Available online at www.sciencedirect.com

SCIENCE DIRECT

Computers & Industrial Engineering 48 (2005) 311–325

computers & industrial engineering

www.elsevier.com/locate/dsw

Evolutionary algorithm for advanced process planning and scheduling in a multi-plant[☆]

Chiung Moon[a], Yoonho Seo[b,*]

[a]Department of Information and Industrial Engineering, Hanyang University, Ansan 425-791, South Korea
[b]Department of Industrial and Information Engineering, Korea University, Seoul, South Korea

Abstract

Integration of process planning and scheduling is one of the most important functions to support flexible planning in a multi-plant. The planning and scheduling are actually interrelated and should be solved simultaneously. In this paper, we propose an advanced process planning and scheduling model for the multi-plant. The objective of the model is to decide the schedules for minimizing makespan and operation sequences with machine selections considering precedence constraints, flexible sequences, and alternative machines. The problem is formulated as a mathematical model, and an evolutionary algorithm is developed to solve the model. Numerous experiments are carried out to demonstrate the efficiency of the proposed approach.

Keywords: Process planning and scheduling; Multi-plant chain; TSP with precedence constraint; Evolutionary algorithm

1. Introduction

Many manufacturers now try to optimize the total system to cope with a global manufacturing. This trend brings the idea of supply chain, which is to optimize not only the plant operations but also the whole activities from a supplier to a customer. As a result, manufacturing companies nowadays are migrating from separated planning processes toward the more coordinated and integrated planning processes to provide high quality products at lower cost.

Integration of process planning and scheduling is one of the most important problems for supporting the total optimization. The two functions are interrelated because both of them take part in the assignment of factory machines to production tasks. Hence, the actual process planning and scheduling

☆ This manuscript was processed by Area Editor Mitsuo Gen.
* Corresponding author. Tel.: +82-2-3290-3393; fax: +82-2-929-5888.
E-mail addresses: cumoon@ihanyang.ac.kr (C. Moon), yoonhoseo@korea.ac.kr (Y. Seo).

0360-8352/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2005.01.016

problem should be solved concurrently, but the problem has more complexities due to the alternative machines and alternative operations sequences. We define the problem as an advanced process planning and scheduling problem (APPS). Therefore, the APPS should focus upon the following issues: (1) How to make a flexible process plan considering shop floor status and design information? (2) How to make an efficient schedule considering the job shop's dynamic situations and the complexity of the machine constraints? (3) How to make an appropriate integrated model which includes various constraints?

In the traditional approaches, process planning and scheduling are done sequentially, where the process plan is determined before the actual scheduling is performed. But, this simple approach ignores the relationship between scheduling and process planning.

Recently, some research results for the integrated process planning and scheduling are presented. Tan (2000) presents a review of the research in the process planning and scheduling area and discusses the extent of applicability of various approaches. Hankins, Wysk, and Fox (1984) discuss the advantages of using alternative operations sequences to improve the productivity of the shop floor. They show that the efficient planning considering the alternative machines results in reduced lead-time and in improved overall machine utilization. Nasr and Elsayed (1990) present two heuristics to determine an efficient schedule for the *n* jobs, *m* machines problem with alternative machine routings for each operation. The objective they adopted is to minimize the mean flow time. Palmer (1996) and Sundaram and Fu (1988) solve the IPPS problem using a simulated annealing. Brandimarte and Calderini (1995) develop a two-phase hierarchical tabu search. Saygin and Kilic (1999) propose a frame for IPPS with the objective of reducing the completion time. To solve the problem, a heuristic method is proposed. Morad and Zalzala (1999) develop an evolutionary algorithm (EA)-based method to tackle the IPPS problem.

However, the major weakness of the models so far introduced lies in that they consider the alternative machines for each operation with a fixed sequence or the non-constraint operational sequence when constructing a schedule. In this paper, we consider an APPS problem for a multi-plant composed of a network of production facilities, and of multiple products flow through manufacturers. The multi-plant means extending the integration concept beyond on production site by means of stronger distribution management capabilities, electronic data interchange, and coordinated multiple plant management. We develop a model incorporating the alternative machines in the chain. The alternative machines have different capabilities and require unequal processing time for an operation. The operations sequences for each job include precedence constraints. In order to obtain good approximate solutions, we develop an EA-based heuristic approach.

2. Problem definition

In a customized manufacturing environment, most jobs have a different sequence of operation steps and may have a set of alternative process plans. Fig. 1 shows an example of the general structure of the network for a customized manufacturing. This kind of structure is common in heavy industries, e.g. turbine manufacturing, generator manufacturing, and ship engine manufacturing. The production cycle in this chain consists of three main units: with the supplier, raw materials are transformed into specified workpieces; with the manufacturers, machining processes—drilling, milling, boring and grinding—are executed to remove pieces of the workpiece; and with the assembler, final products are produced. Among the units of supply chain in Fig. 1, the manufacturer plays the main role because it is a bottleneck unit. Therefore, the manufacturer unit is usually represented as a network of plants.

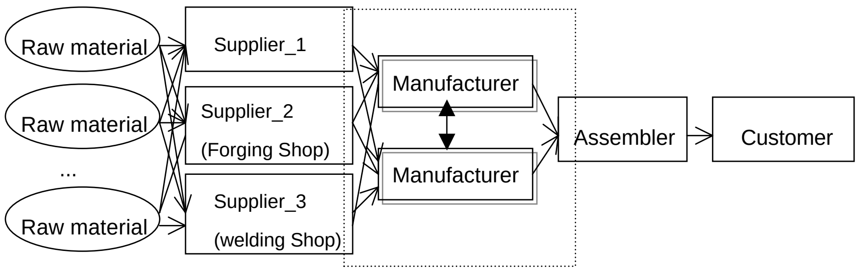

Fig. 1. Structure of a supply chain network for manufacturing.

Each plant includes several machines, which have different functions, processing time, and capabilities. Hence, an actual optimization should be done considering the dynamic status of multiple plants chain.

In multi-plant, a process plan should be able to represent all possible precedence that occurs among the planning and processing decisions. An AND/OR directed graph is used to represent the sequential and parallel structure of machining processes (Homem de Mello & Sanderson, 1990). An example to illustrate the technologically feasible sequence of processes is shown in Fig. 2.

From an unordered set of machining processes with precedence relations, operations sequencing is to determine a sequence by exploiting a sequence space derived from the combination of parallel processes. Since the transition time is sequence dependent, the operations sequencing problem (OSP) can be formulated as a well-known traveling salesman problem (TSP) (Finke, Claus, & Gunn 1984; Kusiak & Finke, 1987). The transition time between operations can be calculated only with a machine tool assigned for each operation.

On the other hand, machine selection for each operation is not trivial since alternative machines for each operation should be considered as a candidate, and each of them has influence on the various performance measures including total machining and transportation times. The OSP can be formulated as a TSP, which solves a subproblem—the machine selection problem—to calculate the cost of the trips. Moreover, since each TSP determines the operation sequence for each part type, multiple TSPs corresponding to the number of parts in part mix should be considered simultaneously.

The objective of APPS in multi-plant chain is to determine an optimal schedule with operation sequences for the jobs. Therefore, the APPS problem we are considering is defined as: given a set of *n* jobs which are to be processed on *m* machines with alternative operations sequences and alternative machines for operations in the environment of multi-plant chain, find an operations sequence for each job and a schedule in which jobs pass between machines and a schedule in which operations on the same jobs are processed such that it satisfies the precedence constraints and it is optimal with respect to

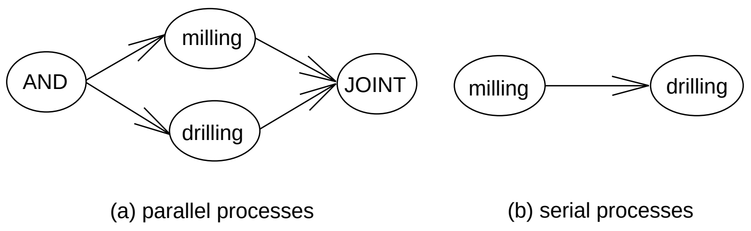

Fig. 2. AND/OR graph representation.

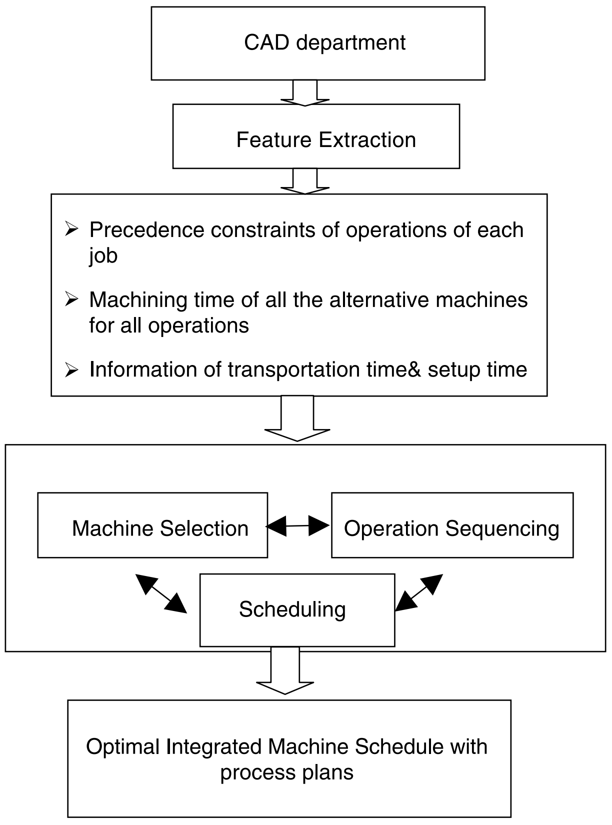
CAD department
Feature Extraction
➢ Precedence constraints of operations of each job
➢ Machining time of all the alternative machines for all operations
➢ Information of transportation time& setup time
Machine Selection
Operation Sequencing
Scheduling
Optimal Integrated Machine Schedule with process plans

Fig. 3. Schematic diagram of IPPS.

makespan criterion. Here, the manufacturing system is composed by a network of plants. The schematic diagram of APPS is shown in Fig. 3.

3. Model development

The APPS problem includes operations sequencing which selects a machine for each operation and decides schedules for all parts. The OSP can be treated as multiple TSPs, each of which determines the machine operations sequence for each part type. Since the sequence should obey the precedence relations intrinsic to the part, the TSP can be considered as a precedence constrained one. Moreover, the transition costs between operations for a part type, which corresponds to the travel costs between cities in original TSP, are not given but should be obtained by solving a machine selection subproblem. Rather than dealing with the whole problem in a single stage, we consider an intermediate problem with assumption that a single machine is fixed to each of machining processes, that is, OSP with a fixed machine for each operation. Then, by relaxing the assumption, we can easily reach to the model formulation of the integrated machine selection and operations sequencing problem.

To formulate a TSP with precedence constraints, the two-commodity network flow model can be used (Finke et al., 1984; Kusiak & Finke, 1987). Suppose that there are two distinct commodities p and q given in the network with J nodes or cities. While commodity p is supplied by $(J-1)$ units at a selected starting node and used by one unit at each node that is not the starting node, q is a commodity to consume $(J-1)$ units at the starting node and to be supplied by one unit at the other nodes. Such network flows of the commodities are characterized by two properties: First, the sum of commodities p and q in any feasible tour should be equal to $J-1$, and second, the quantity of commodity p (or q) outbounded from a node is decreasing as the tour proceeds. These properties are used to model the precedence relations for a constraint TSP. The following notations are used to describe the problem throughout the whole paper:

K	product mix, set of K different part types, i.e. $K=\{1,2,\ldots,k,\ldots,l,\ldots,K\}$.
M	set of machines, $M=\{1,2,\ldots,m,\ldots,M\}$.
G_k	set of operations for part type k, i.e. $G_k=\{g_{ki} \mid \forall\, i=1,2,\ldots,J_k\}$, where g_{ki} is operation name of the ith element and J_k is the number of operations.
p_{kim}	processing time of operation i of part type k on machine m.
τ_{mn}	transportation time from machine m to n.
st_{kij}	set-up time from operation g_{ki} to g_{kj} on part type k.
s_k	first selected operation for part type k.
A_m	set of operations to be processed on machine m.
C_{kim}	completion time of operation i for part type k on machine m.

3.1. OSP with fixed machine for each operation

OSP is defined as: given part mix $\boldsymbol{K}$ and sets of unordered machining operations $\boldsymbol{G_k}$ with precedence constraints for all part types $k\in\boldsymbol{K}$, determine the operation sequences for all part types such that the total transition cost for all part types is minimized. Since a single machine is fixed for each operation in this problem, a notation $\mu(k, i)$ is used to denote the machine assigned to ith operation of part type k. Let c_{kij} be the transition time from operation g_{ki} to g_{kj} of type k. The transition time c_{kij} is defined as the sum of total processing time on a assigned machine $\mu(k, i)$ for operation g_{ki} for q_k parts and total transportation time from the machine $\mu(k, i)$ to machine $\mu(k, j)$ for part k. The two machine, $\mu(k, i)$ and $\mu(k, j)$, may be included in the same plant. If the two machines are not included in the same plant, a longer transportation time is needed than they are in the same plant. That is, there are two types of transportation times from the machine $\mu(k, i)$ to machine $\mu(k, j)$ for part k. Then the transition time h_{kij} can be expressed as follows:

$$h_{kij} = p_{ki\mu(k,i)} + \tau_{\mu(k,i)\mu(k,j)} + st_{kij} \tag{1}$$

Three variables are introduced as follows:

$$y_{kij} = \begin{cases} 1 & \text{if operation } i \text{ is performed immediately after operation } j \text{ for part type } k, \text{ and} \\ 0 & \text{otherwise,} \end{cases}$$

y^{p}_{kij} quantity of commodity p from operation g_{ki} to g_{kj} for part type k.
y^{q}_{kij} quantity of commodity q from operation g_{ki} to g_{kj} for part type k.

The two-commodity network flow model is modified to accommodate the multiple TSPs. The complete model for OSP with a pre-assigned machine for each operation, which can be treated as multiple constraint-case TSPs, can be described as follows:

$$\text{Minimize } \sum_{k=1}^{K}\sum_{i=1}^{J_k}\sum_{\substack{j=1\\ i\neq j}}^{J_k}\frac{1}{(J_k-1)}h_{kij}(y^p_{kij}+y^q_{kij}), \tag{2}$$

$$\text{Subject to } \sum_{j=1}^{J_k} y^p_{kij} - \sum_{j=1}^{J_k} y^p_{kji} = \begin{cases} J_k-1, & \text{for } i=s_k, \\ -1, & \text{elsewhere,} \end{cases} \quad \forall\, k, \tag{3}$$

$$\sum_{j=1}^{J_k} y^q_{kij} - \sum_{j=1}^{J_k} y^q_{kji} = \begin{cases} -(J_k-1), & \text{for } i=s_k, \\ +1, & \text{elsewhere,} \end{cases} \quad \forall\, k, \tag{4}$$

$$\sum_{j=1}^{J_i}(y^p_{kij}+y^q_{kij}) = J_k-1 \quad \forall\, k \text{ and } i, \tag{5}$$

$$y^p_{kij}+y^q_{kij} = (J_k-1)y_{kij} \quad \forall\, k,\ i \text{ and } j, \tag{6}$$

$$\sum_{j=1}^{J_k} y^p_{kuj} - \sum_{j=1}^{J_k} y^p_{kvj} \geq 1, \quad \forall\, k \text{ and } (g_{ku}\rightarrow g_{kv})(g_{kv}\neq s_k), \tag{7}$$

$$y^p_{kij}\geq 0, \quad \forall\, k,\ i \text{ and } j, \tag{8}$$

$$y^q_{kij}\geq 0, \quad \forall\, k,\ i \text{ and } j, \tag{9}$$

$$y_{kij}\in\{0,1\}, \quad \forall\, k,\ i \text{ and } j. \tag{10}$$

The objective function (2) expresses total transition times for all part types, since the sum of commodities p and q between operation g_{ki} to g_{kj} on any feasible operations sequence (i.e. $y_{kij}=1$) is equal to J_k-1 (i.e. $y^p_{kij}+y^q_{kij}=J_k-1$). The constraints (3) and (8) are used to ensure the feasibility of network flow of commodity p. Similarly, constraints (4) and (9) are expressed for commodity q. Constraint (5) ensures the feasible tour, i.e. feasible operations sequence. Constraint (6) explains that if $y_{kij}=1$ the sum of commodities p and q between g_{ki} and g_{kj} be J_k-1. Constraint (7) explains any precedence relationship between machining operations. Since individual TSPs are independent to each other, the formulation is a simple collection of TSPs with precedence constraints.

3.2. Machine selection and operations sequencing problem

By deleting a machine fixation assumption, the OSP can be extended to the integrated operations sequencing and machine selection problem we are considering. The extended problem deals with not only operations sequencing for all part types, but machine selection for each operation. One more decision variable is introduced:

$$x_{kim} = \begin{cases} 1 & \text{if machine } m \text{ is selected for operation } i \text{ of part type } k, \\ 0 & \text{otherwise.} \end{cases}$$

With the introduction of the variable, the transition time from operation i to j for part k, the c_{kij} can be redefined as follows:

$$h_{kij} = \sum_{m=1}^{m}\sum_{n=1}^{M}\{p_{kim}x_{kim} + \tau_{mn}x_{kim}x_{kjn} + \mathrm{st}_{kij}y_{kij}\} \tag{11}$$

Eq. (11) is treated as the objective function for the integrated machine selection and OSP. In the equation, the set-up and the transportation time are dependent upon the sequence and are not included in the processing time. Then, the complete zero-one integer-programming model for the integrated machine selection and OSP can be summarized:

$$\text{Minimize } \sum_{k=1}^{K}\sum_{i=1}^{J_k}\sum_{\substack{j=1 \\ i\neq j}}^{J_k}\sum_{m=1}^{M}\sum_{n=1}^{M}\frac{1}{J_k-1}h_{kij}(y^p_{kij}+y^q_{kij}), \tag{12}$$

$$\text{Subject to } \sum_{m=1}^{M} x_{kim} = 1 \quad \forall\, k \text{ and } i, \tag{13}$$

$$x_{kim} \in \{0,1\} \quad \forall\, k,\ i \text{ and } m, \tag{14}$$

constraints (3), (4), (5), (6), (7), (8), (9), and (10).

Constraint (13) ensures that only one machine for each operation should be selected. Constraint (14) ensures the integrity of variables x_{kim}'s.

3.3. APPS

Scheduling involves the time allocation for operations in the selected operations sequences with the objective of minimizing the makespan. For the objective, we allow that any two operations belonging to the same part can proceed at the same time if available machines exist. Therefore, the lot of a part can be divided for processing on the next operation as much as the load size of the transfer device. To present the IPPS model, the following notation is introduced:

$$z_{ijm} = \begin{cases} 1 & \text{if operation } i \text{ proceeds operation } j \text{ on machine } m, \\ 0 & \text{otherwise.} \end{cases} \qquad \forall\, [i,j] \in A_m, \quad m = 1,2,\dots,M$$

The objective function is to minimize the makespan. The overall model can be described as follows:

$$\text{Minimize } \max_{1\le m\le M}\left\{\max_{1\le i\le Jk} c_{kim}\right\} \tag{15}$$

$$\text{Subject to} \sum_{i=1}^{J_k} \sum_{\substack{j=1 \\ i \neq j}}^{J_k} \sum_{m=1}^{M} \sum_{n=1}^{M} \frac{1}{J_k - 1} h_{kij}(y_{kij}^{p} + y_{kij}^{q}) \leq MS \quad \forall\, k \tag{16}$$

$$C_{ljm} - C_{kim} + W(1 - z_{ijm}) \geq p_{ljm} \times q_l \quad \forall\, [i,j] \in Am, \ \forall\, m, \ i \neq j \tag{17}$$

$$C_{kim} - C_{ljm} + Wz_{ijm} \geq p_{ljm} \times q_k \tag{18}$$

$$c_{kim} \geq 0 \tag{19}$$

constraints (3), (4), (5), (6), (7), (8), (9), (10), (13), and (14).

W is an arbitrarily large positive number. Constraint (16) imposes that the completion time of each order is not greater than the makespan. Constraints (17) and (18) ensure that a resource cannot be processed by more than one operation at the same time. Constraint (19) imposes the non-negative condition.

4. Evolutionary algorithm

As the number of part types and the size of the operation set gets larger, the analytical model proposed becomes intractable. Therefore, development of an efficient heuristic algorithm, that is useful to a real problem, is required. The EA has been successfully applied to many combinatorial optimization problems since it was initially proposed by Holland (Gen & Cheng, 1997, 2000). The most attractive feature of EA is the flexibility of handling various kinds of objective functions with fewer requirements on fine mathematical properties. Therefore, the GA approach can be applied to solve the IPPS model. The main issues in developing an EA-based approach are chromosome representation, initialization of the population, evaluation measure, crossover, mutation, and selection strategy. Also, the genetic parameters such as population size *pop_size*, number of generation *max_gen*, probability of crossover p_c, and probability of mutation p_m, should be determined.

4.1. Solution representation and initial solution generation

In a directed graph, the vertices represent operations, while the edges represent the precedence relations between operations. A topological sort (TS) technique can be used to obtain all the feasible sequences in a directed graph (Horowitz & Sahni, 1984). In this paper, to derive a feasible complete sequence from a directed graph, an ordering technique using the TS and random assignment of priority to each node is proposed. The representation structure is shown in Fig. 4. With this representation scheme including a directed graph and the string of priorities, a feasible solution can be derived for each assignment of positional priorities to the nodes in a directed graph. By associating this ordering

Vertices	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Priority	5	1	7	9	4	6	3	8	2	10

Fig. 4. Representation structure.

```
procedure: a feasible solution generation
    input: set of directed graphs with precedence constraints for all part types and set of
        machine tools for each operation;
    generating a set of priorities on the nodes in the directed graph;
   while (any vertex remains) do
        if every vertex has a predecessor, then the network is infeasible: stop.
        else pick a vertex v with the highest priority among vertices with no predecessors;
            que ← v;
            select a machine with minimum processing time among available machines for
                an operation corresponds to the v;
            delete v and all edges leading out of v from the directed graph;
   end_while.
   while (any vertex remains) do
        allocate all operations from que to the corresponding machines sequentially;
   end_while.
    end_procedure.
```

Fig. 5. Feasible solution generation.

technique with the genetic algorithm, an effective heuristic algorithm for the APPS problem can be developed. Thereafter, since a string of priorities drives a feasible path for a directed graph, it can be considered as a chromosome used in the EA.

A procedure to select machines and generate a feasible sequence from a directed graph by using TS and random priority assignment is described in Fig. 5.

Examples of directed graphs to represent two process plans with precedence constraints are illustrated in Fig. 6. In this graph, a directed edge $<g_{ki}, g_{kj}>$ indicates that vertex g_{ki} must be completed before vertex g_{kj}. By assigning a random priority to each node in the graphs as in Fig. 4, a unique sequence which obeys the precedence relation can be generated. The vertex v_2 is selected as the first position since its priority is higher than that of v_1 and v_6. Then, v_2 is stored in queue and the edge $v_2 \rightarrow v_4$ is removed. In the resulting network, v_1, v_4, and v_6 have no predecessor. The vertex v_1 is selected as the next position since it has the highest priority among them. Continuing this procedure, a final feasible path (v_2, v_1, v_6, v_7, v_9, v_3, v_8, v_4, v_5, v_{10}) is uniquely obtained for the string of priority '5 1 7 9 4 6 3 8 2 10'.

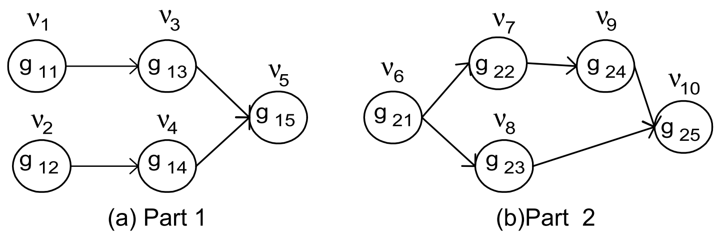

Fig. 6. Examples of part types with precedence constraints.

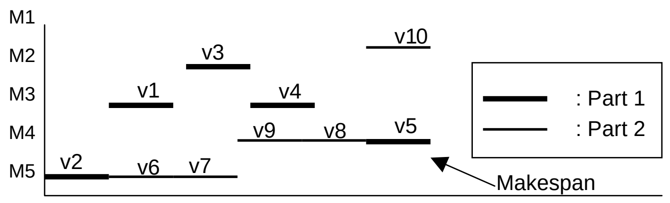

Fig. 7. A feasible schedule.

From this result, we see that the process sequences for two part types are g_{12}–g_{11}–g_{13}–g_{14}–g_{15} and g_{21}–g_{22}–g_{23}–g_{24}–g_{25}. Let the feasible sequence with the corresponding machines for all operations be $v_2(5)$–$v_1(3)$–$v_6(5)$–$v_7(5)$–$v_9(4)$–$v_3(2)$–$v_8(4)$–$v_4(3)$–$v_5(4)$–$v_{10}(1)$, we then obtain a feasible schedule as shown in Fig. 7.

4.2. Initialization

The initialization of the population of chromosomes can be made by randomly generating the chromosomes as much as the populations size *pop_size*. Each chromosome is represented as a string of integers. Each digit of the string means the priority of the gene and ranges between 1 and the number of genes.

4.3. Selection and fitness evaluation

Selection strategy is concerned with choosing chromosomes from population space It may create a new population for the next generation based on either parent and offspring or part of them. A mixed strategy based on the roulette wheel and elitist selection is adopted as the selection procedure (Gen & Cheng, 1997).

To improve the solutions, each chromosome is evaluated using some measures of fitness. A fitness value is computed for each chromosome in the population and the objective is to find a chromosome with the shortest tour. There are various cost functions to measure the effectiveness of a formulation. In this paper, we consider only one objective function, i.e. to minimize makespan.

4.4. Evolutionary operators

To create the next generation, a new set of chromosomes called offspring is formed by the executing of evolutionary operators such as selection, crossover, and mutation. In particular, the crossover operator acts as the main operator and exercises a great influence on performance of the EA approach. On the other hand, the mutation operator acts a background operator. In this paper, the order-based crossover is employed. An example of the order-based crossover is illustrated in Fig. 8. Suppose that two chromosomes are parent 1 = [5 1 2 4 6 3] and parent 2 = [2 6 1 4 3 5], assuming each 2 columns from the left are job1, job2, and job3, respectively.

With the same procedure, we can produce a modified parent 1 as [6 1 2 4 3 5].

The swap mutation operator is introduced here. The swap scheme is select two genes within a chromosome at random and then swap these contents as shown in Fig. 9.

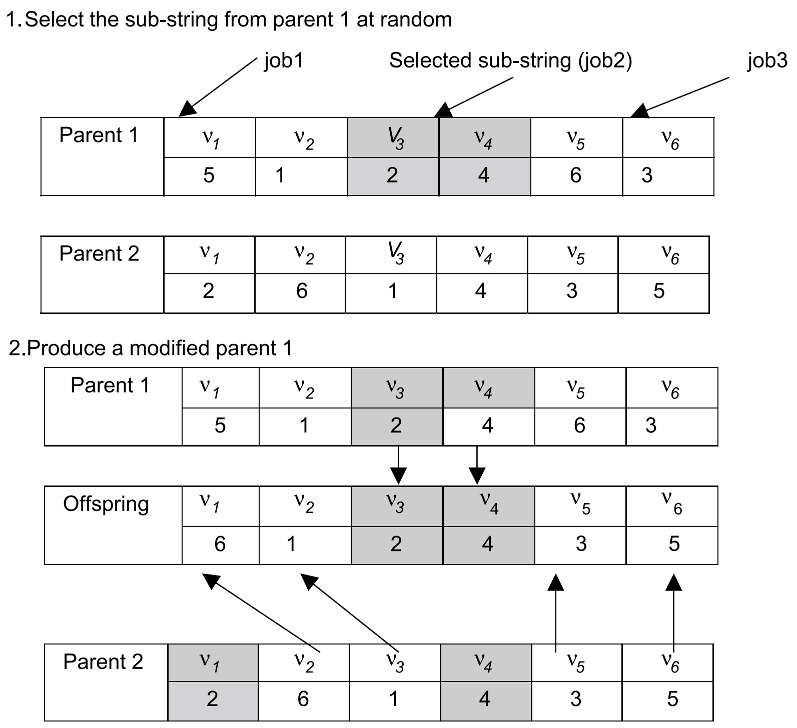

Fig. 8. Illustration of the order-based crossover.

4.5. Overall procedure

Let $P(t)$ and $C(t)$ be, respectively, populations for parent and offspring in generation t, Overall procedure of the proposed genetic algorithm is described as shown in Fig. 10. Since the algorithm is to solve APPS, multiple executions are required. The details are explained below.

5. Experiments

Numerical experiments have been carried out to find the effectiveness and efficiency of the proposed approach. A number of problems with varying sizes are solved using EA with various genetic parameter values. For the illustrative example, we consider 2 plants with 6 machines to produce 5 parts. Their lot sizes are $q=(40, 70, 60, 30, 60)$ and each plant has three machines, Plant 1 = {M1, M2, M3} and Plant

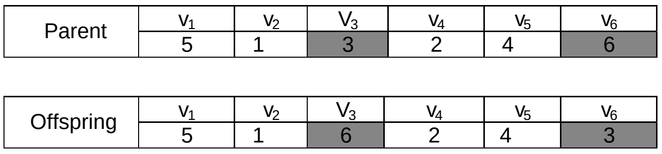

Fig. 9. The swap mutation operator.

```
procedure: EA based approach
  initialization
    t ← 0 ;
    initialize parent population P(t);
    evaluate P(t) and select the best solution σ* with the minimum objective values among P(t);
  while (no termination criteria) do
    obtain C(t) from P(t) by applying the genetic operators;
    evaluate C(t) and select the current best solution σ with the minimum objective values
     among C(t);
   update the best solution σ*, i.e., if σ < σ*, then σ* = σ;
   select P(t+1) from P(t) and C(t);
   t ← 1 + 1;
 end_while.
end_procedure.
```

Fig. 10. Overall procedure of EA-based approach.

2 = {M4, M5, M6}. Also, all unit load sizes are assumed to be 10 for all parts. The operations and their precedence constraints for 5 parts are given in Fig. 11.

The machining time for operations and their alternative machines are given in Table 1, and the set-up times between operations are obtained from random number generation from 1 to 50. The transportation times between machines are given in Table 2.

The transportation time per travel between plants is assumed to be 50, and the unit size per travel is equal to the lot size for each part.

To solve the problem using the proposed EA-based approach, the genetic parameters are set as maximum generation, *max_ge0* = 200; population size, *pop_size* = 100; crossover probability, $p_c = 0.8$; and mutation probability, $p_m = 0.2$. From this experiment, the proposed EA approach can be reached at the optimal solution at the 42th generation. The optimal makespan is 1792, and the results of operations sequences with machine selection and schedules are shown as Table 3. We also used commercial software, GAMS version 2.50, to solve the example problem. From the GAMS, we found the same optimal solution obtained EA approach.

We have investigated how the evolutionary parameters influence the performance of the proposed approach. The population size and the number of generations are the main genetic parameters that affect

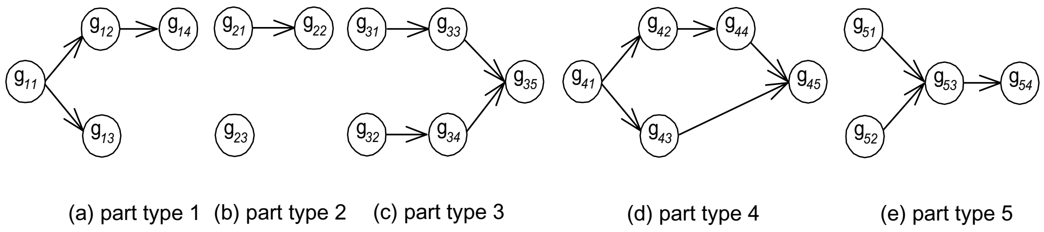

Fig. 11. Precedence constraints for parts.

Table 1
Machining time for operations and their alternative machines

		Part 1				Part 2			Part 3					Part 4					Part 5			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Plant 1	M1	7	7	–	6	–	3	8	–	10	6	15	–	–	–	–	–	5	–	–	13	–
	M2	–	–	6	–	9	5	–	–	–	5	–	6	–	5	–	5	–	8	7	–	–
	M3	–	–	5	–	–	–	12	5	–	–	–	–	6	–	6	–	–	–	10	–	7
Plant 2	M4	5	6	–	–	8	–	9	–	10	–	6	–	6	–	4	3	–	6	–	–	6
	M5	–	–	8	–	–	6	–	8	–	6	–	5	–	9	–	–	4	–	8	8	–
	M6	–	–	–	5	–	–	8	–	7	–	5	–	8	–	–	5	–	8	–	9	–

Table 2
Transportation times between machines

		Plant 1			Plant 2		
		M1	M2	M3	M4	M5	M6
Plant 1	M1	0	5	6	–	–	–
	M2	5	0	7	–	–	–
	M3	6	7	0	–	–	–
Plant 2	M4	–	–	–	0	5	6
	M5	–	–	–	5	0	7
	M6	–	–	–	6	7	0

the performance of the EA approach in the integrated problem. To explore the effects of these parameters, a set of computational experiments is performed. The data for each experiment is generated randomly. The first experiment has 5 parts with 21 operations, 2 plants, and 6 machines. The second experiment has 10 parts with 43 operations, 2 plants, and 6 machines. The third experiment has 15 parts with 64 operations, 3 plants, and 9 machines. The forth experiment has 20 parts with 86 operations, 3 plants, and 9 machines. The EA approach was coded in C++ language, and run on a PC with

Table 3
Operations sequences with machine selection

	Schedules				
Part 1	1-(M4) 0–200	3-(M3) 272–474	2-(M1) 500–780	4-(M6) 833–1033	
Part 2	6-(M1) 0–210	5-(M2) 432–1062	7-(M1) 1077–1637		
Part 3	9-(M6) 0–420	11-(M6) 453–753	8-(M3) 827–1127	10-(M2) 1135–1435	12-(M5) 1492–1792
Part 4	13-(M3) 0–180	14-(M5) 230–500	15-(M3) 579–759	16-(M6) 1033–1183	17-(M5) 1349–1469
Part 5	19-(M2) 0–420	18-(M4) 502–862	20-(M5) 868–1348	21-(M4) 1391–1751	

Table 4
Results on the various size problems

Parts	Oper.	Plants	Mach.	Test	Gen.	Pop.	*C*_rate	*M*_rate	Value
5	21	2	6	1	50	50	0.8	0.2	1865
				2	100	50	0.8	0.2	1865
				3	150	100	0.8	0.2	1865
				4	150	100	0.5	0.5	1865
				5	200	150	0.8	0.2	1865
10	43	2	6	1	100	50	0.8	0.2	2046
				2	100	50	0.5	0.5	2039
				3	150	100	0.8	0.2	2039
				4	150	100	0.5	0.5	2039
				5	200	150	0.8	0.2	2039
15	64	3	9	1	100	150	0.8	0.2	2157
				2	200	100	0.8	0.2	2186
				3	250	100	0.8	0.2	2157
				4	250	150	0.8	0.2	2157
				5	300	150	0.8	0.2	2157
20	86	3	9	1	200	50	0.8	0.2	2297
				2	250	100	0.8	0.2	2295
				3	300	100	0.8	0.2	2283
				4	350	100	0.8	0.2	2283
				5	400	100	0.8	0.2	2283

a Pentium-II CPU and 128 MB of RAM. Table 4 shows the average value of the objective over 5 runs for each parameter setting. For the first problem, the experimental probability of getting the best solution is about 100%. For the second experiment, the probability of getting the best solution is about 86%. For the third and forth experiments, the probability of getting the best solution is about 84 and 81%, respectively. From the results, we can see that the EA approach can find a good solution with very high probability.

6. Conclusion

In this paper, a model is developed to solve the APPS problem in multiple plants chain composed of a network of production facilities, of multiple product flow through two manufacturers. The objective is to determine optimal schedule of machine assignments and operations sequences of all parts so that the makespan is minimized.

The problem is formulated as a mixed integer programming model considering part mix, an unordered set of machining, alternative machines for each operation, machining times, and transportation times for all pairs of machines and inter plants.

To solve the problem, an EA approach is developed. To demonstrate the efficiency of the proposed EA approach on the APPS problem, the numerical experiment is carried out. From the results of the experiments, we see that the population size and the number of generations are the main factors that affect the performance of the EA approach in the integrated problem. We find that the EA approach can find a good solution with very high probability.

References

Bandeimarte, P., & Calderini, M. (1995). A heuristic bicriterion approach to integrated process plan selection and job shop scheduling. *International Journal of Production Research*, *33*(1), 161–181.

Finke, G., Claus, A., & Gunn, E. (1984). A two-commodity network flow approach to the traveling salesman problem. *Congressus Numerantium*, *41*, 167–178.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. New York: Wiley.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimizations*. New York: Wiley.

Hankins, S. L., Wysk, R. A., & Fox, K. R. (1984). Using a CATS database for alternative machine loading. *Journal of Manufacturing Systems*, *3*, 115–120.

Homem de Mello, L. S., & Sanderson, A. C. (1990). AND/OR graph representation of assembly plan. *IEEE Transactions on Robotics and Automation*, *6*(2), 188–199. April.

Horowitz, E., & Sahni, S. (1984). *Fundamentals of data structures in Pascal*. Computer Science Press.

Kusiak, A., & Finke, G. (1987). Modeling and solving the flexible forging module scheduling problem. *Engineering Optimization*, *12*, 1–12.

Morad, N., & Zalzala, A. (1999). Genetic algorithm in integrated process planning and scheduling. *Journal of Intelligent Manufacturing*, *10*, 169–179.

Nasr, N., & Elsayed, A. (1990). Job shop scheduling with alternative machines. *International Journal of Production Research*, *28*(9), 1595–1609.

Palmer, G. J. (1996). A simulated annealing approach to integrated production scheduling. *Journal of Intelligent Manufacturing*, *7*(3), 163–176.

Sundaram, R. M., & Fu, S. S. (1988). Process planning and scheduling. *Computer and Industrial Engineering*, *15*(1–4), 296–307.

Saygin, C., & Kilic, S. E. (1999). Integrating flexible process plans with scheduling in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology*, *15*, 265–280.

Tan, W. (2000). Integration of process planning and scheduling—a review. *Journal of Intelligent Manufacturing*, *11*, 51–63.