from three raw materials through heating, three reactions and separation, involving four intermediate materials. Table A1 presents the state data of the plant, including the capacity (C_s), the initial states ($S0_s$) in the tanks, and the price (PR_s) of the materials. Table A2 presents the task data of the plant with constant processing times (CPT, denoted by τ_j) and variable processing times (VPT, denoted by T_j). T_j is calculated by the formula:

$$T_j = \alpha_j + \beta_j \cdot B_j \tag{1}$$

where B_j is the batch size of task j. According to the batch size limitation, the ranges the processing times of the tasks can be calculated, as shown in Table A2. The objective is to maximize the total production/profit of the plant within a given scheduling time horizon or minimize the makespan with fixed product demand.

This example is well known and extensively studied. Some researchers focused on the problem with CPT,[4,6,15] and others on the case of VPT.[7–14] Up to now, almost all of the researchers have used mathematical programming models for solving such problems. An MILP model based on STN[6,15] normally involves not only the variables and constraints to bound the batch sizes of tasks, to preserve mass balance for the states and to impose bounds on the material states, but also a series of other variables (including a large number of binary variables) and constraints to reflect the logical relationships of the recipe.

In the continuous-time STN/RTN formulations for the VPT case, the time horizon of the schedule was divided into time slots of unequal and unknown duration. Most relevant literature presented the results of small-size instances within time horizons of 8 and 12 h. However, in practice, we have to consider long time horizons, e.g., from 1 to 7 days (from 24 to 168 h). Usually, these large-size instances lead to exponential growth of solution times.

Ierapetritou and Floudas[9] proposed an efficient MILP model for solving small-size instances (e.g. 8 or 12 h), but as stated by Wu and Ierapetritou,[22] when the same formulation was used for a time horizon of 168 h, a feasible schedule cannot be obtained for the whole time horizon. Maravelias and Grossmann[15] also came up with an MILP model for this kind of problems, still presenting results for small-size instances. To solve large-size instances, on the basis of the work of Ierapetritou and Floudas,[9] Wu and Ierapetritou[19] first proposed decomposition approaches that solved the instances with time horizons of 16, 24, and 48 h. Subsequently, Wu and Ierapetritou[22] proposed a cyclic scheduling model for the middle phase and considered start-up and finishing phases for a complete feasible schedule. The overall time horizon is divided into three phases: the initial phase when the necessary amounts of intermediates are prepared to start the cyclic schedule, the main phase when cyclic scheduling is applied, and the final phase to wrap up all the intermediates. Each phase involves a different MP model which still needs long CPU time for a final solution.

In this study, motivated by Example 1, with the purpose to greatly save computational load in solving large-size instances over time horizons from 24 h (1 day) to 168 h (7

Table 1. Notations for a Schedule in Example 1

k	Time slots, $k = 1, 2, \ldots, N$
TS	A task sequence of R1, R2, and R3
H_k	Batches processed by the heater
RI_j_k	Batches of reaction j on RI, $j = 1, 2, 3$
RT_j_k	Batches of reaction j on RT, $j = 1, 2, 3$
FLT_k	Batches processed by the filter
$(S1_k, S2_k, S3_k)$	States of tanks BC, AB and IE
AP_k	Accumulative production
T_k	Maximum process time of slot k
AT_k	Accumulative process time

days), a pattern matching method and a novel decomposition approach are proposed.

Pattern Scheduling for the Motivating Example

SCR equations

To express a schedule of Example 1, some indices and variables are needed. Table 1 presents the notations for a schedule, where k is an integer index for time slots, and TS is a task sequence comprising R1, R2, and R3. In a pattern schedule, one reaction task (R1, R2, or R3) is conducted by both RI and RT in parallel (synchronously). The length of a time slot (T_k) depends on the batch size of a task assigned.

From the STN diagram and the problem data of Example 1, the intermediate state consumption and replenishment equations and the inequalities for capacity limitations can be given as:

$$S1_{k+1} = S1_k + 1.0(RI\_1_k + RT\_1_k) - 0.6(RI\_2_k + RT\_2_k) \quad (2)$$

$$S2_{k+1} = S2_k + 0.6(RI\_2_k + RT\_2_k) - 0.1FLT_k - 0.8(RI\_3_k + RT\_3_k) \quad (3)$$

$$S3_{k+1} = S3_k + 1.0(RI\_3_k + RT\_3_k) - 1.0FLT_k \quad (4)$$

$$S1_k \le C_{BC}, \quad S2_k \le C_{AB}, \quad S3_k \le C_{IE} \quad (5)$$

where C_{BC}, C_{AB}, and C_{IE} are the capacities of tanks BC, AB, and IE, respectively. Eq. 2–4 are used for calculating the states at the beginning of slot ($k + 1$).

Natural periodicity analysis

Master/Slave Task Sequences and Crucial Units. From the STN diagram and the problem data of Example 1, it can be seen that (1) the heater (H) performs only one task—heating, (2) the filter (FLT) also performs only one task—separation; however, (3) Reactor I (RI) and Reactor II (RT) have to perform three tasks—Reactions 1, 2, and 3 (R1, R2, and R3). The task of heating has to be performed right before R2, SP tasks need to carried out as required by R3 considering the limited capacity of the tank IE. The heater and the filter work occasionally as required, a must for production, but not affecting the makespan of the production, because they have enough processing capability. To minimize the makespan of the production (or the schedule) or maximize the total production within a certain time horizon, RI and RT must cowork simultaneously. That is to say, if one of the reactors stops working, the production can still continue, but the makespan must be longer, or the total amount of production within a fixed time horizon must be less. Therefore, RI and RT are two crucial units in Example 1.

A further analysis reveals that the task sequence of R1, R2, and R3 performed by RI and RT is the master sequence, which determines the natural periodicity, the total production, or the time length of the pattern schedule. However, the task sequence of heating and the task sequence of separation are two slave sequences, which provide necessary hot material for R2, or allow tank IE to have enough space to store the product from R3. The heater and the filter do not necessarily work continuously or with full load due to their sufficient processing capabilities.

Natural Periodicity Analysis. In Example 1, to maximize the total production, the two crucial units (RI and RT) should run continuously with full load (or full batches). Under this assumption, a natural cycle can be calculated according to the mass balance requirements given by process recipe. Assume that P_{R1}, P_{R2}, P_{R3}, and P_{FLT} are the total amounts of material produced by tasks R1, R2, R3, and SP, respectively, in a cycle. RI and RT perform the same reaction in parallel and with full load, so a parallel full batch is $80 + 50 = 130$ kg. The initial states $S1_k = 0$, $S2_k = 0$, and $S3_k = 0$, expressed as (0, 0, 0), where $k = 1$. At the end of a cycle, try to let these states return to zero again. So according to the process recipe, the following equations exist:

$$P_{R1} = 0.6P_{R2} \quad (6)$$

$$0.6P_{R2} + 0.1P_{FLT} = 0.8P_{R3} \quad (7)$$

$$P_{R3} = P_{FLT} \quad (8)$$

Hence, $P_{R1}/P_{R2} = 3/5$, $3P_{R2} + 1/2P_{FLT} = 4P_{R3}$, $P_{R3} = P_{FLT}$. That means three parallel full batches (3 × 130 kg) of R1 require five parallel full batches (5 × 130 kg) of R2 to make $S1_k = 0$ at the end of the cycle; and consequently require more than four parallel full batches (4 × 130 kg) of R3 to make $S2_k = 0$ at the end of the cycle; and all the intermediate from R3 has to pass through the filter for separation, thus $S3_k = 0$ at the end of the cycle.

Up to now, it can be estimated that, if a cycle involves three parallel full batches (3 × 130 kg) of R1, five parallel full batches (5 × 130 kg) of R2, and four parallel full batches (4 × 130 kg) of R3, the state $S1_k$ is sure to be zero. After such a cycle, $S2_k$ and $S3_k$ may not return to zero. This cycle is called a nonideal natural cycle considering nonzero $S2_k$ and $S3_k$ at the end of the cycle. The length of such a cycle is (3 × 2.666 + 5 × 2.666 + 4 × 1.333) = 26.660 (h). If the Eq. 7 is changed as:

$$0.6P_{R2} = 0.8P_{R3}, \quad (9)$$

which means no recycle from the filter (FLT), then $P_{R2}/P_{R3} = 4/3$. From the equations $P_{R1}/P_{R2} = 3/5$ and $P_{R2}/P_{R3} = 4/3$, it can be deduced that a cycle involves 12 parallel full batches (12 × 130 kg) of R1, 20 parallel full batches (20 × 130 kg) of R2, and 15 parallel full batches (15 × 130 kg) of R3, the states $S1_k$ and $S2_k$ are sure to be zero. If the filter is removed from the

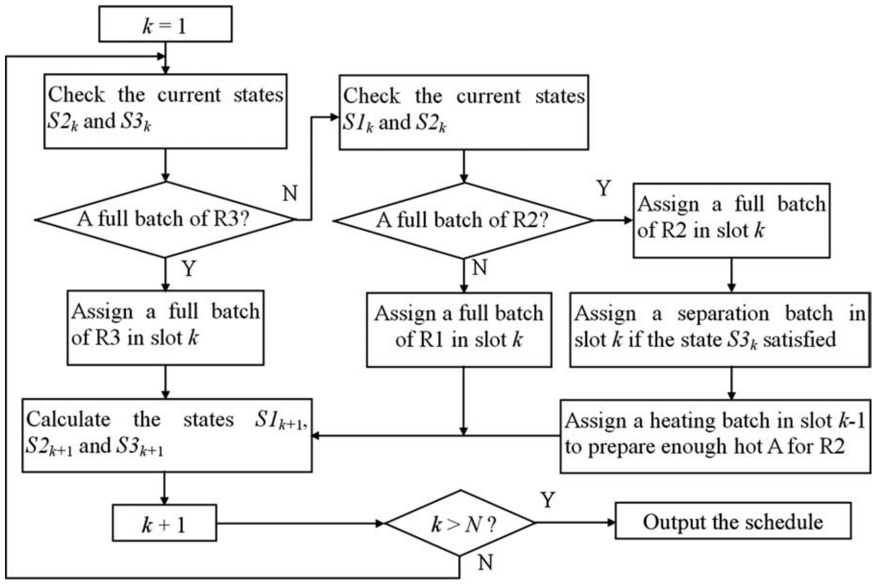

Figure 4. Flow chart for a pattern schedule with maximum production in Example 1.

process recipe, then the tank IE can also be removed. With all these assumptions, a complete (or ideal) natural cycle is achieved with zero intermediate states at the end of the cycle, the duration of which is $(12 \times 2.666 + 20 \times 2.666 + 15 \times 1.333) = 105.307$ (h). This is the case of Example 2 in Chapter 8 of He's PhD dissertation[25] (or the book by He and Hui[26]).

The nonideal/ideal natural cycles in Example 1 can be utilized for decomposition of a long time horizon. Subsequently, one pattern schedule consisting of nonideal natural cycles and another pattern schedule with repeated identical task sequences are to be constructed for Example 1.

Two pattern schedules

Heuristics for Task Assignment in Example 1. After the identification of the master task sequence and the bottleneck units, the problem knowledge and heuristics to make full use of the capabilities of the bottleneck units are to be utilized for task assignment. For each time slot, when assigning two reaction batches to RI and RT, a heating batch to the heater or a separation batch to the filter, the following heuristics should be considered: (1) RI and RT perform the same reaction (R1, R2, or R3) in parallel and with full load. (2) The filter performs a separation batch in time in parallel with R1 or R2 by checking the current state $S3_k$, because a full separation batch has the same processing time as a full batch of R1 or R2. (3) A full batch of R3 is performed by checking the current states $S2_k$ and $S3_k$. (4) A full batch of R2 is performed by checking the current states $S1_k$ and $S2_k$. (5) If neither a full batch of R3 nor a full batch of R2 is satisfied, then a full batch of R1 is executed. (6) The heater performs a heating batch right before R2 in time to prepare enough hot material for the coming R2.

Figure 4 shows the flow chart to construct a pattern schedule with maximum production, where R3 is first considered and then R2 is followed by R1. The rationale for this flow chart is that R3 consumes S2 in time to let tank AB have enough space for the intermediate produced by R2, and R2 consumes S1 for the similar purpose. If both R3 and R2 are not satisfied, R1 should be executed to prepare enough S1 for coming R2.

Table 2 presents part of the pattern schedule by manual operation. For $k = 1$ in Table 2, the initial states $S1_1 = 0$, $S2_1 = 0$, and $S3_1 = 0$, expressed as (0, 0, 0), a parallel full batch of R1 is performed by both RI and RT; for $k = 2$, a parallel full batch of R2 is satisfied, and so on. What should be specified here is that when R2 follows R1, 60% of the intermediate product of R1 needs not to be stored in the tank BC, just directly for R2 in the reactors. So $S1_2 = 52 + (78)$ means that 78 kg intermediate product of R1 retains in the reactors for R2. Similarly, when R3 follows R2, 60% of the intermediate product of R2 needs not to be stored in the tank AB, just directly for R3 in the reactors. So $S2_9 = 78 + (78)$ means that 78 kg intermediate product of R2 retains in the reactors for R3. At slot $k = 12$, the states $S1_{12}$ and $S3_{12}$ returns to zero, a parallel full batch of R3 is satisfied after checking the current states $S2_{12}$ and $S3_{12}$.

Pattern Schedule I. Actually, according to the procedure in Figure 4, a pattern schedule can be constructed through automatic computation. The partial schedule with 12 slots, as shown in Table 2, has five batches of H, three batches of R1, five batches of R2, four batches of R3, and three batches of FLT and a total processing time of 20 h for CPT or 26.660 h for VPT. As the tasks are assigned onward according to the procedure given in Figure 4, until 72 time slots (159.96 h $\approx$ 160 h), a pattern schedule as shown in Table 3 has been obtained. Table 4 presents the statistic data of the periodicity of the pattern schedule comprising six nonideal cycles, where it can be seen that: (1) each cycle has the same numbers of time slots; (2) each cycle has the same numbers of batches of heating, batches of R1 (parallel on RI and RT), batches of R2 (parallel on RI and RT), and batches of R3 (parallel on RI and RT); (3) each cycle has the same total process time (20 h for CPT, 26.660 h for VPT); (4)

Table 2. A Schedule with 12 Time Slots

k	1	2	3	4	5	6	7	8	9	10	11	12
TS	R1	R2	R1	R2	R3	R2	R3	R1	R2	R3	R2	R3
H	52		52		52			52		52		
RI_1	50		50					50				
RI_2		50		50		50			50		50	
RI_3					50		50			50		50
RT_1	80		80					80				
RT_2		80		80		80			80		80	
RT_3					80		80			80		80
FLT						130			130		130	
*S*1	0	**52(+78)**	52	**104(+78)**	104	104	26	26	**78(+78)**	78	78	0
*S*2	0	0	78	78	156	52	143	39	39	130	26	117
*S*3	0	0	0	0	0	130	0	130	130	0	130	0
AP	0	0	52	52	104	104	273	273	273	442	442	611
(*T*)[a]	2	2	2	2	1	2	1	2	2	1	2	1
(AT)[a]	2	4	6	8	9	11	12	14	16	17	19	20
T	2.666	2.666	2.666	2.666	1.333	2.666	1.333	2.666	2.666	1.333	2.666	1.333
AT	2.666	5.332	7.998	10.664	11.997	14.663	15.996	18.662	21.328	22.661	25.327	26.660

[a]For the case of constant processing times (CPT).

cycles 1 and 5 have different numbers of batches of SP in the filter, and thus the total production of each cycle is, respectively, different (but the filter is not a crucial unit in this example).

The pattern schedule is a feasible schedule in which the two key units (RI and RT) work continuously with full load. Because every task is assigned according to the current states, the task sequences in the six cycles are not identical. From Table 3, it can be noticed that after every cycle the state S1 returns to zero but the state S2 has an increase due to the 10% recycle from every batch of the separation. It can be estimated that if the cycle with the same numbers of batches of heating, batches of R1 (parallel on RI and RT), batches of R2 (parallel on RI and RT) and batches of R3 (parallel on RI and RT) is repeated, then after a certain future cycle the state *S*2 will exceed the capacity of tank AB. Therefore, when *S*2 is accumulated to a big value after a certain cycle, a special treatment should be performed: add one parallel full batch of R3 to consume the intermediate of *S*2 before the subsequent cycle and make *S*2 return to a small value. After such a special treatment, several cycles can be repeated.

Pattern Schedule II. One may argue that the pattern schedule in Table 3 does not consist of ideal natural cycles or at least the task sequences of cycles are not identical. In fact, a pattern schedule with repeated identical task sequences can be achieved as shown in Table 5 (cycles 1 and 2 are the same as cycles 1 and 2 in Table 3). In Table 5, it can be found that from the slot $k = 63$ to $k = 65$ the states *S*2 exceed 200, which means the capacity of tank AB is required to increase. However, in pattern schedule I, all states do not exceed the capacities of the corresponding tanks.

Heuristic Method for Small-Size Instances in Example 1

In this section, with the assumption of variable processing times, small instances with up to seven time slots and with the zero initial intermediate states are solved by a heuristic method. In the existing MILP models, integer (binary) variables are used for the optimization of task combinations and continuous variables for the optimization of batch sizes. In this work, we first use heuristics and search tree to identify a proper task sequence and then use a solver to optimize the batch sizes of the tasks, thus avoid using binary variables.

Task sequences based on heuristics and search tree

Search trees are used for selecting a task sequence from possible sequences. The first node of the search tree is determined by the initial states ($S1_1$, $S2_1$, $S3_1$). The subsequent nodes are decided by heuristics underlying the process recipe.

Figure 5 shows a sample search tree for a four-slot schedule, where the initial states ($S1_1$, $S2_1$, $S3_1$) = (0, 0, 0). According to the initial states, the first node is R1. To maximize the total production within the limited time horizon, we should increase the product P1 from R2 and the product P2 from the filter (FLT). The following heuristics are used to determine the task sequence: (1) try best to assign R3 into the sequence, and R3 is not at the end of the sequence so as to be followed by a separation batch; (2) R1 is not at the end of the sequence; and (3) one batch of R2 must follow R1 according to process recipe, and the last slot is desired to be R2 to get more product of P1.

So the task sequence is determined as "R1|R2|R3|R2." On the basis of the current task sequence, for each time slot, a further tuning can be carried out as required with the purpose to increase the total production.

After the task sequence has been determined, a schedule with possible full load can be obtained under the consideration of the least intermediates at the end of the schedule. Table 6 presents a four-slot schedule with possible full load. This schedule has a total production of 174.417 kg completed within 8.720 h. If all the batch sizes are set to be close to zero, the total processing time is 4.669 h. So this task sequence covers a time range of [4.669, 8.720], and the time horizons 6, 7, and 8h are included in this time range. Similarly, the five-, six-, or seven-slot task sequences can be identified and their time ranges can be determined according to the corresponding schedule with possible full load (see Table 8).

Table 3. Six Cycles of Pattern Schedule I within 160 h (VPT) or 120 h (CPT) in Example 1

	Cycle 1												Cycle 2											
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
TS	R1	R2	R1	R2	R3	R2	R3	R1	R2	R3	R2	R3	R1	R2	R1	R2	R3	R2	R3	R1	R2	R3	R2	R3
H	52		52		52			52		52			52		52		52			52		52		
RI	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
RT	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80
FLT						130			130		130			130				130			130		130	
S1	0	52(+78)	52	104(+78)	104	104	26	26	78(+78)	78	78	0	0	52(+78)	52	104(+78)	104	104	26	26	78(+78)	78	78	0
S2	0	0	78	78	156	52	143	39	39	130	26	117	13	13	104	104	182	78	169	65	65	156	52	143
S3	0	0	0	0	0	130	0	130	130	0	130	0	130	130	0	0	0	130	0	130	130	0	130	0
AP	0	0	52	52	104	104	273	273	273	442	442	611	611	611	780	780	832	832	1001	1001	1001	1170	1170	1339
(T)	2	2	2	2	1	2	1	2	2	1	2	1	2	2	2	2	1	2	1	2	2	1	2	1
(AT)	2	4	6	8	9	11	12	14	16	17	19	20	22	24	26	28	29	31	32	34	36	37	39	40
T	2.666	2.666	2.666	2.666	1.333	2.666	1.333	2.666	2.666	1.333	2.666	1.333	2.666	2.666	2.666	2.666	1.333	2.666	1.333	2.666	2.666	1.333	2.666	1.333
AT	2.666	5.332	7.998	10.664	11.997	14.663	15.996	18.662	21.328	22.661	25.327	26.660	29.326	31.992	34.658	37.324	38.657	41.323	42.656	45.322	47.988	49.321	51.987	53.320
	Cycle 3												Cycle 4											
k	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
TS	R1	R2	R3	R1	R2	R3	R2	R3	R1	R2	R2	R3	R1	R2	R3	R1	R2	R3	R2	R3	R1	R2	R2	R3
H	52			52		52			52	52			52			52		52			52	52		
RI	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
RT	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80
FLT		130			130		130				130			130			130		130				130	
S1	0	52(+78)	52	52	104(+78)	104	104	26	26	78(+78)	78	0	0	52(+78)	52	52	104(+78)	104	104	26	26	78(+78)	78	0
S2	39	39	130	26	26	117	13	104	0	0	78	169	65	65	156	52	52	143	39	130	26	26	104	195
S3	130	130	0	130	130	0	130	0	130	130	130	0	130	130	0	130	130	0	130	0	130	130	130	0
AP	1339	1339	1508	1508	1508	1677	1677	1846	1846	1846	1898	2067	2067	2067	2236	2236	2236	2405	2405	2574	2574	2574	2626	2795
(T)	2	2	1	2	2	1	2	1	2	2	2	1	2	2	1	2	2	1	2	1	2	2	2	1
(AT)	42	44	45	47	49	50	52	53	55	57	59	60	62	64	65	67	69	70	72	73	75	77	79	80
T	2.666	2.666	1.333	2.666	2.666	1.333	2.666	1.333	2.666	2.666	2.666	1.333	2.666	2.666	1.333	2.666	2.666	1.333	2.666	1.333	2.666	2.666	2.666	1.333
AT	55.986	58.652	59.985	62.651	65.317	66.650	69.316	70.649	73.315	75.981	78.647	79.980	82.646	85.312	86.645	89.311	91.977	93.310	95.976	97.309	99.975	102.64	105.31	106.64
	Cycle 5												Cycle 6											
k	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
TS	R1	R2	R3	R1	R2	R3	R2	R3	R1	R2	R3	R2	R3	R1	R2	R1	R2	R3	R2	R3	R1	R2	R3	R2
H	52			52		52			52		52			52		52		52			52		52	
RI	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
RT	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80
FLT		130			130		130			130		130			130				130			130		130
S1	0	52(+78)	52	52	104(+78)	104	104	26	26	78(+78)	78	78	0	0	52(+78)	52	104(+78)	104	104	26	26	78(+78)	78	78
S2	91	91	182	78	78	169	65	156	52	52	143	39	130	26	26	117	117	195	91	182	78	78	169	65
S3	130	130	0	130	130	0	130	0	130	130	0	130	0	130	130	0	0	0	130	0	130	130	0	130
AP	2795	2795	2964	2964	2964	3133	3133	3302	3302	3302	3471	3471	3640	3640	3640	3809	3809	3861	3861	4030	4030	4030	4199	4199
(T)	2	2	1	2	2	1	2	1	2	2	1	2	1	2	2	2	2	1	2	1	2	2	1	2
(AT)	82	84	85	87	89	90	92	93	95	97	98	100	101	103	105	107	109	110	112	113	115	117	118	120
T	2.666	2.666	1.333	2.666	2.666	1.333	2.666	1.333	2.666	2.666	1.333	2.666	1.333	2.666	2.666	2.666	2.666	1.333	2.666	1.333	2.666	2.666	1.333	2.666
AT	109.31	111.97	113.31	115.97	118.64	119.97	122.64	123.97	126.64	129.30	130.63	133.30	134.63	137.30	139.97	142.63	145.30	146.63	149.30	150.63	153.30	155.96	157.29	159.96

Solution of small-size instances by a solver

After the schedule with possible full load is achieved (see Table 6), to maximize the total production within a certain time horizon (e.g., 8 h), a solver (e.g. the solver in MS Excel) can be used for optimization of the batch sizes of the tasks. In conducting the optimization, the following factors must be ascertained:

(1) Variables: batch sizes of all (or some) tasks, B_j: RI_1_k, RI_2_k, RI_3_k, RT_1_k, RT_2_k, RT_3_k, FLT_k, $k = 1, 2, \ldots, N$.

(2) Objective to be optimized: Maximize the total production,

$$\max \quad P = 0.4\sum_{k=1}^{N}(\mathrm{RI\_2}_k + \mathrm{RT\_2}_k) + 0.9\sum_{k=1}^{N}\mathrm{FLT}_k \tag{10}$$

(3) Constraints subjected to:

(i) Capacity of the processing units:

$$B_j \le B^{\mathrm{MAX}} \tag{11}$$

Table 4. Periodicity of Pattern Schedule I within 160 h (VPT) in Example 1

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6
Slots	12	12	12	12	12	12
Batches of H	5	5	5	5	5	5
Batches of R1	3	3	3	3	3	3
Batches of R2	5	5	5	5	5	5
Batches of R3	4	4	4	4	4	4
Batches of FLT	3	4	4	4	5	4
Total time	26.660	26.660	26.660	26.660	26.660	26.660
Total production	611	728	728	728	845	728

Table 5. Cycles 3–6 of Pattern Schedule II within 160 h (VPT) in Example 1

Cycle 3												
k	25	26	27	28	29	30	31	32	33	34	35	36
*S*2	39	39	130	130	130(+78)	104	117(+78)	91	91	104(+78)	78	91(+78)
AP	1339	1339	1508	1508	1560	1560	1729	1729	1729	1898	1898	2067
AT	55.986	58.652	61.318	63.984	65.317	67.983	69.316	71.982	74.648	75.981	78.647	79.980
Cycle 4												
k	37	38	39	40	41	42	43	44	45	46	47	48
*S*2	65	65	156	156	156(+78)	130	143(+78)	117	117	130(+78)	104	117(+78)
AP	2067	2067	2236	2236	2288	2288	2457	2457	2457	2626	2626	2795
AT	82.646	85.312	87.978	90.644	91.977	94.643	95.976	98.642	101.31	102.64	105.31	106.64
Cycle 5												
k	49	50	51	52	53	54	55	56	57	58	59	60
*S*2	91	91	182	182	182(+78)	156	169(+78)	143	143	156(+78)	130	143(+78)
AP	2795	2795	2964	2964	3016	3016	3185	3185	3185	3354	3354	3523
AT	109.31	111.97	114.64	117.3	118.64	121.3	122.64	125.3	127.97	129.3	131.97	133.30
Cycle 6												
k	61	62	63	64	65	66	67	68	69	70	71	72
*S*2	117	117	**208**	**208**	**208(+78)**	182	195(+78)	169	169	182(+78)	156	169(+78)
AP	3523	3523	3692	3692	3744	3744	3913	3913	3913	4082	4082	4251
AT	135.97	138.63	141.3	143.96	145.3	147.96	149.3	151.96	154.63	155.96	158.63	159.96

(ii) Batch sizes limited by the current states:

$$0.6(RI\_2_k + RT\_2_k) \leq S1_k$$
$$0.8(RI\_3_k + RT\_3_k) \leq S2_k \quad (12)$$
$$FLT_k \leq S3_k$$

(iii) Constraints to the least final intermediates:

$$S1_{N+1} = 0,\ S2_N = 0,\ S3_{N+1} = 0 \quad (13)$$

(iv) Equal processing times for two reactors RI and RT within a slot:

$$T_{\mathrm{I},j} = T_{\mathrm{T},j} \quad (14)$$

(v) Limitation of the total processing time by the given time horizon:

$$\sum_{k=1}^{N} T_k = \mathrm{TH} \quad (15)$$

After setting of the above variables, objective and constraints, run the solver, and the maximum production within the given time horizon is obtained. Table 7 shows the schedule with maximum production within 8 h, which is the same as the solution obtained in the work of Maravelias and Grossmann.[13] If the given time horizon TH is changed into 6 or 7 h, and then rerun the solver, a 6- or 7-h schedule can be achieved, whose maximum production is presented in Table 8.

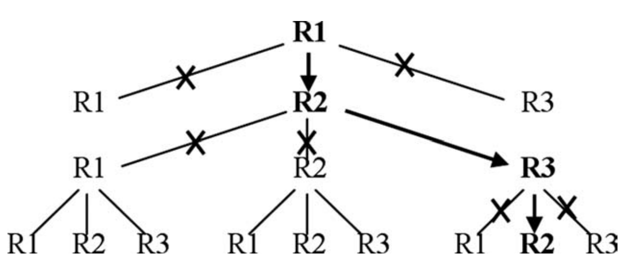

Figure 5. Search tree for four-slot task sequence.

Similarly, after a task sequence of five, six, or seven slots is determined, a schedule with possible full load can be easily obtained by manual calculation or using the MS Excel, subsequently, the corresponding time range can be determined. According to the time ranges, for the given time horizon (e.g., 9 h), we need to determine which task sequence should be adopted. Then use the solver to maximize the production. Table 8 presents statistic results of the schedules within different short time horizons.

A fact that should be noticed is that not any precise time horizons correspond to a maximum production. For example, for the 9-h instance, an optimized production, 147.839, is obtained from the task sequence "R1|R2|R3R3/R1|R2," but this value is less than 174.417, which is the maximum production for the task sequence "R1|R2|R3|R2" completed within 8.72 h. Therefore, the maximum production for the 9-h instance is 174.417, rather than 147.839.

Table 9 presents the 14-h schedule, where "R3R3/R1" means that RI performs two batches of R3 in that time slot while RT performs one batch of R1. In fact, these two

Table 6. A Four-Slot Schedule with Possible Full Load

k	1	2	3	4	5
TS	R1	R2	R3	R2	
RI	50	50	37.5	33.33	
RT	80	80	60	53.33	
Sum	130	130	97.5	86.67	
FLT				97.5	
*S*1	0.000	130	52	52	0.000
*S*2	0.000	0.000	78	0.000	61.75
*S*3	0.000	0.000	0	97.5	0.000
T	T1	T2	T3	T2	sum T
RI	2.666	2.666	1.167	2.222	
RT	2.666	2.666	1.167	2.222	
FLT				1.984	
Max	2.666	2.666	1.167	2.222	8.7200
P	174.417				

Table 7. The Schedule with Maximum Production within 8 h

k	1	2	3	4	5
TS	R1	R2	R3	R2	
RI	40.507	45.381	34.036	22.131	
RT	64.812	72.610	54.458	35.409	
Sum	105.319	117.991	88.494	57.540	
FLT				88.494	
$S1$	0.000	105.319	34.524	34.524	0.000
$S2$	0.000	0.000	70.795	0.000	43.374
$S3$	0.000	0.000	0.000	88.494	0.000
T	T1	T2	T3	T2	sum T
RI	2.413	2.543	1.120	1.924	
RT	2.413	2.543	1.120	1.924	
FLT				1.924	
Max	2.413	2.543	1.120	1.924	8.000
P	149.857				

batches of R3 by RI can be regarded as one batch, where the capacity of RI is supposed to be $50 \times 2 = 100$ and the processing time is calculated by $T3 = 2\alpha_3 + \beta_3 \cdot B_3 = 2 \times 0.667 + 0.01332 \times 97.5 = 2.633$.

Paying attention to the last two slots in Tables 7 and 9, we can notice that R3 is followed by R2 and FLT. Arrangement like this enables the schedules to have maximum production. In this subsection, the small-size problems with zero initial states ($S1_1$, $S2_1$, $S3_1$) have been solved. In case that the initial states ($S1_1$, $S2_1$, $S3_1$) are not all zero, a task sequence has to be re-identified, which will be discussed in the next section for the solution of phase two.

Decomposition of Long-Horizon Instances in Example 1

To maximize the total production within a given time horizon, parallel working and full load of the two reactors RI and RT are two key points in solving the problems. Based on these two points and the logical relationships connoted in the process recipe, a pattern schedule as shown in Table 3 has been obtained. On the other hand, a small-size problem can be easily solved by the existing MILP models or the heuristic method proposed in previous section.

Therefore, for a long time horizon, the novel decomposition method described previously can be applied: divide it into phases one and two. Phase one is copied from the pattern schedule, and thus needs no more optimization. Phase two involves a small-size subproblem that can be optimized easily by the proposed heuristic method. Between phase one and phase two, there is a cut point that can be determined through analysis and reasoning. The cut point may not exactly be at the end of a cycle of the pattern schedule.

Long-horizon instances with VPT

In this subsection, the long time horizons of 24 h (1 day), 48 h (2 days), 72 h (3 days), 96 h (4 days), 120 h (5 days), 144 h (6 days), and 168 h (7 days) are handled by the novel

Table 8. Statistic Results of the Schedules within Different Short Time Horizons

TH (h)	N	TS	Time Range	Max P
6	4	R1|R2|R3|R2	[4.669, 8.720]	59.876
7				104.866
8				149.857
9				174.417 (8.72h)
9	4	R1|R2|R3R3/R1|R2	[5.336, 10.098]	147.839
10				187.928
11	5	R1|R2|R1/R2|R3|R2	[6.003, 11.587]	222.674
12	6	R1|R2|R3|R1/R2|R3|R2	[6.670, 12.457]	256.459
13	6	R1|R2|R3|R1/R2|R1/R3|R2	[7.337, 13.804]	276.460
14	6	R1|R2|R3R3/R1|R2|R3|R2	[7.337, 14.149]	321.948
16	7	R1|R2|R2/R1|R3|R2|R1/R3R3|R2	[8.671, 17.261]	365.568

Table 9. The Schedule with Maximum Production within 14 h

k	1	2	3	4	5	6	7
S	R1	R2	R3R3/R1	R2	R3	R2	
RI	50	50	97.5	50	42.188	30.269	
RT	80	80	73.219	80	67.5	48.430	
Sum	130	130		130	109.688	78.699	
FLT				97.5		109.688	
$S1$	0.000	130	52	125.219	47.219	47.219	0.000
$S2$	0.000	0	78	0	87.75	0.000	58.188
$S3$	0.000	0	0	97.5	0	109.688	0.000
T	T1	T2	T3/T1	T2	T3	T2	sum T
RI	2.666	2.666	2.633	2.666	1.229	2.140	
RT	2.666	2.666	2.553	2.666	1.229	2.140	
FLT				1.984		2.065	
Max	2.666	2.666	2.633	2.666	1.229	2.140	14
P	321.948						

Table 10. Statistic Data of the Schedules by Decomposition from Pattern Schedule I (VPT)

	Decomposition			Phase 2	
TH	Time	Slots	*P*	TS	Initial States
24 h (1 day)	15.996 + 8.004	7 + 4	273 + 305.622 = 578.622	R1|R2|R3|R2	(26, 39, 130)
48 h (2 days)	41.323 + 6.677	18 + 3	1001 + 230.667 = 1231.667	R1/R3|R3|R2	(26, 169, 0)
72 h (3 days)	59.985 + 12.015	27 + 6	1508 + 432.852 = 1940.852	R1|R2|R3|R1/R2|R3|R2	(52, 26, 130)
96 h (4 days)	85.312 + 10.688	38 + 4	2236 + 382.583 = 2618.583	R1/R3R3|R2|R1/R3R3|R2	(5, 52, 130)
120 h (5 days)	111.970 + 8.030	50 + 4	2964 + 318 = 3282.000	R3|R1/R2|R3|R2	(52, 182, 0)
144 h (6 days)	133.300 + 10.700	60 + 5	3640 + 320.667 = 3960.670	R3|R1|R2|R3|R2	(0, 130, 0)
168 h (7 days)	159.960 + 8.040	72 + 4	4368 + 242.125 = 4610.125	R3|R1|R3|R2	(0, 156, 0)

decomposition method. Table 10 presents the statistic data of the schedules by decomposition from pattern schedule I. The cut points in the pattern schedule (see Table 3) are highlighted with shadow in the rows of AT.

The subproblems in phase two are solved by the proposed heuristic method in previous section. The difference is that the initial states ($S1$, $S2$, $S3$) are not all zero. A task sequence has to be constructed according to the specific initial states. After the task sequence of phase two is determined, the solver in MS Excel is used for the batch sizes optimization. Table 11 presents phase two (8.004 h) of the 24-h schedule optimized by the solver. What should be mentioned is that, phase two (6.677 h) of the 48-h schedule in Table 12, does not require the solver for optimization, where the schedules with possible full load can be regarded as having maximum production.

Other subproblems in phase two can be similarly solved. In He's dissertation[25] and the book by He and Hui,[26] we presented the detailed solutions. Phase two (12.015 h) of the 72-h schedule was optimized by the solver. However, phase two (10.688 h) of the 96-h schedule, phase two (8.03 h) of the120-h schedule, phase two (10.7 h) of the144-h schedule, and phase two (8.04 h) of the 168-h schedule do not require the solver for optimization. The schedules with possible full load can be regarded as having maximum production, like phase two of the 48 h schedule in Table 12.

After the subschedule in phase two is obtained, integrated with the subschedule in phase one copied from the pattern schedule in Table 3, a complete schedule is available. The maximum production for each time horizon is presented in Table 10.

When pattern schedule II (see Table 5) is used for pattern matching, it is found that the decompositions of 24, 48, 96, and 120 h are the same as those in Table 10, but the decompositions of 72, 144, and 168 h are changed as shown in Table 13. The details of phase two of 72, 144, and 168 h are presented in He's dissertation[25] and the book by He and Hui.[26] By the decompositions from pattern schedule II, except that the schedule of 168 h has a smaller production of 4584 than before (4610.13), the schedules of from 24 to 144 h have the same productions as before.

Comparison to existing methods

This subsection conducts a comparative study between the proposed method and the existing methods for the motivating example. The existing methods were proposed Ierapetritou and Floudas,[9] Maravelias and Grossmann,[13] and Wu and Ierapetritou.[19,22]

Cyclic scheduling usually overlooks the start-up and finishing phases, but these two phases must be considered to obtain a feasible schedule for operation. Wu and Ierapetritou[22] proposed a three-phase method. The overall time horizon is divided into three phases, the initial phase when the necessary amounts of intermediates are produced to start the cyclic schedule, the main phase when cyclic scheduling is applied and the final phase to wrap up all the intermediates. The sum of time lengths of all three phases equals the whole time horizon. Each of the three phases involves a different MP model. The optimal cycle length results from solving the cyclic scheduling problem with an objective to maximize average profit over the cycle time. The problem for initial

Table 11. Phase 2 (8.004 h) of the 24-h Schedule

k	8	9	10	11	12
TS	R1	R2	R3	R2	
RI	31.320	33.333	50	35.534	
RT	50.113	53.333	80	56.854	
Sum	81.433	86.667	130	92.388	
FLT		130		130	
*S*1	26	107.433	55.433	55.433	0.000
*S*2	39	39	104	0.000	55.433
*S*3	130	130	0	130	0
	T1	T2	T3	T2	sum T
RI	2.168	2.222	1.333	2.281	
RT	2.168	2.222	1.333	2.281	
FLT		2.200		2.200	
Max	2.168	2.222	1.333	2.281	8.004
P	305.622				

Table 12. Phase 2 (6.677h) of the 48-h Schedule

k	19	20	21	22
TS	R1/R3	R3	R2	
RI	50	50	48.718	
RT	70	80	77.949	
Sum		130	126.667	
FLT			200	
*S*1	26	76	76	0
*S*2	169	129	25	121
*S*3	0	70	200	0
	T1/T3	T3	T2	sum T
RI	2.666	1.333	2.632	
RT	1.250	1.333	2.632	
FLT			2.666	
Max	2.666	1.333	2.666	6.665
P	230.667			

Table 13. Statistic Data of the Schedules by Decomposition from Pattern Schedule II (VPT)

TH	Decomposition			Phase two	
	Time	Slots	P	TS	Initial States
72 h (3 days)	58.652+13.348	26+7	1508+432.85 = 1940.85	R3 ǀ R1 ǀ R2 ǀ R3 ǀ R1/R2 ǀ R3 ǀR2	(52, 130, 0)
144 h (6 days)	133.300+10.700	60+5	3523+437.67 = 3960.67	R1 ǀ R3 ǀ R2 ǀ R3 ǀ R2	(0, 117, 130)
168 h (7 days)	159.960+8.040	72+3	4251+333 = 4584	R1 ǀ R2/R3R3 ǀ R2	(0, 143, 130)

*IS = initial states of phase two.

phase is solved first with the objective function of minimum makespan so as to ensure the existence of feasible solution to provide those intermediates for cyclic scheduling. Then the same problem is solved with the objective of maximizing the profit (production, in fact) within the time horizon obtained from the solution of the makespan minimization problem. The problem for the final phase can be solved in parallel once the time horizon for the cycle length and the initial phase are determined. The intermediates considered for the final phase are obtained from the cyclic schedule, and the time horizon for the final phase is the time left for the planning problem.

It took a long CPU time (2884.41 + 1590.35 + 3187.21 + 611.23 = 7661.97 s)[22] for the three-phase method to obtain the final solution of the 168-h schedule for Example 1 (see Table 14). The final optimal cycle length obtained is 23.790 h with the objective value of 279.029 (average profit over the cycle time) within 2884.41 CPU seconds. The problem of production maximization for the initial and final periods were then solved simultaneously based on the time horizon calculated from the makespan minimization problem for the initial period. It took 1590.35 CPU seconds to solve the makespan minimization problem, 3187.21 and 611.23 CPU seconds to solve the production maximization problems for the initial and final period problems, respectively. The overall objective function value representing the total profit over the whole time horizon is 45,698.90 (equivalent to a maximum production of 4569.890, as $P1$ and $P2$ have a price of 10$/kg).

The three-phase method is a good idea for complete feasible schedules within long time horizons and has practical advantages in solving large industrial problems indeed. However, this method still has defects: (1) not guarantee global optimality though feasible and near-optimal solutions may be obtained; (2) each phase still needs relatively long CPU time due to different models involved; (3) in the same batch plant (e.g., Example 1), for instances with different long time horizons, the three phases may change totally, which means a new time-consuming solution procedure for a specific instance; and (4) as the problem size increases, the total CPU time will increase greatly. Therefore, although the models of this method are systematic, the solution procedure is time consuming and instance dependent for the same batch plant.

Our approach for Example 1 is a two-phase method. The first phase schedule is obtained by using the pattern schedule as a template for pattern matching. The subproblem for the first phase needs no solution procedure thus saving plenty of CPU time. The small-size subproblem for the finishing phase is very easily solved by our heuristic method within very short CPU time. Table 14 shows the comparison of the results between the proposed and already existing methods. From Table 14 and our work, the following facts can be noticed:

(1) The pattern schedule, once constructed, can be utilized repeatedly for the decomposition of the instances with different time horizons, providing the practitioners with high convenience.

(2) For the small-size instances (less than 24 h), the solutions by the proposed heuristic method are not better than those by MILP models, but the search time is very short, often several seconds are enough. Although the identification of a proper task sequence needs some time, a large number of binary variables and sequence constraints are avoided. Only (some of) the batch sizes need to be optimized. A quite small number of constraints are used.

(3) In our work, as the overall problem size increases, the total CPU time may not increase, on the contrary, may decrease. From Table 10, we can see that the finishing phases of the above 72-h schedules are shorter than the finishing phase (12.015 h) of the 72-h schedule.

(4) The proposed two-phase decomposition saves plenty of CPU time. The proposed method is basically a two-phase

Table 14. Comparison of the Results Between the Proposed Method and the Existing Methods

TH	Proposed Methods		Wu and Ierapetritou		Maravelias and Grossmann	Ierapetritou and Floudas
	P	Mean P (kg/h)	P/CPU (s)	Mean P (kg/h)	P/CPU(s)	P/CPU(s)
8 h	149.857				149.86/0.34	149.819/0.47[a]
12 h	256.459				261.101/8.06	
16 h	365.568		355.042/18.97[a]			373.710/178[a]
24 h (1 day)	578.62	24.109	591.842/30.77[a]			603.492/92368[a]
48 h (2 days)	1231.67	25.660	1271.098/804.95[a]			
72 h (3 days)	1940.85	26.956				
96 h (4 days)	2618.58	**27.277**				
120 h (5 days)	3282	**27.350**				
144 h (6 days)	3960.67	**27.505**				
168 h (7 days)	4610.13	**27.441**	4569.89/7661.97	**27.202**		

[a]From Wu and Ierapetritou.[19]

Table 15. Comparison of the Results between the Proposed Method and GA (CPT)

	Proposed Method					GA		
TH	Time	Slots	$P1$	$P2$	P	$P1$	$P2$	P
37 h	31+6	18+4	485.33	819	1304.33	472	805.50	1277.50
108 h	100+8	60+5	1386.67	2574	3960.67	1384	2567.25	3951.25

technique. Although in Example 2 three phases are involved in construction of the pattern schedule, the long time horizons for scheduling are still decomposed into two phases.

(5) The solution ($P = 4610.13$) of the 168-h schedule by the proposed method is better than that ($P = 4569.89$) by Wu and Ierapetritou.[22] If used the existing MILP models for solving the subproblems of phase two, the production by our method may be improved further.

(6) The mean production for each above 72-h schedule by our method is close to the mean production (27.9029) of the optimal cyclic schedule by Wu and Ierapetritou.[22]

Long-horizon instances with CPT

In another work,[24] we have proposed a novel genetic algorithm (GA) for the scheduling of Example 1 with CPT, where a GA is developed for makespan minimization and total production maximization. In fact, it is obvious that the proposed method in this paper can be applied to the case with CPT. Table 15 presents the comparison of the results of the 37- and 108-h schedules by the proposed method and the GA. Tables 16 and 17 present the finishing phases of the two schedules. The first phase of each schedule can be referred to Table 3. From Table 15, we can see that the proposed method obtained better solutions than the GA. Furthermore, the proposed method needs neither complex programming as the GA nor MILP models.

Other Examples

Apart from Example 1, other examples are investigated with the purpose of checking the applicability of the proposed method under different conditions in the batch plants. With the change of the process conditions, such as the processing times, the capacities of some units and/or some intermediate tanks, and the process recipe itself, the periodicity of the corresponding pattern schedule will be different, or the master task sequences and the relevant bottleneck units will change either. In He's dissertation[25] and the book by He and Hui,[26] an example modified from Example 1, and another example modified from Example 2 demonstrate these changes.

One may argue that large-scale scheduling not only means long time horizons but also implies a large number of units and tanks, as well as the complexity of the process recipe represented by a relatively complicated STN diagram. To address application to such problems, Examples 3 and 4 are used to demonstrate the effectiveness of the proposed method. From the benchmark applications, the proposed method is expected to solve other multipurpose process scheduling problems.

Example 2

Example 2 is an example already studied by Maravelias and Grossmann,[13] and then by Shaik et al.[17] The STN diagram of this example is shown in Figure 6, and the data of this example is presented in Tables A3 and A4, where the utility requirements are ignored. In this example, there are two types of reactors available for the process (Types I and II) with different numbers of corresponding units available: two reactors (RI1 and RI2) of Type I, but only one reactor (RII) of Type II. Reactions R1 and R2 require a Type I reactor, whereas reactions R3 and R4 require a Type II reactor. The objective here is to maximize the total production (or profit) within a certain time horizon, namely, 120 h (5 days), 144 h (6 days), and 168 h (7 days).

Seen roughly from the STN diagram, Example 2 seems to be a relatively simple problem. Given different prices of P1 and P2, the maximization of profit seems to be different from that of production, and the heuristics for scheduling should aim to yield more P2. However, the detailed analysis and perception to the plant reveal that the maximization of profit is still equivalent to that of production, and the heuristics do not necessarily aim to output more P2. The reasons are given below: (1) Tasks R3 (for P1) and R4 (for P2) use the same reactor RII with the identical capacity, so they cannot be performed at the same time; (2) Although R4 can produce more P2 by using the same amount of S3 as R3, a full batch of R4 takes double processing time compared to a full batch of R3, thus the average production over time by

Table 16. Phase 2 of the 37-h Schedule (CPT)

k	19	20	21	22	23
TS	R3	R2/R1	R3	R2	
RI	50	43.33	50	50	
RT	80	78	80	80	
Sum	130		130	130	
FLT		130		130	
$S1$	26	26	78	78	0
$S2$	169	65	104	0	91
$S3$	0	130	0	130	0
P	303.330	$P1$	69.33	$P2$	234

Table 17. Phase 2 of the 108-h Schedule (CPT)

k	61	62	63	64	65	67
TS	R3	R1	R2	R3	R2	
RI	50	50	50	50	33.333	
RT	80	80	80	80	53.333	
Sum	130	130	130	130	86.667	
FLT			130		130	
$S1$	0	0	130	52	52	0
$S2$	130	26	26	117	13	78
$S3$	0	130	130	0	130	0
P	320.667	$P1$	86.667	$P2$	234	

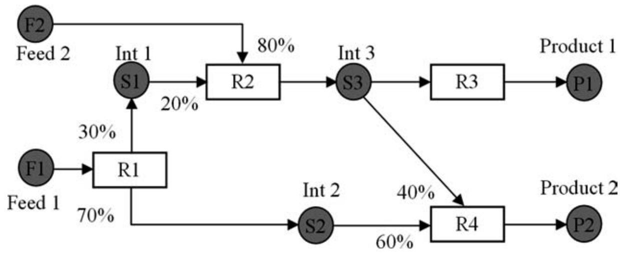

Figure 6. State-task network of Example 2.

R4 is less than that by R3; and (3) even if the higher price of P2 is considered, R3 has a higher average profit over time than R4. Therefore, the time-consuming tasks of R4 should be arranged as early as possible, but try our best to use batches of R3, instead of R4, to wrap up the material of S3 in the end of a schedule.

SCR Equations. Table 18 presents the notations needed for a schedule of Example 2, where TS1 is the task sequence performed by RI1 and RI2, TS2 is the task sequence by RII. The length of a time slot k depends on the task assignment in TS1 and TS2. A full batch of R1 may need one, two, or three time slots, dependent on the task assignment on RII: if no task is performed simultaneously on RII, one time slot is needed; If a batch of R3 and then a batch of R4 performed simultaneously On RII, two slots; if three batches of R3 On RII, three slots (see Table 19).

With the notations given, the SCR equations and constraints can be derived from the STN diagram and the problem data:

$$S1_{k+1} = S1_k + 0.3(\text{RI1\_R1}_k + \text{RI2\_R1}_k) - 0.2(\text{RI1\_R2}_k + \text{RI2\_R2}_k) \quad (16)$$

$$S2_{k+1} = S2_k + 0.7(\text{RI1\_R1}_k + \text{RI2\_R1}_k) - 0.4\text{RII\_}R4_k \quad (17)$$

$$S3_{k+1} = S3_k + 1.0(\text{RI1\_R2}_k + \text{RI2\_R2}_k) - (1.0\text{RII\_R3}_k + 0.6\text{RII\_R4}_k) \quad (18)$$

$$S1_k \leq 100, S2_k \leq 200, S3_k \leq 500 \quad (19)$$

Natural Periodicity Analysis. The procedure to achieve an optimized schedule of this example is similar to that of Example 1. First, under the assumption that crucial units work continuously and with full load as possible, a natural cycle can be calculated as follows. From the equations:

$$0.3P_{\text{R1}} = 0.2P_{\text{R2}} \quad (20)$$

$$0.7P_{\text{R1}} = 0.6P_{\text{R4}} \quad (21)$$

$$P_{\text{R2}} = 0.4P_{\text{R4}} + P_{\text{R3}} \quad (22)$$

where P_{R1}, P_{R2}, P_{R3}, and P_{R4} are the total material amount of reactions R1, R2, R3, and R4, respectively, we have (1) $P_{\text{R3}}/P_{\text{R4}} = 31/35$, which means 31 full batches of R3 (31 × 80 kg) need 35 full batches of R4 (35 × 80 kg) for mass balance, (2) $P_{\text{R1}}/P_{\text{R3}} = 30/31$ and (3) $P_{\text{R2}}/P_{\text{R1}} = 3/2$.

Because R3 and R4 are conducted by RII, the total process time of 31 full batches of R3 (31 × 80 kg) and 35 full batches of R4 (35 × 80 kg) is (31 × 1.25 + 35 × 2.5) = 126.25 h under the assumption of working continuously with full batches of RII.

If RI1 and RI2 perform the same reaction (R1 or R2) in parallel and with full load, and a parallel full batch equals 130 kg, so from $P_{\text{R1}}/P_{\text{R3}} = 30/31$, the number of full batches of R1 is calculated out as (30/31) × 31 × 80/130 = 18.46; and from $P_{\text{R2}}/P_{\text{R1}} = 3/2$, the number of full batches of R2 is 27.69. The total process time of R1 and R2 is (18.46 × 3.75 + 27.69 × 2.5) ≈ 140 h (>126.25 h).

Up to now, it can be deduced that R1 and R2 are two crucial tasks, and RI1 and RI2 are the bottleneck units to maximize the production of Example 2. Therefore, RI1 and RI2 should work continuously, in parallel and with full load. RII perfoms R3 or R4 in time as the states of Int 2 and Int 3 are satisfied. So in Example 2, TS1 is the master sequence, TS2 the slave one.

The Pattern Schedule. Table 19 presents a pattern schedule of Example 2, which is obtained under the following considerations: (1) RI1 and RI2 should work continuously and with full load; (2) RI1 and RI2 perform the same reaction (R1 or R2) in parallel; (3) at the beginning of the schedule, a batch of R1 must be followed by a batch of R2, and then a batch of R3 or R4 is possible to be started; (4) RII performs a full batch of R3 if the current state $S3_k$ is enough; (5) RII performs a full batch of R4 if the current states $S2_k$ and $S3_k$ are enough; and (6) at the end of the schedule, after the final batch of R2 is finished, R3 and/or R4 are still required to wrap up the states of S2 and S3 to satisfy the least intermediates in the tanks. Considering average production (or profit) over time mentioned above, the time-consuming tasks of R4 should be arranged as early as possible, try our best to use batches of R3, rather than R4, to wrap up the material of S3 in the end of a schedule.

With the above considerations, the task sequence "R1|R2|R1|R2|R2" can be repeated in RI1 and RI2, which is regarded as a subcycle. The length of the subcycle is 15 h. A natural cycle consisting of an initial phase, eight subcycles and an end phase is discovered as shown in Table 19. The length of the initial phase is also 15 h. The length of the end phase is 11.25 h. Hence the total length of the natural cycle is (9 × 15 + 11.25) = 146.25 h. In the end phase, one parallel partial batch (60 kg) of R1 and one parallel partial batch (90 kg) of R2 are carried out. At the end of such a natural cycle, all intermediate states (S1, S2, S3) return to zero. We then use the schedule with this natural cycle (see Table 19) for pattern matching.

Table 18. Notations for a Schedule of Example 2

k	Time slots, $k = 1, 2, \ldots, N$
TS1	A task sequence of R1, R2
TS2	A task sequence of R3, R4
RI1_R1_k	Batches of reaction 1 on RI1
RI1_R2_k	Batches of reaction 2 on RI1
RI2_R1_k	Batches of reaction 1 on RI2
RI2_R2_k	Batches of reaction 2 on RI2
RII_R3_k	Batches of reaction 3 on RII
RII_R4_k	Batches of reaction 4 on RII
$(S1_k, S2_k, S3_k)$	States of tanks 1, 2, and 3
$T1$	Process time of a task in TS1
$T2$	Process time of a task in TS2
T	Process time of a phase or cycle

Table 19. The Pattern Schedule of Example 2 (TH = 146.25 h)

	Initial phase						*Sub_cycle 1*							*Sub_cycle 2*						
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
TS1	R1	R2	R1		R2	R2	R1		R2	R1		R2	R2	R1		R2	R1		R2	R2
RI1	80	80	80		80	80	80		80	80		80	80	80		80	80		80	80
RI2	50	50	50		50	50	50		50	50		50	50	50		50	50		50	50
sum	130	130	130		130	130	130		130	130		130	130	130		130	130		130	130
TS2			R3	R4		R3	R4	R3	R4	R4	R3	R4	R4	R3	R4	R4	R3	R4	R4	R4
RII			80	80		80	80	80	80	80	80	80	80	80	80	80	80	80	80	80
S1	0	39	13	13	52	26	0	0	39	13	13	52	26	0	0	39	13	13	52	26
S2	0	91	91	91	134	134	134	86	177	129	81	172	124	76	76	119	71	71	114	66
S3	0	0	130	50	18	148	198	166	86	184	152	72	170	268	188	156	254	174	142	240
T1	3.75	2.5	3.75		2.5	2.5	3.75		2.5	3.75		2.5	2.5	3.75		2.5	3.75		2.5	2.5
T2			1.25	2.5		1.25	2.5	1.25	2.5	2.5	1.25	1.25	2.5	1.25	2.5	2.5	1.25	2.5	2.5	2.5
T	15						15							15						

	Sub_cycle 3								*Sub_cycle 4*								*Sub_cycle 5*							
k	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
TS1	R1			R2	R1		R2	R2	R1			R2	R1		R2	R2	R1		R2	R1			R2	R2
RI1	80			80	80		80	80	80			80	80		80	80	80		80	80			80	80
RI2	50			50	50		50	50	50			50	50		50	50	50		50	50			50	50
sum	130			130	130		130	130	130			130	130		130	130	130		130	130			130	130
TS2	R3	R3	R3	R4	R3	R4	R4	R4	R3	R3	R3	R4	R3	R4		R4	R3	R4	R4	R3	R3	R3		R4
RII	80	80	80	80	80	80	80	80	80	80	80	80	80	80		80	80	80	80	80	80	80		80
S1	0	0	0	39	13	13	52	26	0	0	0	39	13	13	52	26	0	0	39	13	13	13	52	26
S2	18	18	18	109	61	61	104	56	8	8	8	99	51	51	94	94	46	46	89	41	41	41	132	132
S3	338	258	178	98	196	116	84	182	280	200	120	40	138	58	26	156	254	174	142	240	160	80	0	130
T1	3.75			2.5	3.75		2.5	2.5	3.75			2.5	3.75		2.5	2.5	3.75		2.5	3.75			2.5	2.5
T2	1.25	1.25	1.25	2.5	1.25	2.5	2.5	2.5	1.25	1.25	1.25	2.5	1.25	2.5		2.5	1.25	2.5	2.5	1.25	1.25	1.25		2.5
T	15								15								15							

	Sub_cycle 6							*Sub_cycle 7*								*Sub_cycle 8*								*End phase*					
k	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73
TS1	R1		R2	R1		R2	R2	R1			R2	R1		R2	R2	R1			R2	R1		R2	R2	R1	R2				
RI1	80		80	80		80	80	80			80	80		80	80	80			80	80		80	80	36.923	55.385				
RI2	50		50	50		50	50	50			50	50		50	50	50			50	50		50	50	23.077	34.615				
sum	130		130	130		130	130	130			130	130		130	130	130			130	130		130	130	60	90				
TS2	R3	R4	R4	R3	R4	R4	R4	R3	R3	R3	R4	R3	R4	R4	R4	R3	R3	R3		R3	R4		R4	R4	R4	R4	R3	R3	R3
RII	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80		80	80		80	80	80	80	80	80	80
S1	0	0	39	13	13	52	26	0	0	0	39	13	13	52	26	0	0	0	39	13	13	52	26	0	18	0	0	0	0
S2	84	84	127	79	79	122	74	26	26	26	117	69	69	112	64	16	16	16	107	107	107	150	150	102	96	48	0	0	0
S3	228	148	116	214	134	102	200	298	218	138	58	156	76	44	142	240	160	80	0	130	50	18	148	246	214	272	240	160	80
T1	3.75		2.5	3.75		2.5	2.5	3.75			2.5	3.75		2.5	2.5	3.75			2.5	3.75		2.5	2.5	2.135	1.885				
T2	1.25	2.5	2.5	1.25	2.5	2.5	2.5	1.25	1.25	1.25	2.5	1.25	2.5		2.5	1.25	1.25	1.25	2.5	1.25	2.5		2.5	2.5	2.5	2.5	1.25	1.25	1.25
T	15							15								15								11.25					

From the process recipe, we know that at the beginning of the schedule, a batch of R1 must be followed by a batch of R2, and then a batch of R3 or R4 is possible to be started. At the end of the schedule, after the final batch of R2 is finished, R3 and/or R4 are still required to wrap up the material states of S2 and S3. Therefore, to save the process time, we should let RII start work as early as possible at the beginning of the schedule. With this consideration, if the first two parallel batches R1 and R2 are not full, RII can start to work early. After RII starts working, RI1 and RI2 work with full load. In this way, although the process time in the initial phase is shortened, the process time in the end phase is extended, and the total process time will be longer. Hence, we apply full batches of R1 and R2 at the beginning of the pattern schedule.

Decomposition. With the pattern schedule at hand, schedules over long time horizons can be obtained by the decomposition method. The first phase of a schedule can be copied from the pattern schedule. The second phase can be optimized by the proposed heuristic method. Table 20 presents the situation of the decomposition of the schedules of 120, 144, and 168 h in Example 2.

Table 20 shows that, 120 h is divided into a length of 105 h and a length of 15h, and the initial states for the 15-h subschedule are ($S1$, $S2$, $S3$) = (0, 26, 298); 144 h is divided into 135 and 9 h, and the initial states for the 9-h subschedule are (0, 102, 246); 168 h is divided into 150 and 18 h, and the initial states for the 18-h subschedule are (0, 1, 268).The details of subschedules of 15, 9, and 18 h are, respectively, shown in Tables 21–23.

Table 20. Statistic Data of the Schedules of Example 2

TH	Decomposition			End Phase	
	Time	Slots	P	Initial States	Time Length
120 h	105 + 15	51 + 8	3680 + 702 = 4382	(0, 26, 298)	15 (Table 21)
144 h	135 + 9	67 + 5	4800 + 400 = 5200	(0, 102, 246)	9 (Table 22)
168 h	150 + 18	74 + 11	5360 + 788 = 6148	(0, 1, 268)	18 (Table 23)

Example 3

Figure 7 shows the STN diagram for this example which was modified by Maravelias and Grossmann[13] from an example of Papageorgiou and Pantelides.[27] This example comprises resource constraints, mixed storage policies, variable batch sizes but constant processing times (CPT), and utility requirements, and was first solved by Maravelias and Grossmann,[13] then studied by Janak et al.[28] and Shaik et al.,[17] and the relevant data are given in Tables A5 and A6, where the utility requirements are ignored.

This plant consists of six units involving 10 processing tasks and 14 states: unit 1 for tasks T1 and T4, unit 2 for T2, unit 3 for T3, unit 4 for T5 and T6, unit 5 for T7 and T9, and unit 6 for T8 and T10. Unlimited intermediate storage (UIS) is available for raw materials F1 and F2, intermediates I1 and I2, and final products P1, P2, P3, and WS; finite intermediate storage (FIS) is available for states S3 and S4; no intermediate storage (NIS) is available for states S2 and S6; and a zero wait (ZW) policy applies for states S1 and S5. The objective is to maximize the total profit/production of the plant within a given time horizon, namely, 48 h (2 days), 72 h (3 days), 96 h (4 days), 120 h (5 days), 144 h (6 days), and 168 h (7 days).

Superficially, this example seems to be much more complicated. However, actually this example is found to be easily solved through the pattern matching method. In maximizing the total profit/production, ADD, T6, P2 and WS can be ignored. We should note that ZW is the special case of NIS and FIS. So according to the storage policies, NIS can be treated as ZW, thus T1, T2, and T3 are three ZW tasks that can be performed by U1, U2, and U3, respectively, and successively, simply regarded as T1(T2T3) with a full batch of 5 kg; similarly, T4 and T5 (two FIS tasks) can be treated as ZW tasks by U1 and U4, simply as T4(T5) with a full batch of 5 kg; T7 and T8 are ZW tasks by U5 and U6, simply as T7(T8) with a full batch of 3kg; T9 and T10 (two NIS tasks) can be treated as ZW tasks by U5 and U6, simply as T9(T10) with a full batch of 3 kg. With such ZW treatment, S1, S2, S3, S5, and S6 can be ignored. Each full batch of T7(T8) leave (10% − 5%) = 5% of this batch stored in S4, the vacancy of which (30kg) allows 200 full batches of T7(T8). That means, within 200 batches of T7(T8), S4 has enough capacity, so S4 also can be ignored while scheduling.

In this example, T1 and T4 share U1, T7 and T9 share U5, while T8 and T10 share U6. Hence, U1, U5, and U6 may be the bottleneck units, considering that U2, U3, and U4 can complete T2, T3, and T5 in time. To enable a natural cycle have maximum production, T1 and T4 in U1, T7 and T9 in U5 are assumed to be performed in full batches. With such assumption, five full batches of T7(T8), producing 13.5 kg of I2, allow nine full batches of T9(T10) to follow. Meanwhile, nine full batches of T9(T10) need to consume13.5 kg of I1 produced by T1(T2T3). To maximize production, U1 should work continuously and with full load as possible. If U5 complete five full batches of T7 and nine full batches of T9 successively, it needs $5 \times 4 + 9 \times 2 + 3 = 41$ h. However, the actual length of the schedule cycle may be more than 41 h, if not absolutely successively. With these issues taken into account, a pattern schedule with a natural cycle of 46 h has synthesized as shown in Table 24, where each time slot k is 1 h. The relevant states are calculated using the following formulas:

$$I1_{k+1} = I1_k + T3_k - 0.5T9_k - T4_k \tag{23}$$

$$I2_{k+1} = I2_k + 0.9T8_k - 0.5T9_k \tag{24}$$

Table 21. The End Phase of the 120-h Schedule in Example 2

	End Phase								
k	52	53	54	55	56	57	58	59	60
TS1		R1		R2	R1		R2		
RI1		80		80	80		80		
RI2		50		50	50		50		
Sum		130		130	130		130		
TS2	R3	R3	R3	R4	R3	R4	R4	R3	R3
RII	80	80	80	80	80	80	80	80	62
$S1$	0	0	0	39	13	13	52	26	26
$S2$	26	26	26	117	69	69	112	64	64
$S3$	298	218	138	58	156	76	44	142	62
$T1$		3.75		2.5	3.75		2.5		
$T2$	1.25	1.25	1.25	2.5	1.25	2.5	2.5	1.25	1.03
T	14.775 ≈ 15								

Table 22. The End Phase of the 144-h Schedule in Example 2

	End Phase				
k	68	69	70	71	72
TS1	R1	R2			
RI1	38.67	58			
RI2					
Sum	38.67	58			
TS2	R4	R4	R3	R3	R3
RII	80	80	80	80	80
$S1$	0	11.6	0.0	0.0	0.0
$S2$	102	81.07	33.07	33.07	33.07
$S3$	246	214	240	160	80
$T1$	2.2	1.95			
$T2$	2.5	2.5	1.25	1.25	1.25
T	8.75				

Table 23. Subcycle 9 and the End Phase of the 168-h Schedule in Example 2

	Sub_cycle 9							*End phase*											
k	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85
TS1	R1		R2	R1		R2	R2	R1			R2	R1		R2	R2			
RI1	80		80	80		80	80	80			80	80.000		80	80			
RI2	50		50	50		50	50	50			50	50.000		50	20			
sum	130		130	130		130	130	130			130	130.000		130	100			
TS2	R3	R4	R4	R3	R3	R4	R4	R3	R3	R3	R4	R3	R4	R4	R4	R3	R3	R3
RII	80	80	80	80	80	80	80	80	80	80	46.7	80	80	80	80	80	80	20
S1	0	0	39	13	13	52	26	0	0	0	39	13	13	52	26	6.0	6.0	6.0
S2	102	102	54	6	6	97	49	1	1	1	92	64	64	107	59	10.98	10.98	10.98
S3	246	166	134	232	152	72	170	268	188	108	28	139	59.3	27.3	125	193.32	113.32	33.32
T1	3.75		2.5	3.75		2.5	2.5	3.75			2.5	3.750		2.5	2.5			
T2	1.25	2.5	2.5	1.25	1.25	2.5	2.5	1.25	1.25	1.25		1.25	2.5	2.5	2.5	1.25	1.25	0.5
T	15							18										

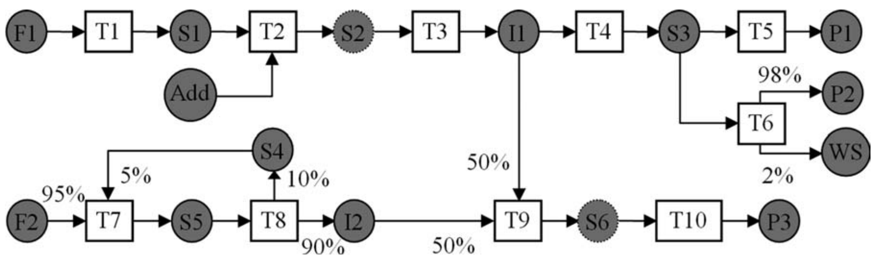

Figure 7. State-task network of Example 3.

Table 24. A Pattern Schedule of Example 3 with a Cycle of 46 h

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
U1	T1		T1		T1		T1		T4		T4		T1		T1		T4		T1		T1		T4		T1	
U2			T2		T2		T2		T2						T2		T2				T2		T2			
U3				T3		T3		T3		T3						T3		T3				T3		T3		
U4											T5		T5						T5						T5	
U5	T7				T7				T9		T7				T9		T7				T9		T7			
U6					T8				T8		T10				T8		T10				T8		T10			
I1	0	0	0	0	5	5	10	10	15	8.5	13.5	8.5	8.5	8.5	7	7	12	7	12	12	10.5	10.5	15.5	10.5	15.5	15.5
I3	0	0	0	0	0	0	2.7	2.7	2.7	1.2	3.9	3.9	3.9	3.9	3.9	2.4	5.1	5.1	5.1	5.1	5.1	3.6	6.3	6.3	6.3	6.3
P1	0	0	0	0	0	0	0	0	0	0	0	0	5	5	10	10	10	10	10	10	15	15	15	15	15	15
P3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	6	6	6	6	6	6	9

k	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	**46**	47	48
U1	T1		T4		T4		T1		T1		T4		T4		T4							
U2	T2		T2						T2		T2											
U3		T3		T3						T3		T3										
U4					T5		T5						T5		T5		T5					
U5	T9			T9			T9			T9			T9			T9						
U6	T8		T10			T10			T10			T10			T10			T10				
I1	14	14	19	14	12.5	12.5	11	11	11	11	14.5	9.5	14.5	8	8	3	1.5	1.5	1.5	1.5		
I3	6.3	4.8	7.5	7.5	6	6	6	4.5	4.5	4.5	3	3	3	1.5	1.5	1.5	0	0	0	0		
P1	20	20	20	20	20	20	25	25	30	30	30	30	30	30	35	35	40	40	**45**	45	45	
P3	9	9	9	9	9	12	12	12	15	15	15	18	18	18	21	21	21	24	24	24	**27**	

Table 25. Part of the Schedule of Example 3 with a Horizon of 48 h

k	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	**46**	47	48	
U1	T1		T4		T4		T1		T1		T4		T4		T1		T4		T4				
U2	T2		T2						T2		T2						T2						
U3		T3		T3						T3		T3						T3					
U4					T5		T5						T5		T5		T5		T5		T5		
U5	T9			T9			T9			T9			T9			T9							
U6	T8		T10			T10			T10			T10			T10			T10					
I1	14	14	19	14	12.5	12.5	11	11	11	11	14.5	9.5	14.5	8	8	8	6.5	1.5	6.5	1.5	1.5	1.5	1.5
I3	6.3	4.8	7.5	7.5	6	6	6	4.5	4.5	4.5	3	3	3	1.5	1.5	1.5	0	0	0	0	0	0	0
P1	20	20	20	20	20	20	25	25	30	30	30	30	30	30	35	35	40	40	40	40	45	45	**50**
P3	9	9	9	9	9	12	12	12	15	15	15	18	18	18	21	21	21	24	24	24	**27**	27	27

Table 26. A 24-h Pattern Schedule for the Combination with the 48-h Pattern Schedule

	4h				**24h**																								
k	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	
U1			T1		T1		T1		T4		T4		T1		T1		T4		T1		T1		T4		T4				
U2					T2		T2		T2						T2		T2				T2		T2						
U3						T3		T3		T3						T3		T3				T3		T3					
U4											T5		T5						T5						T5		T5		
U5	T7				T7				T9			T9		T7				T9			T9			T9					
U6					T8				T8		T10			T10				T8		T10			T10			T10			
I1	0	0	0	0	0	0	5	5	10	3.5	8.5	3.5	2	2	2	2	7	2	5.5	5.5	4	4	9	4	7.5	2.5	2.5	2.5	2.5
I3	0	0	0	0	0	0	2.7	2.7	2.7	1.2	3.9	3.9	2.4	2.4	2.4	2.4	2.4	2.4	0.9	3.6	3.6	2.1	2.1	2.1	0.6	0.6	0.6	0.6	0.6
P1	0	0	0	0	0	0	0	0	0	0	0	0	5	5	10	10	10	10	10	10	15	15	15	15	15	15	20	20	**25**
P3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	6	6	6	6	6	6	9	9	9	9	9	9	**12**

$$P1_{k+1} = P1_k + T5_k \quad (25)$$

$$P3_{k+1} = P3_k + T10_k \quad (26)$$

In the 46-h pattern schedule (Table 24), T1, T2, T3, T4, and T5 are performed with 5 kg for each batch, T7, T8, T9, and T10 are performed with 3 kg for each batch. In such a schedule, U1 works continuously with full load, while U5 and U6 make full use of the time to yield maximum $P3$. As a result, this is a schedule with maximum production (45 + 27 = 72). With minor adjustment of this schedule, the 48-h schedule can be easily obtained (see Table 25). For the 72-h schedule, we synthesize another 24-h schedule (see Table 26) to combine with the 48-h schedule. By using the 48- and 24-h schedules, other schedules with different time horizons can be easily obtained through combination (see Table 27). If refined, the schedules can have a little more production.

It is obvious that the task sequences performed by U5 and U6 is the master sequence, others are slave, and U5 and U6 are the key units.

Example 4

This is also a ''relatively'' complex example from Sundaramoorthy and Karimi,[14] and then studied by Shaik et al.,[17] involving 11 tasks that can be performed in six units involving 13 states. The STN diagram for this example is shown in Figure 8. This problem has several common characteristics of a multipurpose batch plant (i.e., a unit can perform either a single task or multiple tasks; a task can be performed in multiple units, etc.). Additionally, some of the intermediates (S4 and S5) have nonzero initial stock levels and unlimited storage capacity is assumed for all states. The relevant data is shown in Tables A7 and A8. The objective is to maximize the total profit/production of the plant within a given time horizon, namely, 48 h (2 days), 72 h (3 days), 96 h (4 days), 120 h (5 days), 144 h (6 days), and 168 h (7 days).

To get the natural cycle of the example, we have to first identify the master task sequence. Through analysis, it is found that the heater is the bottleneck unit. So the most important work in scheduling this example over long time horizons is how to properly arrange the heating tasks, H1 and H2. The analysis is conducted below in details.

Under the assumption that RI and RT work in parallel and with full load for the same task (R1, R2, or R3), a full batch by both RI and RT is 250 kg. Considering the mass balance with regard to R3 and R2, we have the following equations:

Table 27. Combination of the Schedules in Example 3

TH	Combination: Time	Combination: P
48 h (2 days)	46 + 2	72 + 5 = 77
72 h (3 days)	48 + 24	77 + 37 = 114
96 h (4 days)	48 + 48	77 + 77 = 154
120 h (5 days)	96 + 24	154 + 37 = 191
144 h (6 days)	48 × 3	77 × 3 = 231
168 h (7 days)	144 + 24	231 + 37 = 268

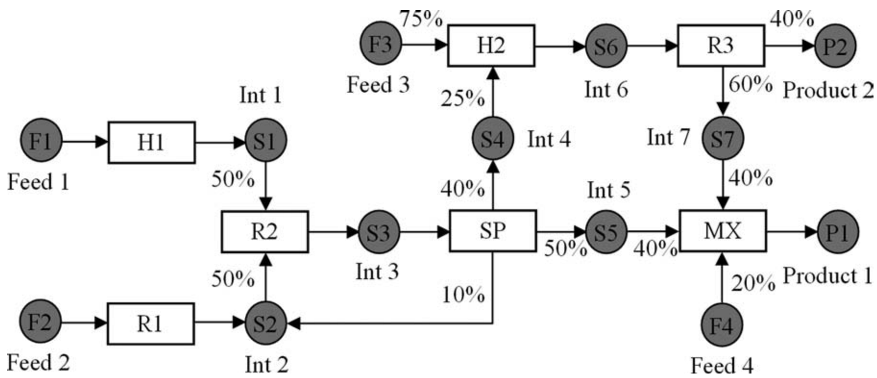

Figure 8. State-task network of Example 4.

$$0.4P_{R2} = 0.25P_{R3} \tag{27}$$

$$P_{R1} = 0.5P_{R2} \tag{28}$$

where P_{R1}, P_{R2}, and P_{R3} are the total material amount of reactions R1, R2, and R3, respectively. The rationale behind Eq. 27 can be explained as follows. All the intermediate production from R2 is put into S3 and then through SP separated into three streams, 40% being put into S4. That means $0.4P_{R2}$ is put into S4. On the other hand, among the material for R3, 25% is from S4, which means $0.25P_{R2}$ is from S4. Hence, $0.4P_{R2} = 0.25P_{R3}$ can let S4 return to its initial state. From these two equations, we have (1) $P_{R2}/P_{R3} = 5/8$, which means five full batches of R2 (5 × 250 kg) need eight full batches of R3 (8 × 250 kg) for mass balance, (2) $P_{R1}/P_{R2} = 1/2$. Accordingly, we could have a scenario A: six full batches of R1 (with 6 × 250 kg), 15 full batches of R2 (with 15 × 250 kg), and 24 full batches of R3 (with 24 × 250 kg). Under such a scenario, (24×250/100) = 60 full batches of H2 and (15 × 250/2/100) ≈ 19 full batches of H1 have to be completed by the heater, totally needing 145 h; but the working time of both RI and RT to complete reaction batches is 100 h. We can see that the heating time (145 h) is far more than the reaction time (100 h), hence, the heating task sequence is the master sequence. Under scenario A, the total production $P = P1 + P2 = 24 \times 250 \times 0.4 + 15 \times 250 \times 0.5/0.4 = 7087.5$ kg, the average production over tome is to be less than 7087.5/145 = 48.88 kg/h.

Considering the mass balance with regard to MX and R2, we have the following equations:

$$0.5P_{R2} = 0.6P_{R3} \tag{29}$$

$$P_{R1} = 0.5P_{R2} \tag{30}$$

we have (1) $P_{R2}/P_{R3} = 6/5$, which means six full batches of R2 (5 × 250 kg) need five full batches of R3 (8 × 250 kg) for mass balance, (2) $P_{R1}/P_{R2} = 1/2$. Therefore, we could have a scenario B: three full batches of R1 ($P_{R1} = 3 \times 250$ kg), six full batches of R2 ($P_{R2} = 6 \times 250$kg), and five full batches of R3 ($P_{R3} = 5 \times 250$ kg). Under such a scenario, (5 × 250/100) ≈ 13 full batches of H2, and (6 × 250/2/100) ≈ 8 full batches of H1 have to be completed by the heater, totally needing 36.66 h; however, the working time of both RI and RT to complete reaction batches is 29.34 h. Still the heating time (36.66 h) is far more than the reaction time (29.34 h), hence, the heating task sequence is still the master sequence. Under scenario B, the total production $P = P1 + P2 = 5 \times 250 \times 0.4 + 6 \times 250 \times 0.5/0.4 = 2375$ kg, the average production over time is to be less than 2375/36.66 = 64.77 kg/h. Therefore, scenario B can yield more production than scenario A within the same duration.

Under scenario B, the initial states in S4 (50 kg)and S5 (50 kg), and the recycle stream from SP to S2 have not been taken into account. If these taken into account, we could have a scenario C: three full batches of R1 (with $P_{R1} = 3 \times 250$ kg), seven full batches of R2 ($P_{R2} = 7 \times 250 = 1750$ kg, thus $P_{S4} = 7 \times 250 \times 0.4 + 50 = 750$ kg), and six full batches of R3 ($P_{R3} = 6 \times 250$kg), nine full batches of H1 ($P_{H1} = 9 \times 100$ kg), 15 full batches of H2 ($P_{H2} = 15 \times 100$ kg), $P1 = 7 \times 250 \times 0.5/0.4 = 2187.5$ kg, $P2 = P_{R3} \times 0.4 = 6 \times 250 \times 0.4 = 600$ kg, $P = P1 + P2 = 2787$ kg. The heating time is (15 × 2 + 9 × 1.333) = 42 h. So the average production over time is to be less than 2787/42 = 66.35 kg/h. Therefore, scenario C is a little better than B.

On the basis of the above analysis, we use scenario C to guide the synthesis of the pattern schedule, where the following heuristics are applied: (1) The heater (H) makes full use of time for two heating tasks, H1 and H2. (2) Considering that RI and RT have enough time and capacity to accomplish their assignment, it is not necessary for RI and RT to carry out the tasks (R1, R2, or R3) in parallel or successively. (3) SP works in time to consume the material in S3. (4) M1 and M2 works in time to consume the material in S5 and S7. When the material in S5 returns to zero, a natural cycle occurs. (5) To maximize P2, try best to carry out as many batches of R3 as possible in the natural cycle.

Consequently, a pattern schedule is obtained as shown in Table 28, where it can be noted that the heater H works continuously with full batches, and three full batches of R1, seven full batches of H1, 18 full batches of H2 are witnessed, the duration of the natural cycle is exactly 48 h, $P = P1 + P2 = 2595$ kg, thus the actual average production over time is 2595/48 = 54.06 kg/h.

To obtain the 72-h schedule, another 24-h schedule (see Table 29) has been synthesized to combine with the 48-h schedule. Because of intermediate materials prepared at the end of the 48 h, the 24-h schedule has a larger average production over time, 1342.5/24 = 55.94 kg/h.

By using the 48- and 24-h schedules, the production of other schedules with different time horizons can be easily

Table 28. A Pattern Schedule of Example 4 with a Cycle of 48 h

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
H	H1(2*100)	H1(100)	H2(2*100,4h)		H1(100)	H2(2*100,4h)		H2(2*100,4h)		H1(100)	H1(100)	H2(2*100,4h)		H1(100)	H2(2*100,4h)		H2(2*100,4h)		H2(2*100,4h)		H2(2*100,4h)		H2(2*100,4h)			
RI	R1(100)	R2(100)	R2(100)	R1(100)	R2(100)	R2(100)	R1(100)	R3(100)				R2(100)			R2(100)	R3(100)			R3(100)		R3(100)		R3(100)		R3(100)	
RT	R1(150)	R2(150)	R2(150)	R1(150)		R2(100)	R1(150)	R3(150)	R3(150)		R2(150)	R2(150)	R3(150)		R2(100)	R3(150)			R3(150)		R3(150)		R3(150)		R3(150)	
SP			250(3.667h)		250(3.667h)		300(4h)						300(4h)			300(4h)										
M1									200		200		200	200				200	200		200		200		200	
M2																									75	
S1	0	200	175	50	50	100	0	0	0	0	100	125	0	0	100	0	0	0	0	0	0	0	0	0	0	0
S2	0	250	125	0	275	225	150	400	430	430	355	355	230	230	260	160	160	190	190	190	190	190	190	190	190	190
S3	0	0	250	250	0	100	300	0	0	0		150	400	100	100	300	0	0	0	0	0	0	0	0	0	0
S4	**50**	50	50	0	100	100	150	100	220	220	220	170	170	170	290	240	190	310	310	260	260	210	210	160	160	160
S5	**50**	50	50	50	175	175	300	300	450	370	370	290	290	210	280	280	280	430	350	270	270	190	190	110	110	0
S6	0	0	0	0	200	200	200	400	150	200	200	200	200	250	250	250	200	200	400	150	350	100	300	50	250	0
S7	0	0	0	0	0	0	0	0	150	70	160	80	80	90	10	10	160	160	80	150	150	220	220	290	290	330
P1	0	0	0	0	0	0	0	0	0	0	200	200	400	600	600	800	800	800	1000	1200	1200	1400	1400	1600	1600	**1875**
P2	0	0	0	0	0	0	0	0	100	100	160	160	160	220	220	220	320	320	320	420	420	520	520	620	620	**720**
Tk	2.667	1.333	1.333	2.667	1.333	1.333	2.667	2.667	1.333	1.333	1.333	1.333	2.667	1.333	1.333	2.667	1.333	2.667	2.667	1.333	2.667	1.333	2.667	1.333	2.667	
T	**48h**																									

estimated. If refined, the schedules should yield more production.

In scheduling this example, it is found that the bottleneck unit is presently the heater (H). Because of limited capability of the heater for H2, the production capabilities of RI, RT, SP, and two mixers M1 and M2 are not sufficiently utilized. If the heater is expanded with enough capability or with enough short heating time, RI and RT will become the key units, and then the master task sequence is to be the one of R1, R2, and R3. As a consequence, the production capability of the plant can be enhanced a lot.

Discussion

From the examples studied, the following points can be observed:

(1) In synthesizing the pattern schedules, master sequences and slave sequences can be distinguished, and the relevant bottleneck/key units can be identified. In example 1, the sequence of R1, R2, and R3 is the master one; RI and RT are the key units. In Example 2, the sequence of R1 and R2 is the master one; RI1 and RI2 are the key units; although RII has to perform R3 and R4, it has enough time to fulfill all the tasks. In Example 3, the task sequence performed by U5 and U6 is the master one, U5 and U6 are the key units. However, in Example 4, the situation is different—the heater is the bottleneck unit.

(2) With the conditions of a batch plant varying, the natural periodicity of the production schedule is changed. In the example[25,26] modified from Example 1, owing to the recycle of the filter removed from Example 1, a complete natural cycle is obtained. The only difference between Example 2 and the example[25,26] modified from Example 2 is that the process times of R1 and R2 in the latter are changed, but this changes the scheduling cycle entirely.

(3) To satisfy complete natural cycles with identical task sequences, the capacities of some intermediate tanks need to be expanded (as shown in Example 1). However, a complete

Table 29. A 24-h Schedule for Combination in Example 4

k	25	26	27	28	29	30	31	32	33	34	35	36	37	
H	**H1(2*100)**	**H1(100)**	H2(2*100,4h)		H1(100)	H1(100)	H2(2*100,4h)		H1(2*100)	H2(2*100,4h)		H2(100)		
RI	R3(100)	**R2(100)**	R1(100)	R2(100)	R2(100)	R2(100)	R2(100)		R3(100)	R1(100)	R2(100)	R3(100)	R3(100)	
RT	R3(150)	**R2(150)**	R1(150)	R2(150)		R2(100)		R3(150)	R3(150)	R1(150)	R2(150)	R3(100)		
SP			250(3.667h)				300(4h)		300(4h)			300(4h)		
M1	200				200				200	200		200	62.5	
M2	75											200		
S1	0	**200**	175	175	50	100	100	50	50	250	250	125	125	125
S2	190	190	65	315	215	165	65	15	45	45	325	200	200	230
S3	0	0	250	0	250	350	550	350	350	50	50	300	0	0
S4	160	160	160	110	210	210	210	160	280	280	350	350	325	445
S5	110	0	0	0	125	45	45	45	195	115	185	185	25	150
S6	250	0	0	0	200	200	200	200	250	0	0	200	100	0
S7	290	330	330	330	330	250	250	250	340	410	330	330	290	325
P1	1600	**(1875)**					200	200	200	400	600	600	1000	**1062.5**
P2	620	**(720)**							60	160	160	160	240	**280**
Tk	2.667	1.333	2.667	1.333	1.333	1.333	1.333	2.667	2.667	2.667	1.333	2.667	2.667	
T		**24h**												

natural cycle is not a must for the pattern schedule. In case of the limited capacities of some intermediate tanks, as shown in pattern schedule I of Example 1, a complete natural cycle is impossible.

(4) Recycle streams (see Examples 1 and 4), initial stock of some intermediate tanks (Examples 3 and 4) can save batches of some tasks for the relevant materials and lead to more production.

(5) Some storage policies, like ZW and NIS in Example 3, can make the scheduling mission easy. ZW is the special case of NIS and FIS.

(6) The proposed method is basically a two-phase technique. Although in Example 2 three phases are involved in the pattern schedule, the long time horizon of the problem to be solved is still cut into two phases.

As a matter of fact, another obvious advantage of the pattern schedule is that the makespan with a given product demand can be quickly estimated from the pattern schedule. Moreover, with minor modification, the novel decomposition method can be applied to the minimization of makespan with fixed product demand. With the estimation of the makespan, we can still cut it into two phases: phase 1 is copied from the pattern schedule, completing most of the given product demand, but phase 2 involves a subproblem to minimize makespan for the remainder of the product demand.

For all the examples in this article, production maximization is equivalent to profit maximization. We should admit that the situation may change for other cases. Scheduling heuristics are objective dependant. In other cases, one has possibly to use heuristics for profit maximization.

In this work, we have ignored the utility requirements and constraints, but we don't think it is a limitation of our approach. If we consider the utility requirements and constraints, then every time to assign batches to a time slot, we should calculate the utility amounts and restrict them within the corresponding maximum availabilities. This is similar to the consideration of intermediate state consumption, replenishment, and constraints.

Conclusion

In this study, we have proposed an effective heuristic method for large-scale multipurpose process scheduling, mainly via manual operation, rather than using exact mathematical programming models which are often intractable.

With the master sequences and the bottleneck/key units identified, a pattern schedule with certain periodicity can be synthesized for repeated use. And then a novel decomposition method has been applied: the overall scheduling problem within a long time horizon is broken into two subproblems of two phases. The subschedule in phase 1 is duplicated from the template of the pattern schedule and needs no more optimization, and the subschedule in phase 2 is a small-size problem that can be optimized easily by the proposed heuristic method. Between phase one and phase two, there is a cut point which can be determined by analysis and reasoning.

In solving a small-size subproblem, a short task sequence is first identified by using the scheduling heuristics and the search tree. By this way, a large number of binary variables and corresponding constraints for the time sequences of the tasks are avoided. And then merely the batch sizes of the tasks are optimized by using a solver. This heuristic method does not require complex programming or MILP models.

The results of case study indicate that (1) the increase of problem size does not lead to the increase of computational time and complexity, (2) the proposed method performs better than the traditional decomposition techniques and the cyclic scheduling methods, and (3) the proposed method even performs better than the genetic algorithm in solving the problems with constant processing times. Therefore, the proposed method is applicable to the large-scale industrial scheduling of multipurpose batch plants.

Acknowledgments

The authors gratefully acknowledge financial support from Hong Kong's Research Grant Council grants 614005 and 614307. The authors sincerely thank the anonymous reviewers for their constructive comments which are very helpful to the improvement of this work.

Literature Cited

1. Reklaitis GV, Sunol AK, Rippin DWT, Hortacsu O, editors. *Batch Processing Systems Engineering*. Berlin: Springer, 1996.
2. Günther HO, van Beek P. *Advanced planning and scheduling in process industry*. In: Günther HO, van Beek P, editors. *Advanced Planning and Scheduling Solutions in Process Industry*. Berlin: Springer, 2003:1–9.
3. Floudas CA, Lin X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng*. 2004;28:2109–2129.
4. Kondili E, Pantelides CC, Sargent RWH. A general algorithm for short-term scheduling of batch operations. Part 1. MILP formulation. *Comput Chem Eng*. 1993;17:211–227.
5. Pantelides CC. *Unified frameworks for optimal process planning and scheduling*. In: Rippin DWT, Hale JC, Davis J, editors. *Proceedings of the Second International Conference on Foundations of Computer-aided Process Operations*, Crested Butte, Colorado, 1993:253–274.
6. Shah N, Pantelides CC, Sargent RWH. Optimal periodic scheduling of multipurpose batch plants. *Annu Oper Res*. 1993;42:193–228.
7. Schilling G, Pantelides CC. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput Chem Eng*. 1996;20:S1221–S1226.
8. Zhang X, Sargent R. The optimal operation of mixed production facilities general formulation and some approaches for the solution. *Comput Chem Eng*. 1996;20:897–904.
9. Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind Eng Chem Res*. 1998;37:4341–4359.
10. Mockus L, Reklaitis GV. Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Ind Eng Chem Res*. 1999;38:197–203.
11. Lee KH, Park HI, Lee IB. A novel nonuniform discrete time formulation for short-term scheduling of batch and continuous processes. *Ind Eng Chem Res*. 2001;40:4902–4911.
12. Giannelos NF, Georgiadis MC. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. *Ind Eng Chem Res*. 2002;41:2178–2184.
13. Maravelias CT, Grossmann IE. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res*. 2003;42:3056–3074.
14. Sundaramoorthy A, Karimi IA. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem Eng Sci*. 2005;60:2679–2702.
15. Maravelias CT, Grossmann IE. Minimization of the makespan with a discrete-time state-task network formulation. *Ind Eng Chem Res*. 2003;42:6252–6257.
16. Castro PM, Barbosa-Povoa AP, Matos HA, Novais AQ. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind Eng Chem Res*. 2004;43:105–118.
17. Shaik MA, Janak SL, Floudas CA. Continuous-time models for short-term scheduling of multipurpose batch plants: a comparative study. *Ind Eng Chem Res*. 2006;45:6190–6209.

18. Bassett MH, Pekny JF, Reklaitis GV. Decomposition techniques for the solution of large-scale scheduling problems. *AIChE J*. 1996;42:3373–3387.
19. Wu D, Ierapetritou MG. Decomposition approaches for the efficient solution of short-term scheduling problem. *Comput Chem Eng*. 2003;27:1261–1276.
20. Schilling G, Pantelides CC. Optimal periodic scheduling of multipurpose plants. *Comput Chem Eng*. 1999;23:635–655.
21. Castro P, Barbosa-Povoa A, Matos H. Optimal periodic scheduling of batch plants using RTN-based discrete and continuous-time formulations: a case study approach. *Ind Eng Chem Res*. 2003;42:3346–3360.
22. Wu D, Ierapetritou MG. Cyclic short-term scheduling of multiproduct batch plants using continuous-time representation. *Comput Chem Eng*. 2004;28:2271–2286.
23. Hoffman KL. Combinatorial optimization: current successes and directions for the future. *J Comput Appl Math*. 2000;124:341–360.
24. He Y, Hui CW. A binary coding genetic algorithm for multi-purpose process scheduling: A case study. *Chem Eng Sci*. doi:10.1016/j.ces.2010.05.032 (In Press).
25. He Y. Research on meta-heuristic methods for large-scale complex scheduling in process industry. PhD dissertation, Hong Kong University of Science and Technology, 2007.
26. He Y, Hui CW. *Meta-Heuristics for Large-Scale Process Scheduling*. Saarbrücken, Germany: VDM Verlag, 2009.
27. Papageorgiou LG, Pantelides CC. Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 2. Mathematical decomposition approach. *Ind Eng Chem Res*. 1996;35:510–529.
28. Janak SL, Lin X, Floudas CA. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. *Ind Eng Chem Res*. 2004;42:2516.

Appendix: Problem Data of the Examples

Table A1. State-Relevant Data for Example 1

	F1	F2	F3	SA	S1 (BC)	S2 (AB)	S3 (IE)	P1	P2
C_s (kg)	UL	UL	UL	100	150	200	200	UL	UL
$S0_s$ (kg)	AA	AA	AA	–	–	–	–	–	–
PR_s ($/kg)	0	0	0	0	0	0	0	10	10

UL, unlimited; AA, available as and when required.

Table A2. Task-Relevant Data for Example 1 with CPT and VPT

Task	Unit	τ (CPT)	α	β	B^{MAX}	Batch Time Range
Heating (H)	Heater (H)	1	0.667	0.00667	100	[0.667, 1.333]
Reaction 1 (R1)	Reactor I (RI)	2	1.333	0.02664	50	[1.333, 2.666]
	Reactor II (RT)	2	1.333	0.01665	80	[1.333, 2.666]
Reaction 2 (R2)	Reactor I (RI)	2	1.333	0.02664	50	[1.333, 2.666]
	Reactor II (RT)	2	1.333	0.01665	80	[1.333, 2.666]
Reaction 3 (R3)	Reactor I (RI)	1	0.667	0.01332	50	[0.667, 1.333]
	Reactor II (RT)	1	0.667	0.008325	80	[0.667, 1.333]
Separation (SP)	Filter (FLT)	2	1.333	0.00667	200	[1.333, 2.666]

Table A3. State-Relevant Data for Example 2

	F1	F2	S1	S2	S3	P1	P2
C_s (kg)	UL	10000	100	200	500	UL	UL
$S0_s$ (kg)	AA	AA	–	–	–	–	–
PR_s ($/kg)	0	0	0	0	0	30	40

UL, unlimited; AA, available as and when required.

Table A4. Task-Relevant Data for Example 2 with VPT

		R1		R2		R3		R4	
Task	B^{MAX}	α	β	α	β	α	β	α	β
RI1	80	0.75	0.0375	0.5	0.025	–	–	–	–
RI2	50	0.75	0.060	0.5	0.040	–	–	–	–
RII	80	–	–	–	–	0.25	0.0125	0.5	0.025

Table A5. State-Relevant Data for Example 3

	F1	F2	S1	S2	S3	S4	S5	S6	I1	I2	P1	P2	P3
C_s (kg)	UL	UL	0(ZW)	0(NIS)	15	40	0(ZW)	0(NIS)	UL	UL	UL	UL	UL
$S0_s$ (kg)	AA	AA	0	0	0	10	0	0	0	0	0	0	0
PR_s ($/kg)	0	0	0	0	0	0	0	0	0	0	1	1	1

UL, unlimited; AA, available as and when required.

Table A6. Task-Relevant Data for Example 3 with CPT

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Unit	U1	U2	U3	U1	U4	U4	U5	U6	U5	U6
B^{MAX}	5	8	6	5	8	8	3	4	3	4
τ(CPT)	2	1	1	2	2	2	4	2	2	3

Table A7. State-Relevant Data for Example 4

	F1	F2	F3	F4	S1	S2	S3	S4	S5	S6	S7	P1	P2
C_s (kg)	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL
$S0_s$ (kg)	AA	AA	AA	AA	0	0	0	50	50	0	0	0	0
PR_s (\$/kg)	0	0	0	0	0	0	0	0	0	0	0	5	5

UL, unlimited; AA, available as and when required.

Table A8. Task-Relevant Data for Example 4 with VPT

Task	Unit	α	β	B^{MIN}	B^{MAX}	Batch Time Range
Heating 1 (H1)	Heater (H)	0.667	0.00667		100	[0.667, 1.333]
Heating 2 (H2)	Heater (H)	1.000	0.01000		100	[1.000, 2.000]
Reaction 1 (R1)	Reactor I (RI)	1.333	0.01333		100	[1.333, 2.667]
	Reactor II (RT)	1.333	0.00889		150	[1.333, 2.667]
Reaction 2 (R2)	Reactor I (RI)	0.667	0.00667		100	[0.667, 1.333]
	Reactor II (RT)	0.667	0.00445		150	[0.667, 1.333]
Reaction 3 (R3)	Reactor I (RI)	1.333	0.01333		100	[1.333, 2.667]
	Reactor II (RT)	1.333	0.00889		150	[1.333, 2.667]
Separation (SP)	Separator (SP)	2.000	0.00667		300	[2.000, 4.000]
Mixing (MX)	Mixer 1 (M1)	1.333	0.00667	20	200	[1.466, 2.667]
	Mixer 2 (M2)	1.333	0.00667	20	200	[1.466, 2.667]

Manuscript received Mar. 23, 2010, and revision received Apr. 28, 2010.