

Effective Genetic Approach for Optimizing Advanced Planning and Scheduling in Flexible Manufacturing System

Haipeng Zhang

Graduate School of Information, Production & Systems,
Waseda University,
808-0135, Kitakyushu, JAPAN
+81(90)-1161-7402
hpzhang@suou.waseda.jp

Mitsuo Gen

Graduate School of Information, Production & Systems,
Waseda University,
808-0135, Kitakyushu, JAPAN
+81(93)692-5273
gen@waseda.jp

ABSTRACT

In this paper, a novel approach for designing chromosome has been proposed to improve the effectiveness, which called multistage operation-based genetic algorithm (moGA). The objective is to find the optimal resource selection for assignments, operations sequences, and allocation of variable transfer batches, in order to minimize the total makespan, considering the setup time, transportation time, and operations processing time. The plans and schedules are designed considering flexible flows, resources status, capacities of plants, precedence constraints, and workload balance in Flexible Manufacturing System (FMS). The experimental results of various Advanced Planning and Scheduling (APS) problems have offered to demonstrate the efficiency of moGA by comparing with the previous methods.

Categories and Subject Descriptors

F.2.1 [Numerical Algorithms and Problems]

General Terms

Algorithms.

Keywords

Resource Selection and Operation Sequences, Multistage Operation-based Genetic Algorithm (moGA).

1. INTRODUCTION

The planning and scheduling of manufacturing systems always require resource capacity constraints, disjunctive constraints and precedence constraints, owing to the tight due dates, multiple customer-specific orders, and flexible process strategies [1]. Advanced planning and scheduling (APS) is very important in this manufacturing environment, because it is at the leading edge of Flexible Manufacturing System (FMS). APS includes a range of capabilities from finite capacity planning at the plant floor level through constraint-based planning to the latest applications of

advanced logic for supply chain planning and collaboration [2]. Moreover, APS is likely the most important functions in a multi-plant chain planning, because it is essential to reduce lead times, and lower inventories, provide available-to-promise information, and increase throughput. APS evolved from the manufacturers' need for better process planning and scheduling tools [3].

During the past several years, many researchers have put great effort into the area on integrated process planning and scheduling problem. Tan (2000) reported a briefly review of the research in the process planning and scheduling area and discussed the extent of applicability of various approaches [4], and also proposed a linearized polynomial mixed integer programming model for this problem [5]. Kolisch and Hess (2000) considered the problem of scheduling multiple, large-scale, make-to-order assemblies under resource, assembly area, and part availability constraints. They introduced three efficient heuristic solution methods, a biased random sampling method and two Tabu Search-based large-step optimization methods [6]. Dellaert et al. (2000) discussed multi-level lot-sizing problem in material requirements planning (MRP) systems [7]. They developed a binary encoding genetic algorithm (GA) and design five specific genetic operators to ensure that exploration takes place within the set of feasible solutions. Chung et al. (2000) considered a kind of job shop scheduling problem where each job has its due date and temporal adjustment of machine capacity is possible through outsourcing. They presented a heuristic solution procedure that addressed the total job-shop scheduling problem by solving a series of smaller sub-problems to optimality [8]. Gen and Syalif proposed a spanning tree-based Genetic Algorithm for solving production/distribution problem to determine an efficient integration of production, distribution and inventory system [9].

Recently, for improving the flexibility of machine assignment, Kacem et al. (2002) proposed a genetic algorithm controlled by the assigned model which is generated by the approach of localization [10]. Najid (2002) used simulated annealing (SA) for optimizing the flexible assignment of machines in flexible Job-shop Scheduling Problem [11]. Lopez and Ramirez (2005) newly describe the design and implementation of a step-based manufacturing information system to share flexible manufacturing resources data [12].

All researchers above considered the alternative machines for each operation, and they wanted to apply their models for solving the flexible assignment of various resources (machines). However, there exists a weakness which is fixing all the operation sequence

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-186-4/06/0007...\$5.00.

or non-constraint operation sequence. That is, they ignore the flexibility especially for orders and there are some precedence constraints in each orders but the sequencing is alternative.

Furthermore, recently manufacturers realized the importance of being responsive to the customer. As a result, the simplistic theory of constraint and queuing models for APS designed to address bottleneck problems or work station sequencing is being replaced. The next generation of APS system will need to handle transaction level scheduling, MPC in outsourcing environment, and resource capacity requirements in multiple plants, and they will also need to be able to provide precise capable-to-promise dates across multiple sites. APS methods tend to take a holistic and collaborative approach to provide global optimization. This new business model is an attempt to optimize not only plant operations but all the activities from a supplier to a customer. This collaborative approach brings the idea of extending a MPC beyond a production site. Especially in recent work by Moon et al [13][14], GA approaches are proposed to solve such kinds of APS problems considering MPC environment. However, it encoded the chromosome only considering the information of operation sequence, and hybridized with some heuristic strategy for resource selection (by minimum processing time). This approach actually improved the efficiency of convergence, but may skip some optimal solution. To avoid this limitation, our proposed approach is built on the basic concept of a multistage decision making (MSDM) model. We separate all the operations into a set of stages, and in each stage (operation) several alternative states (machines) are offered for selection. Hence our job is to make a decision in all the stages for choosing states and obtain an optimal schedule. For this reason we propose a new multistage operation-based Genetic algorithm (moGA), and define the chromosome with two vectors which contain both of the two information types, i.e., operation sequence and machine selection.

2. APS PROBLEM DEFINITION

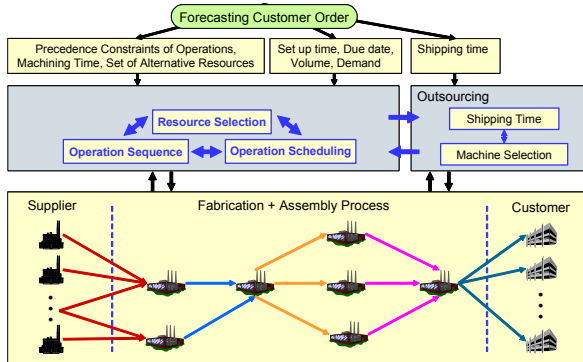


Figure 1. Modern APS System

The APS problem includes finding the optimal resource selection for operations, operations sequences, allocation of variable transfer batches, and schedules considering flexible flows, resources status, capacities of plants, precedence constraints, and workload balance. The relationship between all the main parts of APS problems in MPC is shown in Figure 1. We find the process is driven since the orders come from our customer. Moreover, to satisfy the requirements, some other constraints should be considered such as due date, setup time and shipping time. The

main integrated model involves the time allocation for operations based on the selected operations sequences to minimize the makespan. Thus, the operations sequencing problem should be integrated into the scheduling problem. We allow any two operations belonging to the same order proceed concurrently if resources are available. In multi-plant environment, we should also consider the situation of outsourcing that means the same order can also be delivered to some other plant for assigning to the resources in different locations; therefore, a lot of orders can be divided further for the subsequent operations if it is allowed by the load size of transfer device. In this case, the transfer batch and process batch may not coincide. On the other hand, if inter-plants transportation is needed, then the transfer batch should be equal to the process batch.

To clearly demonstrate the process planning in the APS problem in detail we create a simple example shown in Figure 2, which presents the two kinds of materials to be machined in a modern manufacturing system with the lot sizes 40 and 50 orders by the customer. Concretely, 10 volumes should be removed from two materials for obtaining the final two products. All the manufacturing plans of this example are offered in Table 1, which includes all the types of operations and the corresponding machine selection.

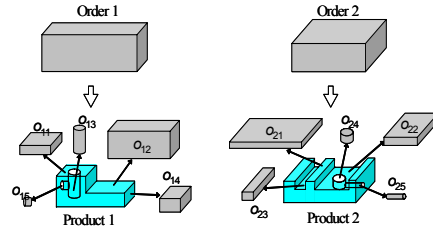


Figure 2. Simple example for process planning problem

Table 1. Manufacturing Plan

Operation Index (Order k , Operation o_{ki})	Operation Type	Machine Selection M_m
1 (1, o_{11})	milling	M_1, M_4
2 (1, o_{12})	milling	M_1, M_4
3 (1, o_{13})	drilling	M_2, M_3, M_5
4 (1, o_{14})	milling	M_1
5 (1, o_{15})	drilling	M_2
6 (2, o_{21})	milling	M_1, M_2
7 (2, o_{22})	milling	M_1, M_3
8 (2, o_{23})	milling	M_3, M_5
9 (2, o_{24})	drilling	M_1, M_4, M_5
10 (2, o_{25})	drilling	M_1, M_2

In Figure 3, we can describe operation sequence constrains of the two orders by using node graphs, and, we can confirm that operation sequence is alternative and follow some precedence constraints for each order. For instance, in order 1: both $\{o_{11}, o_{13}, o_{12}, \dots\}$ and $\{o_{11}, o_{12}, o_{13}, \dots\}$ are legal, while $\{o_{13}, o_{11}, o_{12}, \dots\}$ is illegal.

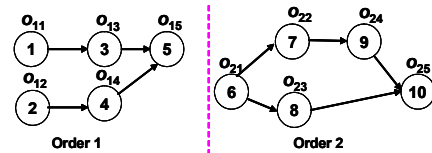


Figure 3. Operation precedence constraints of two orders

Tables 2 and 3 prepare the data set of processing time for each operation and transition time between different machines. Table 3 shows that five machines have different capabilities for each corresponding operation, and are not available for some of them. In addition, concerning this simple processing planning example, we do not consider the setup time and both of the orders are completed in one plant.

Table 2. Processing time p_{kim} of operations

Operation Machine	Order 1					Order 2				
	1	2	3	4	5	6	7	8	9	10
M_1	7	7	-	6	-	3	8	-	10	6
M_2	-	-	6	-	9	5	-	-	-	5
M_3	-	-	5	-	-	-	12	5	-	-
M_4	5	6	-	-	-	-	-	-	10	-
M_5	-	-	8	-	-	-	-	8	7	-

Table 3. The transition time t_{mn}^S between different machines

$M_m \backslash M_n$	M_1	M_2	M_3	M_4	M_5
M_1	-	7	27	26	17
M_2	19	-	15	19	10
M_3	5	17	-	36	27
M_4	8	20	4	-	30
M_5	18	18	14	5	-
Available capacity	1500	1500	1500	1500	1500

A process plan should also be able to represent all the possible precedents that occur during the planning and processing decisions. From an unordered set of operations with precedence relations, operations sequencing is to determine a sequence considering the combination of parallel processes and alternative resources for operations. Nevertheless, we should clarify the simple example in this section only indicates the planning horizon in single plant, while the APS in multi-plant chain will be solved in experimental section.

As shown previously, the objective of an APS system in a multi-plant chain is usually to determine an optimal schedule with operation sequences for all the orders (jobs). That is, the problem we are treating can be defined as: there are a set of K orders which are to be processed on N machines with alternative operations sequences and alternative machines for operations in the environment of the multi-plant chain, we want to 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 the makespan minimization.

3. MATHEMATICAL MODELS

To formulate the mathematical model, some notations and symbols are defined firstly as follows:

Indices

- i, j : index of operation number, $i, j = 1, 2, \dots, J_k$
- k, l : index of orders, $k, l = 1, 2, \dots, K$
- m, n : index of machines, $m, n = 1, 2, \dots, N$
- d, e : index of plants, $d, e = 1, 2, \dots, D$

Parameters

- K : number of orders
- D : number of plants
- N : number of machines
- O_k : set of operations for order k , i.e., $O_k = \{o_{ki} | i=1, 2, \dots, J_k\}$
- o_{ki} : the i th operation for order k
- J_k : number of operations for order k
- M_m : the m th machine
- q_k : lot size of order k
- A_m : set of operations that can be processed on machine m
- p_{kim} : unit processing time of operation o_{ki} on machine m
- L_m : capacity of machine m
- r_{kij} : precedence constraints

$$r_{kij} = \begin{cases} 1, & \text{if } o_{ki} \text{ is predecessor of } o_{kj} \text{ in order } k \\ 0, & \text{otherwise} \end{cases}$$

b_{md} : capability of resources in plant d

$$b_{md} = \begin{cases} 1, & \text{if } m \text{th machine belongs to plant } d \\ 0, & \text{otherwise} \end{cases}$$

t_{kij}^U : setup time from operation o_{ki} to operation o_{kj}

u_{kij} : unit load size of order k from operation o_{ki} to operation o_{kj}

c_M : total makespan for all the orders

d_D : due date

B_d : set of machines that are included in plant d

t_{mn} : unit shipping time between machine m to machine n

$$t_{mn} = \begin{cases} t_{mn}^S, & \text{if resources } m \text{ and } n \text{ are included in the same plant,} \\ t_{mn}^D, & \text{otherwise.} \end{cases}$$

v_{kij} : number of shipping times from operation o_{ki} to operation o_{kj}

$$v_{kij} = \begin{cases} \left\lceil \frac{q_k}{u_{kij}} \right\rceil, & \text{if resources } m \text{ and } n \\ & \text{are included in the same plant,} \\ 1, & \text{otherwise.} \end{cases}$$

T_{kij} : transition time from operation o_{ki} to operation o_{kj}

$$T_{kij} = \sum_{m=1}^N \sum_{n=1}^N \{q_k p_{kim} x_{kim} + v_{kij} t_{mn} x_{kim} x_{kjn} + t_{kij}^U y_{kij}\}$$

c_{ki} : completion time of operation o_{ki}

$$c_{ki} = s_{ki} + q_k \sum_{m=1}^N p_{kim} x_{kim}$$

Decision Variables

$$y_{kij} = \begin{cases} 1, & \text{if operation } o_{ki} \text{ is performed} \\ & \text{immediately before operation } o_{kj} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{kim} = \begin{cases} 1, & \text{if machine } m \text{ is selected for operation } o_{ki} \\ 0, & \text{otherwise} \end{cases}$$

s_{ki} : starting time of operation o_{ki}

The objective to APS problem in this paper is to minimize total makespan t_M , the overall model can be described as follows:

$$\min c_M = \max_{k, i} \{c_{ki}\} \quad (1)$$

$$\text{s. t. } \left\{ s_{lj} - (c_{ki} + t_{kij}^U) \right\} x_{kim} x_{ljm} y_{kij} \geq 0 \quad \forall (k, i), (l, j), m \quad (2)$$

$$\left\{s_{kj} - (s_{ki} + u_{kij} p_{kim} + t_{mn}^S)\right\} r_{kij} x_{kim} x_{ijn} \geq 0 \quad \forall i, j, k, \forall m, n \in B_d, \forall d \quad (3)$$

$$\left\{c_{kj} - u_{kij} p_{kin} - (c_{ki} + t_{mn}^S)\right\} r_{kij} x_{kim} x_{ijn} \geq 0 \quad \forall i, j, k, \forall m, n \in B_d, \forall d \quad (4)$$

$$\left\{s_{kj} - (c_{ki} + t_{mn}^D)\right\} r_{kij} x_{kim} x_{ijn} \geq 0 \quad \forall i, j, k, \forall m \in B_d, \forall n \in B_e, \forall d, e \quad (5)$$

$$\sum_{k=1}^K \sum_{i=1}^{J_k} q_k p_{kim} x_{kim} \leq L_m \quad \forall m \quad (6)$$

$$r_{kij} y_{kij} = 0 \quad \forall i, j, k \quad (7)$$

$$y_{kiki} = 0 \quad \forall i, k \quad (8)$$

$$y_{kij} + y_{lji} = 1 \quad \forall (k, i), (l, j), (k, i) \neq (l, j) \quad (9)$$

$$\sum_{m=1}^N x_{kim} = 1 \quad \forall i, k \quad (10)$$

$$x_{kim} = 0 \quad \forall (k, i) \notin A_m, \forall m \quad (11)$$

$$s_{ki} \geq 0 \quad \forall i, k \quad (12)$$

$$y_{kij} \in \{0, 1\} \quad \forall (k, i), (l, j) \quad (13)$$

$$x_{kim} \in \{0, 1\} \quad \forall i, k, m \quad (14)$$

Equation (2) imposes that for any resource (machine), it can not be selected for one operation until the predecessor is completed and also set up time must be considered (see Figure 4).

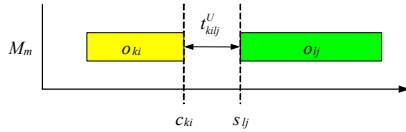


Figure 4. Time chart for the constraints on the same machine

Equations (3) and (4) impose the transportation instances in local plant. Both of the two constraints must be satisfied simultaneously to ensure operations can be run uninterrupted on one machine (see Figure 5).

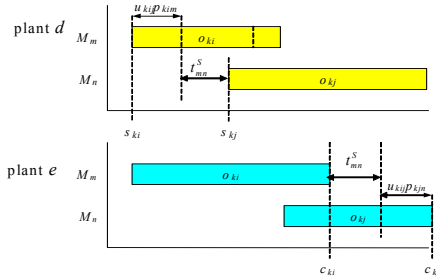


Figure 5. Time chart for the constraints in the same plant

Equation (5) restricts the other transportation between different plants (in MPC environment), which indicate that the operation can not move to another plant until all the lot size have been finished (see Figure 6)

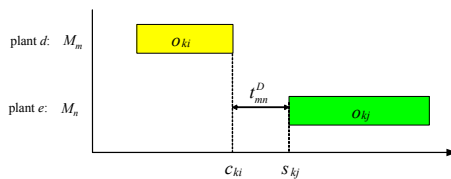


Figure 6. Time chart for the constraints in different plants

Equation(6) restricts the available capacity for each machine. Equation (7) ensures that the precedence constraints are not violated. Equations (8) and (9) ensure the feasible operation sequence. Equations (10) and (11) ensure the feasible resource selection. Equations (12), (13) and (14) impose nonnegative condition.

4. MULTISTAGE OPERATION-BASED GENETIC ALGORITHM (MOGA)

4.1 Moon-Kim-Gen's Approach

Several related works by Moon [13][15] have reported a GA approach especially for solving such kinds of APS problems. However, to derive a feasible complete schedule, they only considered the optimal operation sequence, but selected the resources in terms of minimum processing time. That is, for machines assignment we consider that the minimum processing time assignment is the optimal choosing strategy for the solution. However the transition time between plants, and setup time between operations are ignored. Anyway, Moon-Kim-Gen's approach will inevitably fail to find some optimal solutions for some large size examples, even if it can improve the searching speed to some degree.

4.2 Designing Chromosome

Following the development of GA, neither the optimization of GA's parameter (crossover rate and mutation rate) setting, nor the approach of GA's operators (crossover approach and mutation approach) can significantly improve the effectiveness of algorithm. Hence, more and more researcher tried to find an optimal designing of chromosome, which contains more information and can also improve both effectiveness and efficiency of the algorithm to the corresponding combinatorial optimization problem [16]. Especially, some researchers used two dimensional schemes: Ulusoy et al. (1997) used a two-array representation (one for operation sequencing and the other for AGV assignment) [17], Gonçalves et al. (2005) also used a two string representation, one for operation priorities and the other for delay times [18].

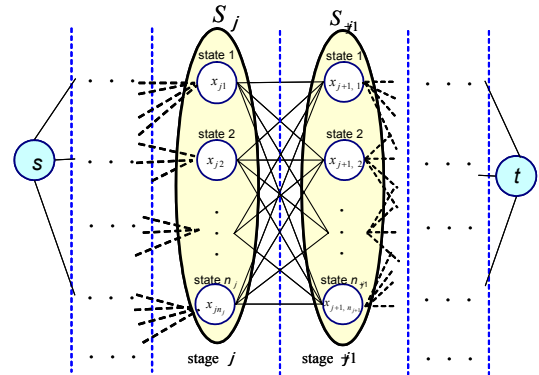


Figure 7. Basic concept of multistage decision making model

Originally, our idea of moGA comes from the basic concept of a multistage decision making model shown in Figure 7. There are several stages separating the route from the starting node to the terminal node, and in each stage several states are offered to be

chosen from. After we make all the decisions for choosing states, we can draw a solution [19], the fitness of the result is in terms of the different decisions made along the route, and such kind of optimization has already been used in Zhang and Gen (2005) especially for solving flexible job-shop scheduling problem [20].

Since that both operation sequence and machine selection can affect the solution in an APS problem, the chromosome presentation of multistage operation-based Genetic Algorithm (moGA) for APS problems consists of two parts:

- Priority-based encoding for operation sequence;
- Machine permutation encoding for machine selection;

Phase 1: Sequencing Operations

Phase 1 is a procedure to get a feasible fixed operation sequence, hence we input the operation set for all the orders, and the precedence constraints in each order. After legalization (making precedence feasible), we can output the legal fixed operation sequence.

In this phase, we use the priority encoding procedure to formulate chromosomes, and draw a chromosome for the simple example in Section 2 as shown in Figure 8:

Node ID j :	1	2	3	4	5	6	7	8	9	10
Priority $v_1(j)$:	5	1	7	9	4	6	3	8	2	10

Figure 8 Chromosome v_1 drawn by priority-based encoding

Following the precedence constraints, one feasible operation sequence for the simple example in Section 2 will be obtained as follows:

a final feasible path = $\{v_2, v_1, v_6, v_7, v_9, v_3, v_8, v_4, v_5, v_{10}\}$
 operation sequences = $\{o_{12}, o_{11}, o_{21}, o_{22}, o_{24}, o_{13}, o_{23}, o_{14}, o_{15}, o_{25}\}$

Phase 2: Selecting Machines

After finishing Phase 1, we draw a fixed operation sequence, which means the position of all stages (operations) has been decided. Hence, in Phase 2 we input the fixed operation sequence, processing time data, setup time data, and transition time, and after states (machines) assignment, we will output the whole schedule and solution of makespan.

Since the APS problem involves the flexible machine selection, we use a machine permutation coding procedure in this paper to make another part of the chromosome. That is, we will firstly build a multistage operation-based frame shown in Figure9.

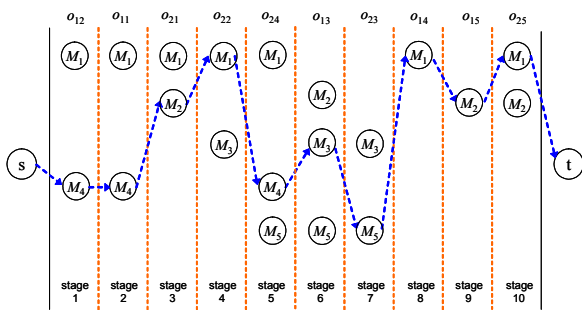


Figure 9. Node graph of machine selection in moGA

It is obvious to find the number of stages is just the total operation number, and also in each stage, the machines available are treated as the corresponding state. We can randomly generate second moGA chromosome as follows in Figure 10.

Node ID j :	1	2	3	4	5	6	7	8	9	10
machine $v_2(j)$:	4	4	3	1	2	2	1	5	4	1

Figure 10 Chromosome v_2 drawn by machine permutation encoding

So, we can assign the machine selection to the fixed stage sequence offered by Phase 1, and finally draw the feasible solution as follows:

Schedule :

S = {Order 1, Order 2}
 = $\{(o_{12}, M_4:0-240), (o_{11}, M_4:240-440), (o_{13}, M_3:294-494), (o_{14}, M_1:469-709), (o_{15}, M_2:536-896), (o_{21}, M_2:0-250), (o_{22}, M_1:69-469), (o_{24}, M_4:440-940), (o_{23}, M_5:650-1050), (o_{25}, M_2:828-1128)\}$

After calculating the makespan using the data in Tables 2 and 3, we can draw the Gantt chart shown in Figure 10:

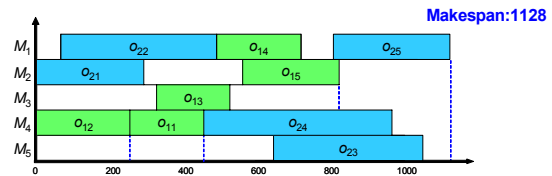


Figure 10. Gantt chart for the best solution

4.3 Genetic Operators

Crossover

Especially for the first phase (operation sequence), we use one-cut point method, which randomly select one cut point. However, to ensure the feasibility of the new offspring, we combined the mapping strategy as showing in Figure 11.

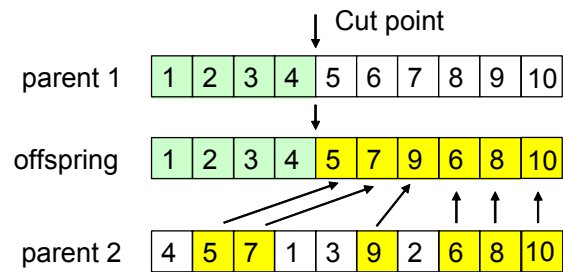


Figure 11. One-cut point crossover for phase 1

Mutation

Mutation 1: For the first phase (operation sequence), we adapt swap mutation to generate the new offspring as showing in Figure 12.

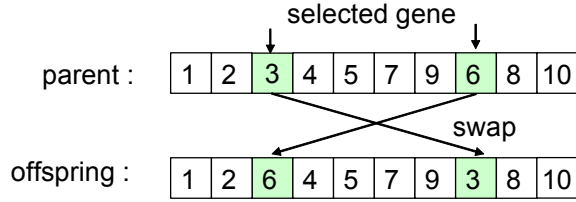


Figure 12. Swap mutation for phase 1

Mutation 2: While for the special case of phase 2 (machine selection), we use neighbor search mutation. Moreover, we firstly define a neighborhood to keep the feasibility. In Figure 13, after randomly select the third position gene, we can do mutation around the known neighborhood {2, 3, 5} (indicate to M_2, M_3, M_5).

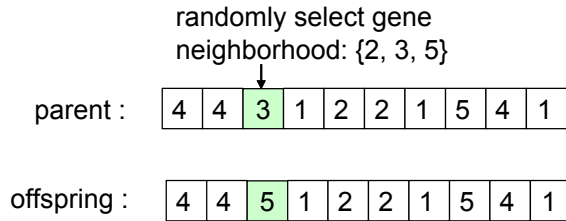


Figure 13. Neighbor search mutation for phase 2

4.4 Overall Procedure

Finally, we can design the overall procedure especially for the APS problem using moGA which is shown in Figure 14. In this procedure, we defined $P(t)$ and $C(t)$ respectively as populations for parent and offspring in generation t . In overall procedure, we choose the appropriate function value of crossover and mutation corresponding to the two vectors of the chromosome, especially only one crossover for Phase 1 because, after the operation sequence is decided, we do not need to change the stages (operations) position in Phase 2 for choosing states (machines).

procedure: moGA for APS

input: APS data set, GA parameters

output: best solution

begin

$t \leftarrow 0$;

initialize $P_1(t)$ (consisting of $V_1(\bullet)$) by priority encoding;

initialize $P_2(t)$ (consisting of $V_2(\bullet)$) by machine permutation encoding;

fitness $eval(P_1, P_2)$;

while (not termination condition) **do**

 crossover $P_1(t)$ to yield $C_1(t)$ by one-cut point crossover;

 mutation $P_1(t)$ to yield $C_1(t)$ by swap mutation;

 mutation $P_2(t)$ to yield $C_2(t)$ by neighbor search mutation;

 fitness $eval(C_1, C_2)$ by priority decoding and machine permutation decoding;

 select $P(t+1)$ from $P(t)$ and $C(t)$ by roulette wheel selection;

$t \leftarrow t+1$;

end

output best solution;

end

Figure 14. Overall procedure for APS using moGA

5. NUMERICAL EXPERIMENTS

In this paper, we firstly consider a relatively small size problem with 2 plants with 6 resources to treat four orders. The lot sizes for the orders are $q = (40, 70, 60, 30)$ and each plant has three resources. Plant 1 = $\{M_1, M_2, M_3\}$ and Plant 2 = $\{M_4, M_5, M_6\}$. Their available capacities are:

$$L_1 = 1000, L_2 = 1000, L_3 = 2000, \quad L_4 = 2000, \quad L_5 = 2000,$$

$L_6 = 2000$, The unit load size for transportation is assumed to be 10 for all orders. The operations and their precedence constraints for the 4 orders are given in Figure 15. The transportation times between resources are given in Table 4, and the processing times for each operation and their alternative resources are given in Table 5. The setup times between operations are given in Table 6.

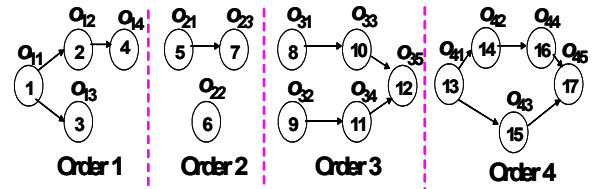


Figure 15. Operation precedence constraints

Table 4. Transition time t_{mn}^S between machines

		Plant 1					Plant 1		
		M_1	M_2	M_3			M_4	M_5	M_6
Plant 1	M_1	0	5	6	Plant 2	M_4	0	5	6
	M_2	5	0	7		M_5	5	0	7
	M_3	6	7	0		M_6	6	7	0
Available capacity		1000	1000	2000	Available capacity		2000	2000	2000

Table 5. Processing time p_{kim} for operations in alternative machines

operation		Order 1			Order 2			Order 3				Order 4						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Plant 1	M_1	7	7	-	6	-	3	8	-	10	6	15	-	-	-	-	-	5
	M_2	-	-	6	-	9	5	-	-	5	-	6	-	5	-	5	-	-
	M_3	-	-	5	-	-	-	12	5	-	-	-	-	6	-	6	-	-
Plant 2	M_4	5	6	-	-	8	-	9	-	10	-	6	-	6	-	4	3	-
	M_5	-	-	8	-	-	6	-	8	-	6	-	5	-	9	-	-	4
	M_6	-	-	-	5	-	-	8	-	7	-	5	-	8	-	-	5	-

Table 6. Setup time t_{kij}^U between operations

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-	17	36	6	37	20	11	30	5	32	30	36	23	21	1	28	20
2	42	-	32	3	2	15	15	22	44	39	30	37	47	12	5	38	31
3	6	6	-	37	26	0	23	29	12	5	13	3	13	24	29	26	8
4	12	3	40	-	19	46	31	30	31	49	49	27	39	45	9	0	3
5	2	48	43	25	-	49	10	11	4	8	17	39	34	31	11	0	24
6	24	26	43	31	49	-	22	31	21	43	31	10	30	23	2	34	38
7	20	45	28	43	22	16	-	39	46	25	43	34	9	22	38	12	7
8	15	41	44	35	14	10	30	-	2	14	7	8	22	3	18	45	18
9	25	47	22	21	47	39	26	0	-	22	33	7	37	20	25	20	7
10	3	46	9	10	35	18	5	21	24	-	33	40	22	23	41	37	31
11	1	17	31	3	30	15	23	21	37	3	-	15	23	32	3	46	6
12	4	18	41	37	26	39	43	46	44	28	13	-	45	47	7	32	2
13	18	45	24	27	47	21	8	21	35	38	26	39	-	21	2	12	33
14	48	37	46	44	25	24	1	8	38	46	48	37	6	-	6	41	10
15	43	3	39	3	44	17	46	24	46	33	9	16	15	4	-	4	12
16	9	44	40	21	16	12	36	37	44	16	41	31	7	3	8	-	44
17	14	21	5	29	44	48	8	31	10	44	1	7	18	2	11	22	-

The transportation time t_{mn}^D per trip between plants is assumed to be 50, and the unit size per trip is equal to the lot size of each order. To solve the problem using the multistage operation-based Genetic Algorithm (moGA) approach, the genetic parameters are set to maximum generation $maxGen = 200$; population size $popSize = 100$; crossover probability $p_C = 0.7$ and mutation probability $p_M = 0.3$.

The makespan of the best solution is 1,102, with the corresponding chromosomes and schedules are shown as follows:

index,j:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$v_1(j)$:	2	3	7	13	8	1	9	4	11	5	12	17	6	10	14	15	16
$v_2(j)$:	4	4	3	6	2	1	1	3	6	5	4	5	6	2	3	2	1

$$S = \{\text{Order 1, Order 2, Order 3, Order 4}\}$$

$$= \{(o_{11}, M_4: 0-200), (o_{12}, M_4: 217-457), (o_{13}, M_3: 551-751), (o_{14}, M_6: 893-1093), (o_{22}, M_1: 0-210), (o_{21}, M_2: 35-665), (o_{23}, M_1: 232-792), (o_{31}, M_3: 0-300), (o_{33}, M_5: 350-710), (o_{32}, M_6: 452-872), (o_{34}, M_4: 578-938), (o_{35}, M_5: 750-1050), (o_{41}, M_6: 0-240), (o_{42}, M_2: 696-846), (o_{43}, M_3: 782-962), (o_{44}, M_2: 890-1040), (o_{45}, M_1: 952-1102)\}$$

The best schedule in detail is shown in Table 7. Furthermore, we can also draw the resource utilization factor for this experiment shown in Figure 16.

Table 7. Best schedule

Plant d	Machine M_d	Operation o_{ij} (start time – finishing time)		
Plant 1	M_1	$o_{21}(0-210)$	$o_{22}(232-792)$	$o_{45}(952-1102)$
	M_2	$o_{23}(35-665)$	$o_{41}(696-846)$	$o_{44}(890-1040)$
	M_3	$o_{31}(0-300)$	$o_{33}(551-751)$	$o_{43}(782-962)$
Plant 2	M_4	$o_{11}(0-200)$	$o_{12}(217-457)$	$o_{13}(578-938)$
	M_5	$o_{35}(350-710)$	$o_{32}(750-1050)$	
	M_6	$o_{41}(0-240)$	$o_{42}(452-872)$	$o_{44}(893-1093)$

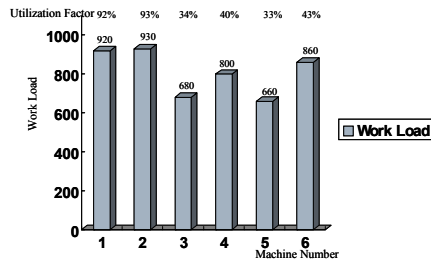


Figure 16. Resource utilization

As in the Gant chart shown in Figure 17, we compare the best results with the solution obtained by Moon-Kim-Gen's approach. The figure obviously presents the reason why they miss chances to find the best solution. It is because they only consider the minimum processing time. For example, operations in order 4 changed plants twice, but the best schedule of ours only once. They neglect the transportation time between Plant 1 and Plant 2.

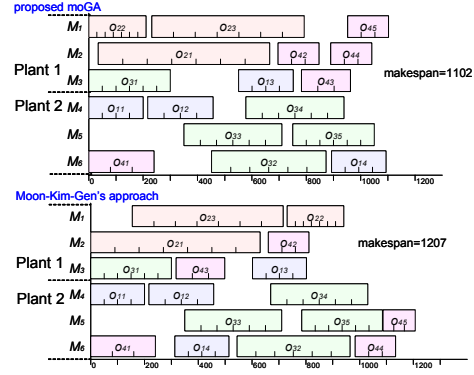


Figure 17. Gant chart of the best schedule

Moreover, we set the parameter $p_C=0.7$ (for crossover), $p_M=0.3$ (for mutation), but changing the $maxGen$ and $popSize$ into 4 different tests within each experimental data, set the comparison of results are shown in Table 8.

Table 8. Experimental result comparisons for different maxGen and popSize

No. of orders	No. of operations	No. of plants	No. of machines	maxGen	popSize	C_M
4	17	2	6	100	20	1102*
				100	50	1102*
				200	100	1102*
				500	100	1102*
5	21	2	6	100	50	1402
				100	100	1370*
				200	100	1370*
				500	100	1370*
8	33	3	9	100	50	1680
				100	100	1557
				200	100	1513*
				500	150	1513*
15	64	3	9	200	50	3408
				200	100	2356
				500	100	2239
				500	150	2207*
29	124	5	15	200	50	3106
				200	150	2920
				500	150	2862
				500	200	2798*

Furthermore, to prove the efficiency of our approach, we compared our experimental results with Moon-Kim-Gen's approach (Moon, Kim & Gen, 2004) by using the same experimental data. All the data in shadow and marked with "*" are best known solution (by enumerative algorithm operated on multiprocessor computer). The solution shown in Table 9 illustrate that using moGA can get better solution in some large cases. That means, in some large cases, the assignment of machines may not obey the strategy for minimum processing time selection.

Table 9. comparisons of experimental result

No. of orders	No. of operations	No. of plants	No. of machines	Moon-Kim-Gen's approach	moGA
4	17	2	6	1207	1102*
5	21	2	6	1561	1370*
8	33	3	9	1902	1513*
15	64	3	9	2640	2207*
29	124	5	15	3246	2798*

6. CONCLUSION

In this paper we have addressed the multistage operation-based genetic algorithm (moGA) approach to solve the APS problems in FMS. In order to minimize the makespan, we should find an optimal resource selection for assignments and operations sequences simultaneously. Hence we use the concept of multistage to formulate an efficient model, and divide the problem into 2 main phases through analyzing the character of APS problems. Some numerical experiment has been offered to prove the efficiency of moGA. The result shows that moGA can find a better solution when enlarging the size of the APS problem. In the end, we analyze the experimental result in detail, and get the optimal parameter setting of our proposed moGA, and also we compared our approach with Moon-Kim-Gen's approach using the same data. The results of various sizes of numerical experiments have demonstrated the efficiency of moGA by comparing it with the previous methods. Finally, we can consider that the proposed approach can be effectively used to solve the complex and large sized mixed integer programming model problem in a FMS.

7. ACKNOWLEDGEMENT

This work was partly supported by Waseda University Grant for Special Research Projects 2004 and the Ministry of Education, Science and Culture, the Japanese Government: Grant-in-Aid for Scientific Research (No.17510138).

8. REFERENCES

- [1] Kusiak, A. *Computational Integrated in Design & Manufacturing*. John Wiley & Sons, New York, 2000.
- [2] Turbide, D. *Advanced Planning and Scheduling (APS) Systems*. *Midrange ERP Magazine*, 1998.
- [3] Eck, M. *Advanced Planning and Scheduling: Is logistic everything? A research on the use (fullness) of advanced planning and scheduling systems*. Technique report, Vrije University, Amsterdam, Holland, 2003.
- [4] Tan, W. Integration of process planning and scheduling - a review. *Journal of Intelligent Manufacturing*, 11,1, 2000, 51-63.
- [5] Tan, W. A linearized polynomial mixed integer programming model for the integration of process planning and scheduling. *Journal of Intelligent Manufacturing*, 15, 5, 2004, 593-605.
- [6] Kolisch, R., and Hess, K. Efficient methods for scheduling make-to-order assemblies under resource, assembly area and part availability constraints. *International Journal of Production Research*, 38, 2000, 207-228.
- [7] Dellaert, N., Jeunet, J., and Jornard, N. A genetic algorithm to solve the general multi-level lot-sizing problem with time-varying costs. *International Journal of Production Economy*, 68, 2000, 241-257.
- [8] Chung, D., Lee, K., Shin, K., and Park, J. An algorithm for job shop scheduling problem with due date constraints considering operation subcontract. In *Proc. of 15th International Conference on Production Research*, 2000.
- [9] Gen, M., and Syalif, A. Hybrid genetic algorithm for multi-time period production /distribution planning. *Computers & Industrial Engineering*, 48, 4, 2005, 799-809.
- [10] Kacem, I., Hammadi, S., and Borne, P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 32, 1, 2002, 408-419.
- [11] Najid, N.M., Dauzere-Peres, S., and Zaidat, A. A modified simulated annealing method for flexible job shop scheduling problem. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 5, 2002, Pages 6.
- [12] Lopez, O., and Ramirez, M. A STEP-based manufacturing information system to share flexible manufacturing resources data. *Journal of Intelligent Manufacturing*, 16, 3, 2005, 287-301.
- [13] Moon, C., Kim, J. S., and Gen, M. Advanced planning and scheduling based on precedence and resource constraints for e-plant chains. *International Journal of Production Research*, 42, 15, 2004, 2941-2955.
- [14] Moon, C., and Seo, Y. Evolutionary Algorithm for Advanced Process Planning and Scheduling in a Multi-plant. *Computer and Industrial Engineering*, 48, 2, 2005, 311-325.
- [15] Moon, C. *Evolutionary System Approach for Advanced Planning in Multi-Plant Chain*. PhD thesis, Waseda University, Japan, 2004.
- [16] Gen, M., and Cheng R. *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
- [17] Ulusoy, G., Sivrikaya-Serifoglu, F., and Bilge, U. A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, 24, 4, 1997, 335-351.
- [18] Gonçalves, J. F., Magalhães Mendes J. J., and Resende, M. G. C. A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167, 1, 2005, 77-95.
- [19] Gen, M., and Cheng, R. *Genetic Algorithms & Engineering Optimization*. John Wiley & Sons, New York, 2000.
- [20] Zhang, H., and Gen, M. Multistage-based genetic algorithm for flexible job-shop scheduling problem. *Journal of Complexity International*, 11, 2005, 223-232.