



# Incorporating heat integration in batch process scheduling

R. Adonyi<sup>a</sup>, J. Romero<sup>b</sup>, L. Puigjaner<sup>b</sup>, F. Friedler<sup>a,\*</sup>

<sup>a</sup> Department of Computer Science, University of Veszprém, Egyetem u. 10, Veszprém H-8200, Hungary

<sup>b</sup> Department of Chemical Engineering, Universitat Politècnica de Catalunya, Av. Diagonal 647, Barcelona 08028, Spain

Received 10 April 2003; accepted 17 April 2003

---

## Abstract

This work presents a methodology for incorporating heat integration in batch process scheduling. The formerly introduced S-graph approach [AIChE J. 48 (2002) 2557–2570] for solving multipurpose scheduling problems is proved to be appropriate for the representation of both scheduling problems and the corresponding heat exchanger network synthesis problem. The proposed procedure is based on the branch-and-bound framework, where two optimisation problems, the scheduling and the heat integration one, are considered simultaneously instead of consecutively. This method is primarily based on combinatorics and combinatorial algorithms.

Solution of several examples illustrates the efficiency of the proposed approach and the benefits of considering heat exchanger network synthesis while scheduling batch processes.

© 2003 Elsevier Ltd. All rights reserved.

*Keywords:* Batch process scheduling; Heat integration; S-graph

---

## 1. Introduction

The high-level heating and cooling duties in a batch process may require heat integration for reducing the cost of utilities. The lack of continuous availability of cold and hot streams prevents the direct application of heat integration methodologies developed for continuous plants.

Corominas et al. [2] and Font et al. [3] has solved the problem of energy integration in batch plants for a given production schedule, therefore, the overall problem is decomposed into two sequentially solved problems of scheduling and heat integration. Since the level of heat integration depends on the production schedule, this simple decomposition may result in poor heat integration. Conversely, if the heat integration problem is solved first to find an optimal heat

---

\* Corresponding author. Tel.: +36-88-424-483; fax: +36-88-428-275.

E-mail address: [friedler@dcs.vein.hu](mailto:friedler@dcs.vein.hu) (F. Friedler).

integration, it can easily result in an unsolvable scheduling problem or a scheduling problem with an extremely high makespan. Consequently, simple decomposition of the problem into heat integration and scheduling may not provide an acceptable solution; the integrated problem is to be solved. Theoretically, an integrated mathematical model can be formulated. Since heat integration and batch process scheduling are significantly different types of optimisation problems with different objective functions, their combination cannot be solved by available solvers as a multi-objective optimisation problem. Another opportunity is the implicit decomposition into a system of master and slave subproblems. In the present work scheduling is considered to be the master while heat integration is its slave problem.

Production scheduling has received significant attention [4–6] in the literature. General purpose mathematical models, based on either MILP or MINLP formulations, run out of proportion when trying to solve problems far more complex than the straight scheduling ones, i.e., when energy integration wants to be considered. In this paper we propose a new graph theoretic approach to efficiently solve the scheduling of multiproduct/multipurpose batch plants maximizing energy integration. Specifically, the graph theoretical approach for solving the scheduling of multipurpose batch plants [1,7], called S-graph representation, is extended to consider heat exchanger networks. The S-graph representation has the advantage of exploiting the problem-specific knowledge from the very beginning to develop efficient algorithms. This performance is used in this work to derive an effective algorithm for solving scheduling problems with energy integration. In this paper, our goal is to integrate heat integration and scheduling to determine a solution that requires minimal utility and satisfies a constraint on the makespan.

## 2. S-graph framework for batch process scheduling

A multipurpose batch scheduling problem is defined by the recipe or recipes, the amount to be produced from each product, and the plausible tasks to equipment units assignments. A recipe can be conventionally represented by a directed graph; e.g., on Fig. 1, the graph shows a recipe with three consecutive reaction stages, where nodes represent the production tasks and the arcs precedence relationships among them. The production time (PT) and the set of plausible equipment units (Eq) of a task are given at the corresponding node.

Batch process scheduling problems can be classified according to the properties of the intermediate materials, the equipment units, and the rules of material transfers among them. One of the major classes follows the nonintermediate storage (NIS) policy. Within this policy the intermediate materials have to be stored in the equipment units, and so an equipment unit is not useable for the next task until the intermediate materials stored in the equipment unit are not transferred to the equipment unit assigned to the next task in the recipe.

A recipe can be either simple, i.e., the production is a sequence, or complex, i.e., the production may include parallel lines and branches. Even though the methodology given in the present paper



Fig. 1. Conventional representation of a recipe.

may consider scheduling problems of complex recipes, the methodology will be described here for simple recipes.

The S-graph framework [7] is an effective graph representation and algorithm designed for the NIS case. The nodes of an S-graph represent the tasks of the recipe and its arcs; represent the precedence relationships of the tasks (recipe-arc) and the order of the application of an equipment unit (schedule-arc).

An S-graph is given in the form of  $G(N, A_1, A_2)$ , where  $N, A_1, A_2$  denote the set of nodes, the set of the recipe-arcs and the set of schedule-arcs, respectively. A nonnegative value,  $c(i, j)$ , is assigned to each arc. In practice, if an arc is established from node  $i$  to node  $j$ , the task corresponding to node  $j$  cannot start its activity earlier than  $c(i, j)$  time after the task represented by node  $i$  started. Since the arcs of an S-graph represent precedence in time, an S-graph that represents a recipe or a schedule of a problem, always acyclic. There are two specific S-graphs, the recipe-graph and the schedule-graph. The former represents the recipe, the later a solution of a scheduling problem. A recipe-graph defines the order and type of tasks, the material transfers among them, and the set of plausible equipment units available for each task. The recipe-graph of a recipe can be simply generated by assigning one node to each task (task node) and one to each product of the recipe. An arc is established between the nodes of the consecutive tasks of the recipe, and between a task node and a product node if the corresponding task generates the corresponding product. The weight of an arc is specified by the processing time of the task corresponding to the initial node of the arc; if more than one equipment unit is available for this task, the weight of the arc is the minimum of the processing times of all plausible equipment units. If  $G(N, A_1, A_2)$  is a recipe-graph, set  $A_2$  is empty ( $A_2 = \emptyset$ ). Fig. 2 illustrates the recipe-graph for three products; A, B, and C; each product is created by three consecutive tasks. The task nodes are labelled by 1 through 9 and product nodes by 10, 11, and 12. Sets S1 through S9 contain the equipment units that can be assigned to the corresponding task of the recipe.

S-graph  $G'(N, A_1, A_2)$  is called a schedule-graph of recipe-graph  $G(N, A_1, \emptyset)$  if all tasks represented in the recipe-graph has been scheduled by taking equipment-task assignment into account. The S-graph framework can handle the NIS policy effectively. According to the NIS case an equipment unit is not applicable for its next task until the intermediate material of its current task has not been transferred to another equipment unit. This constraint on the equipment units can be expressed by the way the schedule-arcs are added to the S-graph. Let  $\tau_j$  denote the set of tasks that follow task  $j$  according to the recipe. If equipment unit  $E_i$  is assigned to task  $k$  after completion of

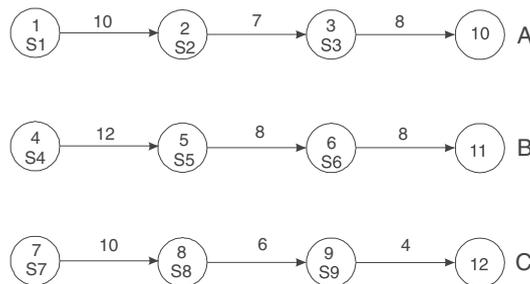


Fig. 2. Recipe-graph for three products.

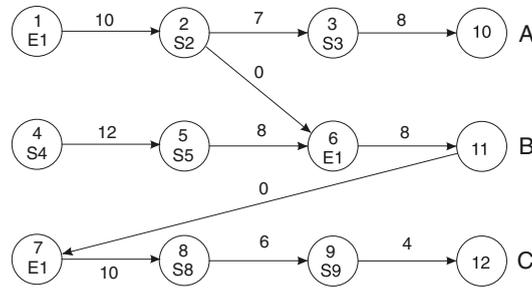


Fig. 3. S-graph representation of task sequence for equipment unit E1.

task  $j$ , then a changeover time weighted arc is established from each element of  $\tau_j$  to node  $k$ . This representation ensures that all acyclic schedule-graphs fulfill the requirements of the NIS policy. Fig. 3 illustrates a schedule for equipment unit E1, that performs task 1 first, then task 6, and finally task 7.

S-graph  $G'(N, A_1, A_2)$  is defined to be a schedule-graph for recipe-graph  $G(N, A_1, \emptyset)$ , if all tasks are scheduled with feasible tasks-equipment units assignments. A schedule-graph has to satisfy some simple properties. The recipe-graph is always a subgraph of any of its schedule-graphs with identical sets of nodes. Each arc of a schedule-graph that does not belong to the recipe-graph is a schedule-arc. Extending the recipe-graph in all possible directions can result all schedule-graphs.

Because of its combinatorial characteristics, a branch-and-bound (B&B) procedure may generate the optimal schedule of a scheduling problem, i.e., the schedule-graph that corresponds to the minimal makespan. The recipe-graph with no equipment unit assignment serves as the root of the enumeration tree of the B&B procedure. At any partial problem, one equipment unit is selected and then all child partial problems are generated through the possible assignments of this equipment unit to unscheduled nodes [7].

The bounding procedure tests the feasibility of a partial problem. If this test is positive, it determines the lower bound for the makespan of all solutions that can be derived from this partial problem simply by using the well-known longest path algorithm [8].

**Example 1.** For recipe-graph given in Fig. 4, and for sets of equipment units  $S1 = \{E1\}$ ,  $S2 = \{E3\}$ ,  $S3 = \{E2\}$ ,  $S4 = \{E2\}$ ,  $S5 = \{E3\}$ ,  $S6 = \{E1\}$ ,  $S7 = \{E1\}$ ,  $S8 = \{E2\}$ , and for  $S9 = \{E3\}$ , Fig. 5 illustrates the optimal schedule in the form of a schedule-graph, and Fig. 6 illustrates the corresponding Gantt chart. The makespan, determined by the longest path algorithm, is 63.

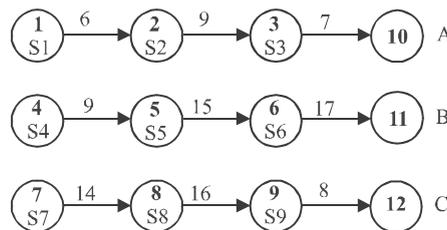


Fig. 4. Recipe-graph of Example 1.

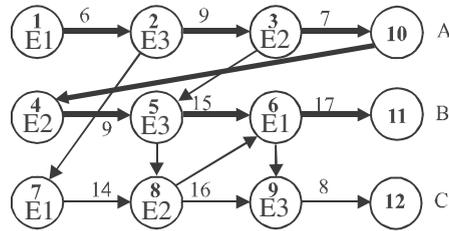


Fig. 5. Schedule-graph of the optimal solution. Bold line shows the longest path, its length determines the makespan (63).

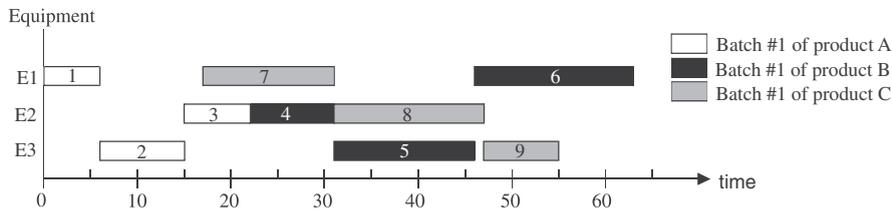


Fig. 6. Gantt chart of the optimal solution of Example 1.

### 3. Heat integration in batch processes

By nature, batch process scheduling and heat integration are two significantly different highly complex optimisation problems. Many algorithmic and heuristic based methods exist for solving heat integration problems by resorting to pinch technology [9], superstructure based mixed integer programming [10–12], and integration with process network synthesis [13]. These methods have been developed for continuous processes where scheduling is obviously of no concern.

In principle, these two different problems can be solved sequentially, i.e., solving scheduling first and then heat integration or vice versa. Since the solution of one of them influences the other, the result of this simplistic approach is usually very poor. Consequently, an integrated consideration of scheduling and heat integration may provide appropriate solution. Since no method known up to now for solving the integrated model, the development of a new method is desired for effective design and operation of batch processes. To do so, the main issue is how to operate tasks with potential heat exchange simultaneously without sacrificing the quality of the solution of scheduling.

#### 3.1. Problem formulation

The optimal solutions of a scheduling problem, i.e., solutions with minimal makespan, may not provide enough flexibility for the simultaneous operation of the tasks that is required for heat exchange between them. Therefore, higher degree of freedom is necessary for heat integration in affecting scheduling for heat integration. To increase the flexibility of the scheduling problem an upper bound is given for the makespan in the form of a constraint, instead of looking for the minimal makespan solution. Then, we are searching for a schedule that requires minimal utility and its makespan satisfies the constraint.

If the temperature of the input stream of a task is to be altered by heat exchange, either by utility or by other stream, the time period while the input stream is heated or cooled, is called heat-exchange phase of the task. The heat-exchange phase is followed by the active phase when the equipment unit starts its designated activity. It is supposed that at most two heat exchangers can be fitted to a feed stream of an equipment unit, one heat exchanger for the heat exchange with the feed of another task and one for the utility. Thus, at most one hot or cold stream can be assigned to the input stream of a task in addition to the potential use of utility. The area of every heat exchanger is assumed to be identical and prespecified; therefore, the rate of heat flow depends only on the temperature difference of the hot and cold streams. Thus, the time required for the heat exchange phase of a task can be determined for every pair of streams independently of the scheduling.

The integrated problem can be specified by the recipe together with the thermodynamic data of the streams (temperatures, heat transfer coefficients, etc.). The recipe is conveniently given by a recipe-graph in the S-graph framework. Fig. 7 illustrates the recipe-graph of products A and B, where the heating or cooling duty associated with a task is denoted by a vertical arrow at the corresponding node; heating is indicated by an arrow pointing upward, and the cooling pointing downward. For example, the input stream of task 2 of Fig. 7 is to be cooled.

The main difficulty of the scheduling part of the integrated procedure is to determine the candidate heat exchanger units. Even if a hot and a cold stream satisfy the thermodynamic criteria for heat exchange, they can only be matched if they are simultaneous in time.

A feasible solution of a simple scheduling problem is given in Fig. 8, this solution excludes the potential heat exchange between task 3 and 6. If the upper limit for the makespan of the integrated problem is 18, there is a solution that provides simultaneous operation of task 3 and 6 for the heat exchange (see Fig. 9).

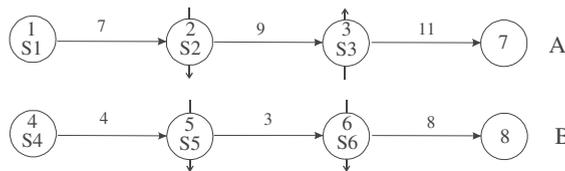


Fig. 7. Recipe-graph showing the heating and cooling duties.

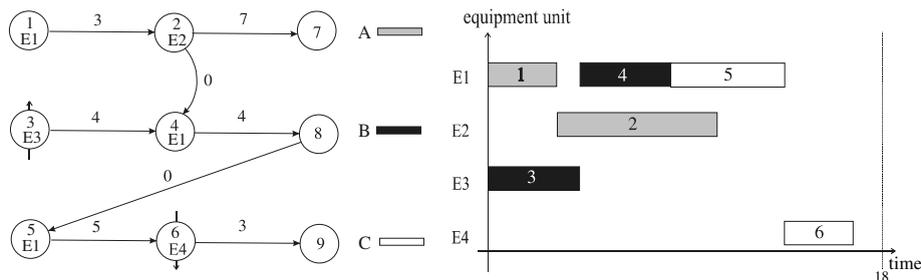


Fig. 8. Schedule where the heating and cooling duties of task 3 and 6 cannot be matched.

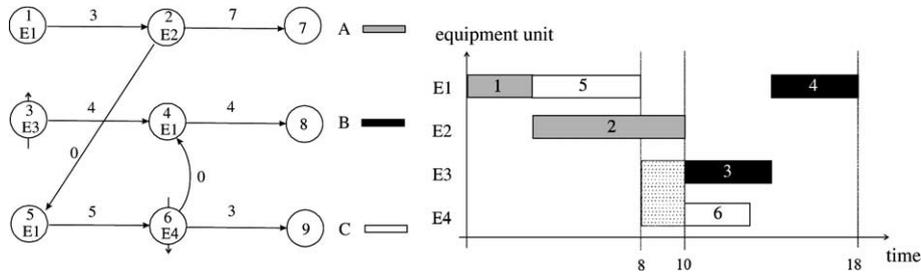


Fig. 9. Schedule where the heating and cooling duties of task 3 and 6 can be matched (dotted area of the Gantt chart).

### 3.2. Procedure for the scheduling problem with heat integration

The basic scheduling algorithm of the S-graph framework was developed to find a schedule with minimal makespan. In the integrated method such a schedule is to be generated, that satisfies the constraint on the makespan and provides the minimal cost heat-exchanger network. The integrated procedure has been based on three main components; the basic algorithm of the S-graph framework, the time interval management, and the determination of the utility cost of a heat-exchanger network.

The proposed algorithm follows the B&B framework. The branching procedure is responsible for scheduling the equipment units. The bounding procedure tests the feasibility of a partial problem and determines a lower bound for the utility costs. The S-graph representation is extended with time intervals in order to manage the possible simultaneity of the hot and cold streams.

#### 3.2.1. Time interval management

A heat exchange can be established between the feed streams of two tasks if their temperatures are appropriate for heat transfer and there is enough time for simultaneous operation of the heat-exchange phases of the tasks. Supposing that a schedule is given by S-graph  $G(N, A_1, A_2)$ , then, for a task node  $i \in N$ , interval  $[t_i, T_i]$  is defined to include the set of possible starting times of task  $i$ . In other words, if  $t \in [t_i, T_i]$ , then, there exists a feasible starting time for all tasks, task  $i$  starts at  $t$ , and the deadline is satisfied.

Fig. 10 illustrates the relation between the feasible starting times of two tasks connected by an arc. The dependencies of the lower and upper bounds are indicated.

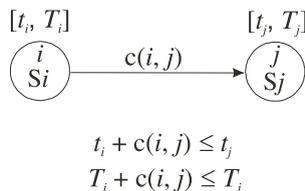


Fig. 10. Relations between time intervals of two adjacent nodes of an S-graph.

The time intervals for an S-graph can be determined by using the longest path algorithm. The algorithm can generate the earliest time for each task when they can start their activity relative to the starting time. Application of the longest path algorithm backward, it determines the latest starting time of a task that does not extend the predefined makespan.

Fig. 11 illustrates an S-graph with the earliest starting times of the tasks. They are determined by the longest path algorithm and are given an underlined number above the corresponding node of the S-graph.

Fig. 12 illustrates the application of the longest path algorithm backwards from the product nodes for the determination of the latest starting times of the tasks. For example, value  $-5$  of node 3 indicates that the task corresponding to node 3 has to start its activity not later than 5 units of time before the deadline.

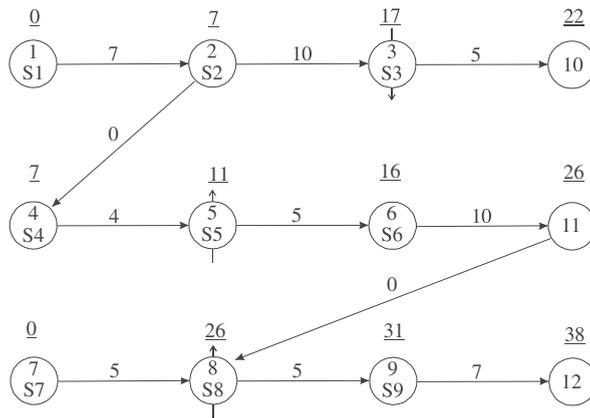


Fig. 11. Earliest starting times of the tasks of an S-graph (numbers underlined).

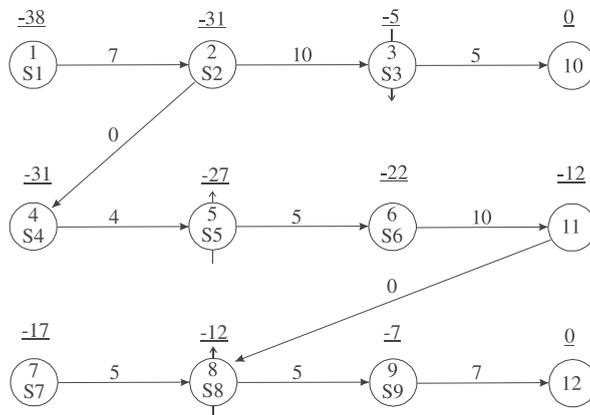


Fig. 12. Latest starting times of the tasks of an S-graph relative to the deadline (numbers underlined).

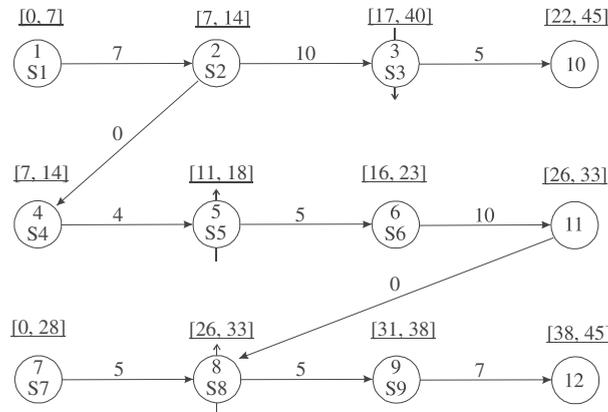


Fig. 13. Time intervals for an S-graph expressing the earliest and the latest start times of a task.

The latest starting time of a task relative to the overall starting time can be determined by adding the value of the deadline (45) to the values given in Fig. 12. Fig. 13 illustrates the time intervals of the S-graph in Figs. 11 and 12.

### 3.2.2. Branching step of the procedure

The proposed B&B procedure generates the schedule-graph that can provide minimal utility cost through solving a set of relaxed partial problems. These partial problems are organized in a so-called search tree. The recipe-graph represents the root partial problem of the search tree. At any partial problem, one equipment unit is selected, through the assignment to all unscheduled nodes according to the rules of task-equipment unit assignment all child partial problems are generated. If a new schedule-arc is added to the S-graph of the partial problem, the time intervals are updated. The pseudo code of the branching step of the procedure is given in Appendix A. For simplicity, it is assumed that there is exactly one equipment unit to perform a task. There is a high degree of freedom in realizing the search strategy for the branching procedure. For instance the order of selection the next equipment unit for scheduling can affect the efficacy of the algorithm.

### 3.2.3. Bounding step of the procedure

The bounding step of the procedure tests the feasibility of a partial problem and then determines a lower bound for the utility cost. The partial problem is feasible if the S-graph from the partial problem is acyclic and the makespan is lower than the deadline for the production. The procedure for bounding is given in Appendix A.

The feasibility test of a partial problem includes the update of time intervals and the loop search of the S-graph of the partial problem. If the S-graph contains a loop or the makespan is higher than the deadline for production the feasibility test fails.

For determining a lower bound for the utility cost, let  $H$  and  $C$  denote the sets of hot and the cold streams of the process determined by the recipe. For  $h \in H$ , let  $h_{in}$  and  $h_{out}$  denote the initial and final temperatures of hot stream  $h$  ( $h_{out} < h_{in}$ ), respectively. Similarly, for  $c \in C$ ,  $c_{in}$  and  $c_{out}$  are the initial and final temperatures of cold stream  $c$  ( $c_{in} < c_{out}$ ). For technical reasons, the temperatures of the cold streams are shifted upward by the minimal approach temperature.

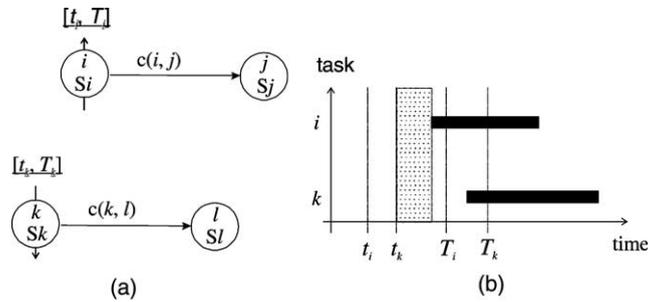


Fig. 14. Simultaneous availability of hot and cold streams: dotted area shows the feasible region for matching the hot and the cold stream.

Set  $HP$  contains all possible pairs of hot and cold streams that can be matched according to both thermodynamic and time constraints. Therefore, pair  $(h, c)$  is an element of set  $HP$  ( $h \in H$ ,  $c \in C$ ) if the initial temperature of hot stream  $h$  is greater than or equal to the final temperature of cold stream  $c$ , moreover, the final temperature of hot stream  $h$  is greater than or equal to the initial temperature of cold stream  $c$ . It is also supposed that the hot and cold streams can be available simultaneously according to the partial problem (i.e., the related time intervals are not disjoint).

Fig. 14(a) illustrates an S-graph with four tasks. The hot stream of task  $i$  can be matched with the cold stream of tasks  $k$  if time intervals  $[t_i, T_i]$  and  $[t_k, T_k]$  are not disjoint. This condition is necessary for the simultaneity of the two streams. Fig. 14(b) illustrates an instance of time intervals, in this case the dotted area indicates the time period where the two streams can be matched for heat exchange. The horizontal bars represent the production phase of the tasks. The heat exchange between the stream associated with task  $i$  and  $k$ , can only start in the earliest time when both task is ready to begin its activity, i.e., at time  $\max(t_i, t_k)$ .

**3.2.3.1. Variables of the mathematical model for the determination of a lower bound of the utility cost.** For all  $(h, c) \in HP$ , the value of binary variable  $y_{hc}$  is 1 if heat-exchanger unit between hot stream  $h$  and cold stream  $c$  is included in the heat exchanger network, otherwise, the value is 0. Nonnegative variable  $t_{hc}$  expresses the time used for heat exchange between hot stream  $h$  and cold stream  $c$ . Nonnegative variable  $Q_{hc}$  is defined to express the rate of heat flow from hot stream  $h$  to cold stream  $c$ . Variable  $x_{hc}$  expresses the potential starting time of the heat transfer from hot stream  $h$  to cold stream  $c$ , variable  $x_{ch}$  expresses the potential starting time of the heat transfer from cold stream  $c$  to hot stream  $h$ .

For all  $c \in C$  variable  $t_{c,util}$  represents the time used for heat transfer between cold stream  $c$  and the hot utility. Similarly, for all  $h \in H$ ,  $t_{h,util}$  is the time used for heat exchange between hot stream  $h$  and the cold utility. Nonnegative variables  $U_h$  and  $U_c$  express the rate of heat transferred from utilities for the hot stream  $h$  and for the cold stream  $c$ , respectively.

Let  $G(N, A_1, A_2)$  denote the S-graph of the partial problem. For all  $i \in N$  nonnegative variable  $x_i$  expresses the starting time of the task belonging to a task node, or the production time of the product belonging to a product node of the S-graph.

3.2.3.2. *Constraints of the mathematical model.* At most one heat exchanger unit can be fitted to the input stream of a task according to the problem definition, i.e.,

$$\sum_c y_{hc} \leq 1 \quad \text{for all } h \in H \tag{1}$$

$$\sum_h y_{hc} \leq 1 \quad \text{for all } c \in C \tag{2}$$

If a heat exchanger unit is excluded from the heat exchanger network, the time of heat exchange must be zero, i.e.,

$$t_{hc} \leq M y_{hc} \quad \text{for all } (h, c) \in HP \tag{3}$$

where  $M$  is a sufficiently big constant.

For all  $h \in H$ , constant  $QF_h$  denote the rate of release of heat of hot stream  $h$ , inducing the change in its rate of enthalpy flow. Similarly, for all  $c \in C$ , constant  $QF_c$  denote the rate of absorption of heat of cold stream  $c$ .  $QF_h$  and  $QF_c$  are determined by multiplying the specific heat, the flow rate of the material, and the change of the initial and final temperature of the stream. The rate of release or absorption of heat is equal to the sum of the heat transferred from another stream and the heat transferred from the utility, i.e.,

$$\sum_c Q_{hc} + U_h = QF_h \quad \text{for all } h \in H \tag{4}$$

$$\sum_h Q_{hc} + U_c = QF_c \quad \text{for all } c \in C \tag{5}$$

It is assumed that rate of heat flow is proportional to the time used for heat exchange, i.e.,

$$Q_{hc} = D_{hc} t_{hc} \quad \text{for all } (h, c) \in HP \tag{6}$$

where constant  $D_{hc} = U_{hc} \text{LMTD}_{hc} \text{AREA}$  and constants  $U_{hc}$ ,  $\text{LMTD}_{hc}$ , and  $\text{AREA}$  are the heat transfer coefficient, the logarithmic mean temperature difference of hot stream  $h$  and cold stream  $c$ , and the area of the heat-exchanger units, respectively.

Because of the constraint on the makespan none of the variables  $x_i$  can be greater than the upper bound on the makespan ( $MS$ ), i.e.,

$$x_i \leq MS \quad \text{for all } i \in N \tag{7}$$

If  $i, j \in N$  and they are connected by an arc and there is no heating or cooling requirement on the feed streams of task  $i$  (see Fig. 15), the task belonging to node  $j$  cannot be started earlier than the sum of the starting time of task  $i$  and the weight of the arc  $(i, j)$ , i.e.,

$$x_i + c(i, j) \leq x_j \quad \text{for all } (i, j) \in A_1 \cup A_2 \tag{8}$$

Fig. 16 shows an S-graph with a possible heat exchange between the pair  $(h, c) \in HP$ .

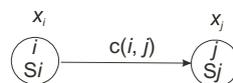


Fig. 15. Two nodes connected by arc.

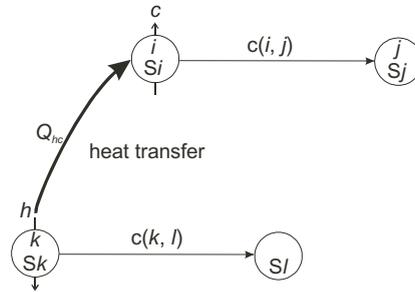


Fig. 16. S-graph with potential heat exchange.

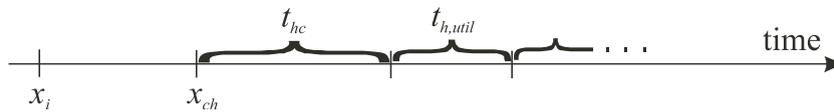


Fig. 17. Sequence of activities of a task with hot stream.

Fig. 17 illustrates the sequence of the activities that can occur at task  $i$ . The material is transferred to task  $i$  at time  $x_i$ , the heat exchange can begin in time  $x_{ch}$  and it takes  $t_{hc}$  units of time. Before the heat-exchange phase of the task  $t_{h,util}$  units of time can be used to supply the heat requirements from utilities if it is necessary.

On the basis of Figs. 16 and 17, constraints (9)–(12) express the required time intervals for cold stream  $c$ ,

$$x_i \leq x_{ch} \tag{9}$$

$$x_{ch} + t_{hc} + t_{c,util} + c(i, j) \leq x_j \tag{10}$$

Similarly for hot stream  $h$ ,

$$x_k \leq x_{hc} \tag{11}$$

$$x_{hc} + t_{hc} + t_{h,util} + c(k, l) \leq x_l \tag{12}$$

The utility consumption of a task depends on the time used for the utility transfer and the rate of heat flow between the heat stream and the source of utility, i.e.,

$$U_c = t_{c,util} D_c \quad \text{for all } c \in C \tag{13}$$

$$U_h = t_{h,util} D_h \quad \text{for all } h \in H \tag{14}$$

where  $D_c$  and  $D_h$  depend on the temperature difference between the stream and the utility.

If a heat exchanger unit with hot stream  $h$  and cold stream  $c$  is included in the heat exchanger network, constraint (15) expresses the requirement for the parallel availability of the stream  $h$  and  $c$ , i.e.,

$$-M(1 - y_{hc}) \leq x_{hc} - x_{ch} \leq M(1 - y_{hc}) \quad \text{for all } (h, c) \in HP \tag{15}$$

3.2.3.3. *Objective function of the model for the lower bound calculation of the utility usage.* The objective is to minimize the utility usage in the system; the utility usage for the hot and the cold streams is given by formula (16):

$$\min \left( K_c \sum_c U_c + K_h \sum_h U_h \right) \tag{16}$$

where constants  $K_c$  and  $K_h$  are the parameters of the cost of the cold and hot utilities, respectively.

### 3.3. Examples

Two examples, Examples 2 and 3, illustrate the integration of scheduling and HENS. Demonstration program together with input files of these examples are freely available at <http://www.dcs.vein.hu/demo/sch-hens>.

**Example 2.** The recipe graph is given in Fig. 18. There are three hot stream,  $h1$ ,  $h2$ ,  $h3$ , and one cold stream,  $c1$ , in the process. Sets  $S1 = \{E1\}$ ,  $S2 = \{E2, E3\}$ ,  $S3 = \{E1\}$ ,  $S4 = \{E1\}$ ,  $S5 = \{E2, E3\}$ , and  $S6 = \{E4\}$  give the plausible tasks to equipment units assignments.

The parameters for the heat streams are given in Table 1. The initial and the final temperature of the cold stream are shifted upward by the approach temperature that is regarded to be 10 K. The rate of heat flow between a utility and a stream is 150 MJ/h.

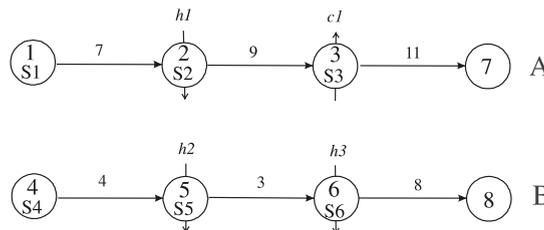


Fig. 18. Recipe-graph of Example 2.

Table 1  
Heat streams of the illustrative example

Name	Type	Initial temperature (K)	Final temperature (K)	Heat (MJ)
$c1$	Cold	283	323	200
$h1$	Hot	313	283	400
$h2$	Hot	313	303	100
$h3$	Hot	333	313	300

Based on set *HP*, only hot stream *h3* and cold stream *c1* can be matched and it is assumed that, for these specific conditions, the rate of heat transfer is of 20 MJ/h between hot stream *h3* and cold stream *c1*.

If no heat exchange is established between hot stream *h3* and cold stream *c1*, the total utility usage is 1000 MJ (200 + 400 + 100 + 300). If a heat exchanger unit is established between *h3* and *c1*, 200 MJ heat can be transferred between them, i.e., the total utility usage is reduced to 600 MJ. The transfer of 200 MJ takes 10 h. Solving the integrated problem will answer that the scheduling problem is flexible enough to provide 10 h for heat exchange between hot stream *h3* and cold stream *c1*.

The minimal makespan for Example 2 (using the utilities for heating or cooling is 31 h. In the integrated example the upper bound for the makespan is set to 36 h.

In the first branching step, an equipment unit *E1* is chosen and assigned to task 1. In the present partial problem one matching is possible. The model can be derived based on the previous section,

$$\min(U_{h1} + U_{h2} + U_{h3} + U_{c1})$$

s.t. according to constraints (1) and (2),

$$y_{h3c1} \leq 1$$

according to constraint (3),

$$t_{h3c1} \leq M y_{h3c1}$$

according to constraints (4) and (5),

$$Q_{h3c1} + U_{c1} = QF_{c1} = 200 \text{ (MJ)}$$

$$U_{h1} = QF_{h1} = 400 \text{ (MJ)}$$

$$U_{h2} = QF_{h2} = 100 \text{ (MJ)}$$

$$Q_{h3c1} + U_{h3} = QF_{h3} = 300 \text{ (MJ)}$$

according to constraint (6),

$$Q_{h3c1} \leq 20 t_{h3c1}$$

according to constraint (7),

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \leq 40$$

according to constraints (8)–(12),

$$\begin{aligned}
 x_1 + 7 &\leq x_2 \\
 x_2 + t_{h1,util} + 9 &\leq x_3 \\
 x_3 &\leq x_{c1h3} \\
 x_{c1h3} + t_{h3c1} + t_{c1,util} + 11 &\leq x_7 \\
 x_4 + 4 &\leq x_5 \\
 x_5 + t_{h2,util} + 3 &\leq x_6 \\
 x_6 &\leq x_{h3c1} \\
 x_{h3c1} + t_{h3c1} + t_{h3,util} &\leq x_8
 \end{aligned}$$

according to constraints (13) and (14),

$$\begin{aligned}
 U_{h1} &= 150t_{h1,util} \\
 U_{h2} &= 150t_{h2,util} \\
 U_{h3} &= 150t_{h3,util} \\
 U_{c1} &= 150t_{h4,util}
 \end{aligned}$$

according to constraint (15),

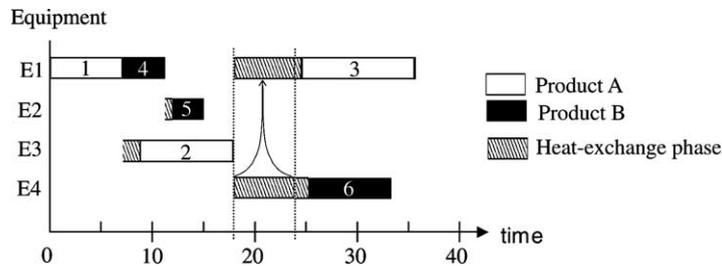


Fig. 19. Gantt chart of the optimal solution of the illustrative example.

Table 2  
Recipe for products A, B, and C

Task	Product A		Product B		Product C	
	Eq	Time (h)	Eq	Time (h)	Eq	Time (h)
1	E1	5	E1	5	E1	6
	E2	5	E2	5	E2	6
2	E3	4	E5	4	E5	3
	E4	4	E6	4	E6	3
3	E5	4	E7	4	E7	4
	E6	4	E8	4	E8	4
4	–	–	E1	5	–	–
	–	–	E2	5	–	–

$$-M(1 - y_{h3c1}) \leq t_{h3c1} - t_{c1h3} \leq M(1 - y_{h3c1})$$

The optimal solution is 769 MJ. Fig. 19 illustrates the Gantt chart of the optimal solution. The heat exchanger unit is established, between hot stream *h3* and cold stream *c*; the heat exchange between them starts at 18.667 and takes 5.769 long times.

**Example 3.** The recipes of products A, B, and C are given in Table 2. The number of batches to be produced is given in Table 3 for each product. The recipe-graph of Example 3 is given in Fig. 20.

The parameters of the heat streams are given in Table 4. The available hot and cold utilities are listed in Table 5.

The heat transfer coefficient between a hot and a cold streams is 1500 W/m<sup>2</sup> K. The area of a heat exchanger unit is 3 m<sup>2</sup>.

Table 3  
Number of batches of the products

Product	A	B	C
Number of batches	4	3	2

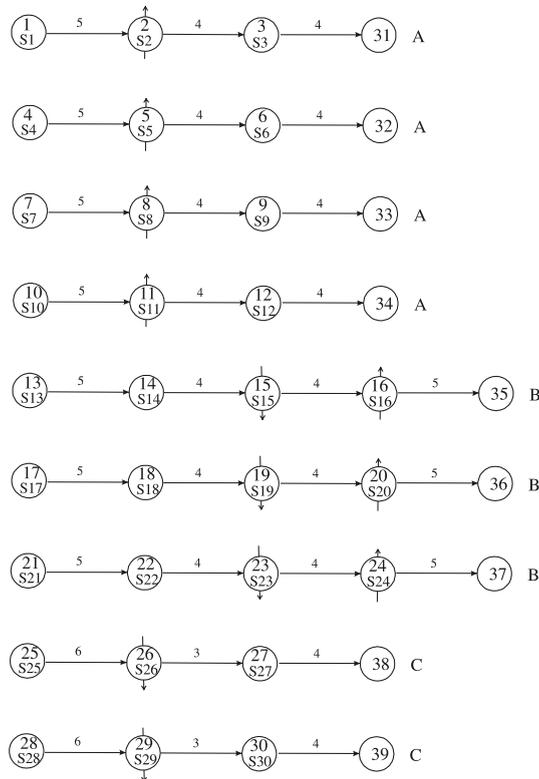


Fig. 20. Recipe-graph of Example 3.

If the heating and cooling duties are satisfied by utilities, the minimal makespan is 33.1 h with 3100 MJ utility. Extending the upper bound for the makespan to 36 h, the required utility is

Table 4  
Heat streams of Example 3

Name	Type	Initial temperature (K)	Final temperature (K)	Heat (MJ)	Product, task
c1	Cold	313	393	400	A, 2
h1	Hot	413	323	200	B, 3
c2	Cold	353	403	100	B, 4
h2	Hot	423	313	300	C, 2

Table 5  
Hot and cold utilities of Example 3

Type	Temperature (K)
Hot	473
Cold	283

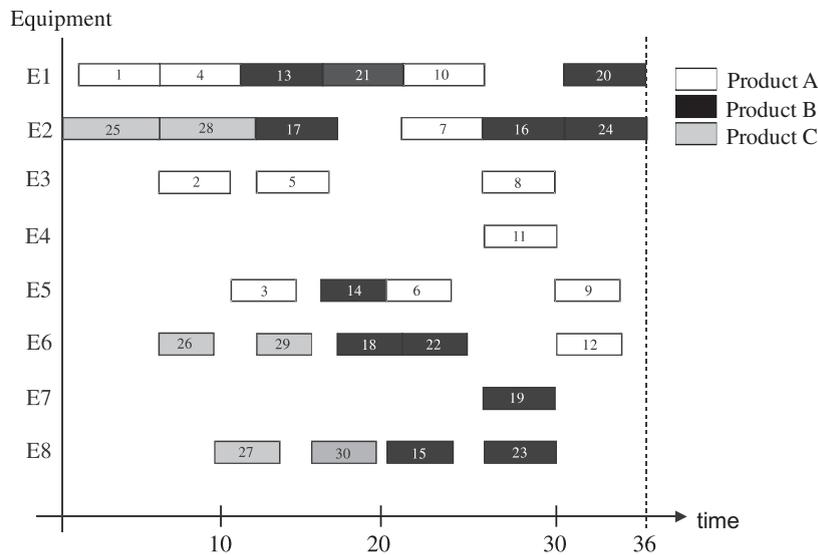


Fig. 21. Gantt chart of the optimal solution of Example 3.

Table 6  
Heat exchanges between the feed streams of the tasks in the optimal solution of Example 3

#	Heat transfer (task numbers)	Starting time (h)	Length of heat exchange (h)
1	19 → 8	25.933	0.428
2	23 → 11	26	0.428
3	26 → 2	6	0.375
4	29 → 5	12	0.375

reduced to 1100 MJ. Fig. 21 shows the Gantt chart of the optimal solution, Table 6 lists the heat exchanges in this solution.

#### 4. Concluding remarks

In this paper a methodology based on the S-graph approach for incorporating heat integration into batch production scheduling is presented. Here is shown how the S-graph is capable to efficiently solve heat integration and scheduling problems in an integrated manner. Results of this paper show how utility usage can be reduced considerably with just slight increase of production makespan.

#### Acknowledgements

This research has been supported in part by the Hungarian National Science Foundation (OTKA T029-309). Financial support also received from the “Generalitat de Catalunya”, FI programs and project GICASA-D are fully appreciated. Support received in part by the European community (project no. G1RD-CT-2001-00466) is acknowledged. Funds were also received from MCyT (project no. DPI2002-00856).

#### Appendix A

The main procedure, the branching, and the bounding procedure is given in Figs. 22–24.

```

procedure main
notation:   $n$  : number of equipment units
             $N_i$  ( $i = 1, 2, \dots, n$ ) : set of tasks that can be performed by equipment unit  $i$ 
             $last\_node$  : set of pairs  $(i, j)$ , where  $i$  is an equipment unit, and  $j$  is a task (node)
             $time$  : set of triples  $(j, LT, UT)$  where  $j$  is a task (node),  $LT$  and  $UT$  are the lower
                    and upper bounds of starting time of task  $j$ 
             $PP = (G(N, A_1, A_2), bound, last\_node, SOUN, time)$ 
input: recipe-graph  $G(N, A_1, \emptyset)$ ,  $N_i$  ( $i = 1, 2, \dots, n$ ) and parameters for hot and cold streams
begin
     $SET = \emptyset$ ;  $bound = 0$ ;  $SOUN = N_1 \cup N_2 \cup \dots \cup N_n$ ;  $last\_node = \emptyset$ ;  $current\_best = \infty$ ;
    initialize set  $time$  based on  $G(N, A_1, \emptyset)$ ;
    put  $(G(N, A_1, \emptyset), bound, last\_node, SOUN, time)$  into  $SET$ ;
    while  $SET \neq \emptyset$  do
        begin
        select and remove one element from  $SET$ , it is denoted by  $PP$ ;
        branching( $PP$ );
        end;
        if  $current\_best < \infty$  then print solution;
    end

```

Fig. 22. Main procedure of the integrated algorithm.

```

procedure branching(PP)
comment: generates all child partial problem of partial problem PP
notation: graph(PP) : S-graph  $G(N, A_1, A_2)$  from PP
           bound(PP) : bound from PP
           last_node(PP) : set last_node from PP
           SOUN(PP) : set SOUN from PP
           time(PP) : set time from PP
begin
  let EQ be an equipment unit that can be assigned to an unscheduled task (node);
  let  $SO = N_{EQ} \cap SOUN(PP)$ ;
  for all  $k \in SO$  do
    begin
      if there is no pair  $(i, j) \in last\_node(PP)$  such that  $i = EQ$  then
        begin
          put(graph(PP), bound(PP), last_node(PP)  $\cup \{(EQ, k)\}$ , SOUN(PP)\{k\}, time(PP))
            into SET;
          end;
        else
          begin
            let  $G_0(N, A_1, A_2) = graph(PP)$ ;
            let  $time_0 = time(PP)$ ;
             $new\_arc = \emptyset$ ;
            for all  $(j, l) \in A_1$  do
              begin
                 $G_0(N, A_1, A_2) = G_0(N, A_1, A_2 \cup \{(l, k)\})$ ;
                let  $new\_arc = new\_arc \cup \{(l, k)\}$ ;
              end;
            bounding( $G_0(N, A_1, A_2)$ , bound,  $time_0$ ,  $new\_arc$ );
            if bound < current_best then
              begin
                if SOUN(PP)\{k\} =  $\emptyset$  then
                  update current_best, SET, and solution;
                else
                  put ( $G_0(N, A_1, A_2)$ , bound, lastnode(PP) $\cup \{(EQ, k)\} \setminus \{(EQ, j)\}$ ,
                    SOUN(PP)\{k\},  $time_0$ ) into SET;
                end;
              end;
            end;
          end;
        end;
      return;
    end;
  end;

```

Fig. 23. Branching procedure of the integrated algorithm.

```

procedure bounding( $G(N, A_1, \emptyset)$ , bound,  $time_0$ ,  $new\_arc$ )
begin
  for all  $(i, j) \in new\_arc$  do update( $time_0$ ,  $(i, j)$ );
  if feasible then
    bound = solve_hens( $G(N, A_1, \emptyset)$ ,  $time_0$ );
  else
    bound =  $\infty$ ;
  end;

```

Fig. 24. Bounding procedure of the integrated method.

**References**

- [1] E. Sanmartí, F. Friedler, L. Puigjaner, Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation, *Comput. Chem. Engng.* 22 (1998) S847–S850.
- [2] J. Corominas, A. Espuna, L. Puigjaner, Method to incorporate energy integration considerations in multiproduct batch processes, *Comput. Chem. Engng.* 18 (1994) 1043–1055.
- [3] E. Font, A. Espuna, L. Puigjaner, Advanced concepts for energy integration in time-dependent processes, in: *ECOS'96*, Stockholm, Sweden, 1996, pp. 353–359.
- [4] G.V. Reklaitis, Perspectives on scheduling and planning of process operations, in: *Proc. PSE'91 Conf.*, Montebello, Canada, 1991.
- [5] N. Shah, Single and multisite planning and scheduling: current status and future challenges, foundations of computer-aided process operations, *AIChE Symp. Ser.* 94 (1998) 91–110.
- [6] J.M. Pinto, I.E. Grossman, Assignment and sequencing models for the scheduling of process systems, *Ann. Oper. Res.* 81 (1998) 433.
- [7] E. Sanmarti, L. Puigjaner, T. Holzinger, F. Friedler, Combinatorial framework for effective scheduling of multipurpose batch plants, *AIChE J.* 48 (2002) 2557–2570.
- [8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, MIT Press, 1997.
- [9] B. Linnhoff, J. Flower, Synthesis of heat exchanger network, *AIChE J.* 24 (1978) 633–642.
- [10] R. Smith, *Chemical Process Design*, McGraw-Hill, New York, 1995.
- [11] J.M. Douglas, *Conceptual Design of Chemical Processes*, McGraw-Hill, New York, 1988.
- [12] L.T. Biegler, I.E. Grossman, A.W. Westerberg, *Systematic Methods of Chemical Process Design*, Prentice Hall PTR, New Jersey, 1997, pp. 567–617.
- [13] A.B. Nagy, R. Adonyi, L. Halasz, F. Friedler, L.T. Fan, Integrated synthesis of process and heat exchanger networks: algorithmic approach, *Appl. Therm. Engng.* 21 (2001) 1407–1427.