A Hybrid Genetic Algorithm-based Approach to Solve Parallel Machine Scheduling with Job Delivery Coordination

Cheng-Hsiang Liu, Tzu-Ling Wu and Pei-Shiun Lin

***Abstract*—This paper considers a problem in which orders are processed by either one of two parallel machines and delivered by a single delivery truck to one customer area. Coordination among production stage and distribution stage in the supply chain to achieve ideal overall system performance has become more practical and has received a lot of attention from both industry practitioners and academic researchers. The coordinated scheduling problem of production and distribution operations deals with scheduling orders on the two identical parallel machines and grouping the completed orders into batches for delivery. To solve this complex problem, a regular genetic algorithm (GA) and an efficient approach which is based on a hybrid of GA and a parallel scheduling procedure (PSP) and is called hybrid GA (HGA), are proposed. Experimental results demonstrate that the regular GA and HGA perform very well with respect to the objective function. Besides, the HGA can find even or better solution in a shorter period of time than the regular GA. Thus, the proposed HGA should be the scheduling approach of choice.**

***Index Terms*—parallel machines scheduling, genetic algorithms, production-distribution coordination.**

I. INTRODUCTION

There are many production systems with transportation operations where finished jobs are delivered to customers by transporters. Ideally, each customer wishes to receive her orders from the supplier as early as possible. However, since consolidation of shipments results in substantial cost/time saving in transportation and handling, the supplier wishes to consolidate the order delivery as much as possible to minimize the distribution cost. Delivery consolidation implies that some completed orders may have to wait for other orders to be completed so that they can be delivered in the same shipment. Coordinating both the production stage and distribution stage in the supply chain in order to achieve an ideal overall system performance has become more practicable and has received a lot of attention from both industry practitioners and academic researchers [5][6]. The problem we consider in this paper is to find a joint schedule for order production and finished goods delivery so that the makespan of a schedule is minimized. For the convenience of the analysis, we define the makespan of a schedule, denoted by C_{max}, as the time when the delivery truck finishes delivering the last batch to the customer(s) and returns to the production system. Lee and Chen [4] studied this class of scheduling problems by analyzing their complexities. They showed that many of the problems were computationally difficult and they proposed polynomial or pseudo-polynomial algorithms for some of these problems.

This paper examines a simplified version of the problem in which each order is processed by either one of two parallel machines and delivered by a single delivery truck to customers located in close proximity to each other (defined as a one customer area). Wang and Cheng [9] have studied the parallel machine scheduling with batch delivery costs. They showed that the problem of minimizing the sum of the total flow time and delivery cost is NP-complete in the strong sense, and they provided a dynamic programming algorithm to solve the problem. The scheduling problem was extended by Chang and Lee [1] who considered the situation where each job might occupy a different amount of physical space in a delivery truck. They proved that the problem is NP-hard in the strong sense, and they presented a polynomial time heuristic 2 (H2) with a worst case ratio of 2. Chen and Vairaktarakis [2] considered a two-stage scheduling problem in which the first stage is the manufacturing facility and the second stage is the delivery to the customers. There were two machine configurations in the processing facility – single machine and parallel machines. Their objective function was a combination of customer service level and total distribution cost. For each problem they investigated they provided an algorithm, or a proof of intractability accompanied by a heuristic algorithm based on dynamic programming techniques.

Based on the above descriptions, previous related studies focused on developing optimization-based solutions. Although mathematical models have been used to determine coordinated schedules, managers often prefer to use simpler approaches. The H2 proposed by Chang and Lee [1] can produce “good” solutions much faster than the optimization model, but the performance is very sensitive to problem instance variability. As such, there is a need to develop an efficient approach to find an acceptable solution in an

Manuscript received January 11, 2010. This work was supported by National Science Council of Taiwan under grant no. NSC 98-2221-E-020-037.

Cheng-Hsiang Liu is an Assistant Professor of Department of Industrial Management at the National Pingtung University of Science & Technology in Taiwan. He received his M.S. and Ph.D. degrees from the National Chiao Tung University (NCTU) in Taiwan, both in industrial engineering and management. His research areas include data mining, decision support systems, and operation scheduling and production control. (Tel: 886-8-7703202, ext. 7717; E-mail: chliu.uw@gmail.com).

Tzu-Ling Wu and Pei-Shiun Lin are undergraduate students in the Department of Industrial Management at the National Pingtung University of Science & Technology.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

acceptable amount of computation time.

Metaheuristic-based approaches, such as Genetic Algorithms (GAs) and Tabu Search (TS), can be used to improve upon any heuristic's performance when local search is implemented as a post-processing mechanism [7]. In this paper, a regular GA and a hybrid GA (HGA) algorithm are proposed to analyze the hybrid problem of parallel machines scheduling and order delivery. Each order might occupy a different amount of physical space in a transport vehicle. The two metaheuristic approaches are compared with known solutions by H2 in terms of solution quality. The results of the experiments indicate that the metaheuristic approaches can find a solution better than the one obtained by H2 in an acceptable amount of computation time. Among these two proposed approaches, the HGA can find even or better solution in a shorter period of time than the regular GA. Thus, the HGA should be the scheduling approach of choice.

II. Background of the problem

B_k: batch k
C_{tk}: the completion time of B_k on the delivery truck, i.e., the time when the delivery truck finishes delivering B_k to the customer and returns to the factory
$Mt_k^{(1)}$: the completion time of B_k on the machine 1
$Mt_k^{(2)}$: the completion time of B_k on the machine 2
N: number of orders in the model
NB: the number of batch processes
p_i: processing time of order i on the machines
S_i: size of order i ($S_i \leqq Z$)
S_{tk}: the time at which B_k is being delivered by the delivery truck
T: transportation time on the delivery truck
X_{ijk}: 1, if job i is assigned to be processed on machine j and belongs to the kth batch on the delivery truck; 0, otherwise
Z: fixed capacity of the delivery truck

The following Mixed Integer Programming (MIP) model represents the $P2 \rightarrow D|v=1, c=Z|C_{\max}$ problem we studied, in which "P2→D" means that orders are first processed by either one of two parallel machines and then delivered to customers. "v=1, c=Z" means that there is only one delivery truck with capacity Z.

$$Min \quad C_{\max} \tag{1}$$

$$s.t. \quad \sum_{j=1}^{2}\sum_{k=1}^{N_B} X_{ijk} = 1 \quad for \ 1 \le i \le N \tag{2}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{2} S_i X_{ijk} \le Z \quad for \ 1 \le k \le N_B \tag{3}$$

$$Mt_k^{(1)} \ge \sum_{h=1}^{k}\sum_{i=1}^{N} p_i X_{i1h} \quad for\ 1 \le k \le N_B \tag{4}$$

$$Mt_k^{(2)} \ge \sum_{h=1}^{k}\sum_{i=1}^{N} p_i X_{i2h} \quad for\ 1 \le k \le N_B \tag{5}$$

$$St_k = Max\{Mt_k^{(1)}, Mt_k^{(2)}\} \quad for\ 1 \le k \le N_B \tag{6}$$

$$Ct_1 \ge St_1 + T \tag{7}$$

$$Ct_k \ge Ct_{k-1} + T \quad for\ 1 \le k \le N_B \tag{8}$$

$$Ct_k \ge St_{k-1} + T \quad for\ 1 \le k \le N_B \tag{9}$$

$$C_{\max} \ge Ct_k \quad for\ 1 \le k \le N_B \tag{10}$$

$$X_{ijk} \in \{0,1\} \quad for\ 1 \le i \le n, 1 \le j \le 2, 1 \le k \le N_B \tag{11}$$

$$Mt_k^{(1)} \ge 0 \quad for\ 1 \le k \le N_B \tag{12}$$

$$Mt_k^{(2)} \ge 0 \quad for\ 1 \le k \le N_B \tag{13}$$

$$St_k \ge 0 \quad for\ 1 \le k \le N_B \tag{14}$$

$$Ct_k \ge 0 \quad for\ 1 \le k \le N_B \tag{15}$$

$$1 \le i \le n \tag{16}$$

$$1 \le j \le 2 \tag{17}$$

$$1 \le k \le N_B \tag{18}$$

$$1 \le h \le N_B \tag{19}$$

III. Genetic algorithms

Genetic algorithm (GA) is a kind of evolutionary optimization method as proposed by Holland [3]. The principle of GA is based on the natural evolution and has been applied to analyze various types of combinatorial problems. In terms of GA, solutions to the problem are coded in a so-called chromosome structure. Some number of chromosomes is generated to create a solution pool called POPULATION. The FITNESS VALUE of each chromosome is evaluated by computing its objective function value. Each population represents a GENERATION. Once each chromosome's fitness value has been assessed, the best θ% of the population is transferred from the previous generation to the current generation. A total of (100－θ)% new chromosomes must now be generated. The rule of survival-of-the-fittest is implemented to select chromosomes from the previous generation for the crossover operation. In this connection, good chromosomes will be chosen to produce new chromosomes with a probability associated with its fitness value. This process is called CROSSOVER. After a few iterations of crossover operations, the objective of each chromosome in the population often tends to reach some common value. To mitigate this case, these new chromosomes will be subjected to self-tuning called MUTATION with a probability called MUTATION RATE to propagate offspring with more diverse characteristics. The above procedure continues until the terminating criteria are met. Finally, the best solution is obtained at the end of the procedure. Recalled that the number of chromosomes in a pool is referred to as population size (PS) and the total number of pools is regarded as generation number (GEN).

To successfully implement GAs, it is vital to encode solutions in chromosomes. Figure 1 shows the proposed chromosome structure in a 2-dimention matrix. Each gene in the first column, Y_{i1}, denotes the index of the order in position O of the order sequence (O　[1, …, N]); if a order is assigned to machine 1, the gene Y_{i2} is set to zero and is set to one otherwise; if a order is the last order of a batch, then Y_{i3} is set to one and otherwise it is set to zero. Figure 2 shows the crossover and mutation operations of the proposed GA for an example of five orders. From 2a, the genes in the third row have been exchanged between Parent 1 and Parent 2. From 2b, the fourth row of chromosome Offspring 2 has been mutated. The mutation operation is defined as the re-generation of Yij values.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ \vdots & \vdots & \vdots \\ Y_{i1} & Y_{i2} & Y_{i3} \\ \vdots & \vdots & \vdots \\ Y_{N1} & Y_{N2} & Y_{N3} \end{bmatrix}$$

Figure 1. The proposed chromosome structure

$$\begin{array}{ccccc} Parent\ \ 1 & & Offspring\ \ 1 & & Offspring\ \ 1 \\ \begin{bmatrix} 3 & 1 & 1 \\ 5 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 0 & 0 \\ 4 & 1 & 1 \end{bmatrix} & \begin{array}{c} (a) \\ \Rightarrow \end{array} & \begin{bmatrix} 3 & 1 & 1 \\ 1 & 1 & 0 \\ 5 & 1 & 1 \\ 2 & 0 & 0 \\ 4 & 1 & 1 \end{bmatrix} & \begin{array}{c} (b) \\ \Rightarrow \end{array} & \begin{bmatrix} 3 & 1 & 1 \\ 1 & 1 & 0 \\ 5 & 1 & 1 \\ 2 & 0 & 0 \\ 4 & 1 & 1 \end{bmatrix} \\ Parent\ \ 2 & & Offspring\ \ 2 & & Offspring\ \ 2 \\ \begin{bmatrix} 4 & 0 & 0 \\ 1 & 1 & 0 \\ 5 & 1 & 1 \\ 2 & 1 & 1 \\ 3 & 0 & 1 \end{bmatrix} & & \begin{bmatrix} 4 & 0 & 0 \\ 5 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 1 & 1 \\ 3 & 0 & 1 \end{bmatrix} & & \begin{bmatrix} 4 & 0 & 0 \\ 5 & 1 & 0 \\ 1 & 0 & 1 \\ 3 & 0 & 0 \\ 2 & 0 & 1 \end{bmatrix} \end{array}$$

Figure 2. (a) crossover and (b) mutation operation of the proposed GA.

Since the genes used in both crossover and mutation operations are randomly generated, the feasibility of the resulting offspring is not known in advance. It is evident that both crossover and mutation operations do not always produce a feasible solution. Hence, an additional feasibility checking routine is performed after an offspring is generated. This correction mechanism is designed to move orders from the over-capacity batches to other batches with surplus capacity.

IV. Hybrid Genetic Algorithm (HGA)

As scheduling problems with delivery coordination involves three decisions: order sequence, order-to-machine assignment and order-to-batch assignment, the results representation in Figure 1 cannot be simplified until one of the three decisions is known. This paper incorporates a heuristic to assign orders to one of two parallel machines, so as to reduce the results representation in Figure 1 to $N\times 2$ matrix. Assume that order-to-machine assignment is performed in a "black box" that can always find an assignment solution with the minimum completion time of last batch on the machines for any sequence of orders. In this case, feeding the resulting order sequence and order-to-batch assignment produced by crossover and/or mutation operations to the black box would clearly enhance the efficiency of the regular GA approach.

Figure 3 shows the mechanism of the proposed Hybrid Genetic Algorithm (HGA). In HGA, a robust evolutionary algorithm, GA, is proposed to solve SP1: determination of order sequence and order-to-batch assignment; and a parallel scheduling procedure (PSP), which can perform quite well in minimizing the maximum of the sum of processing time on each machine, is proposed to solve SP2: solving the parallel machines scheduling problem. Figure 4 describes the corresponding results representation used in HGA for a five order example. From Figure 4, order 3 is in batch 1; and orders 1 and 5 are grouped into batch 2; and orders 4 and 2 are included in batch 3. All jobs in batch i precede each of those in batch i+1 on the machines. Same crossover and mutation operations described in section 3 are also used in HGA. Again, the correction mechanism used in original GA is also applied to HGA for moving orders from the over-capacity batches to other batches with surplus capacity.

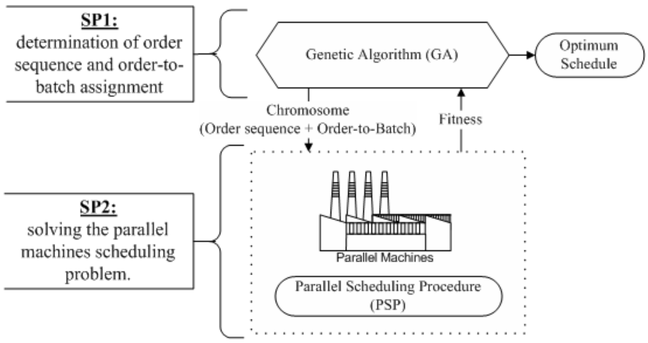

Figure 3. The mechanism of HGA

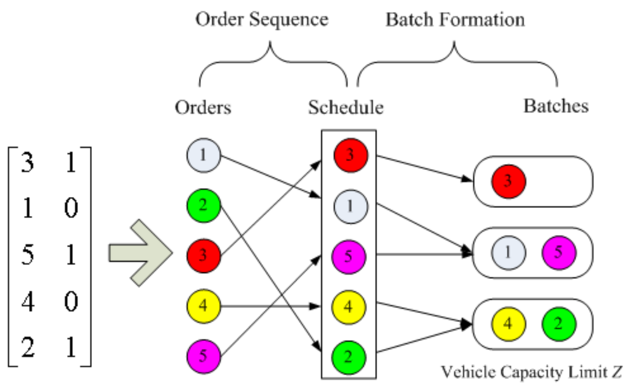

Figure 4. The chromosome structure of the proposed HGA

After the crossover and/or mutation operation, the resulting decisions of order sequence and order-to-batch (chromosome) are fed into the PSP, which can assign orders to one of two parallel machines. The PSP we proposed is based on the work of Sule [8]. Division of orders among two identical machines to achieve minimum completion time of the last batch on the machines is the objective of PSP. The PSP describes the polynomial time heuristic for the parallel machines scheduling problem.

Parallel Scheduling Procedure (PSP)

Step 1: The lower bound of the minimum achievable makespan is given by the sum of processing times divided by the number of available parallel processors.

Step 2: Form a permutation π in which all orders in Set B_i precede each of those in Set B_{i+1}; the orders in each set are sequenced in non-increasing order of p_i.

Step 3: Start allocating the orders to one machine until one of the following happens:

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(a) The sum of processing times of the orders assigned to the machine under consideration becomes equal to the lower bound. If this happens, start assigning orders to the next available machine.
(b) The sum of processing times of the orders allocated to the machine becomes greater than the lower bound. If this happens, the order that has caused the sum to be greater than the lower bound is allocated in the following manner: Sweep across the machines (in order 1 and 2). If the sum of processing times on the next machine is less than the lower bound and allocation of the order will not increase the cumulative processing time on the machine beyond the lower bound, assign the order there. If not, continue the check with the next machine. If, on all machines, the assignment of the present order will increase the sum beyond the lower bound, assign the order to the machine on which such increase would be minimum.

Once all orders are assigned, three decisions have been made completely. The fitness of each individual chromosome (i.e., candidate schedule) is assessed by computing its objective function value (makespan). The resulting fitness are fed back GA for evolving the better solutions.

V. Computational results

Two criteria are used to compare the performances for both the GA and HGA approaches: required convergence generation and resulting objective function value, makespan. Also, the makespan values yielded by GA and HGA were compared with the solution value of algorithm H2 proposed by Chang and Lee [1]. The shape of H2 can be described as follows.

Step 1: Assign orders to batches by algorithm FFD. Let the total number of the resulting batches be b^{H2}.

Step 2: Calculate the sum of the processing times of the orders in B_k and denote it as P_k, for k=1, 2, …, b^{H2}. Re-index these batches so that $P_1 \leq P_2 \leq \ldots \leq P_b^{H2}$.

Step 3: Starting with B_1, assign batches one by one to the machine that has the smaller load before the batch is assigned. Within each batch, orders are sequenced in an arbitrary.

Step 4: Dispatch each finished but undelivered batch whenever the delivery truck becomes available. If multiple batches have been completed when the delivery truck becomes available, then dispatch the batch with the smallest index.

After some preliminary tests, the GA and HGA parameters are set as population size (PS) = 100, crossover rate = 0.9 and mutation rate = 0.1. Both GA and HGA procedures continue until the terminating criterion, which specifies to terminate after 500 consecutive generations without performance improvement, is met.

There are two major experiments (I and II) are conducted in this study. The experiment I is performed to evaluate different scheduling approaches considering the three different levels of T (5, 10, 15) when Z is set to be constant (Z=20). Alternatively, the experiment II is conducted to investigate the effects of the different levels of Z (15, 20, 25) on the performances of the scheduling approaches when T=10. In both experiments, for a given number of orders (N=10, 20, 30), ten problems were randomly generated. In each problem, job sizes and processing times are both uniformly distributed over the integer set [1, 9].

The results of the experiment I are summarized in Table 1. Each item in this table is a makespan average of the ten replications of the experiment. The “Improved %” column in the table indicates the percentage difference between the average objective value obtained by the current approach and algorithm H2. The “Converged Generation” column in the table indicates the average required convergence generation. From Table 1, some important observations can be made: (1) HGA and GA can defeat algorithm H2 for all of the tested conditions; (2) it is noted that HGA can outperform its counterpart, GA, for most of the tested conditions, except for the case T=5 where the transportation is not a scarce resource in the system; (3) the improvements of HGA and GA over algorithm H2 appears in an increasing trend as the transportation time of the delivery truck (T) decreases. (4) it also clearly shows that the HGA is easier to converge and the convergence speed is faster than the regular GA. In summary, HGA can find even or better solution in a shorter period of time than the regular GA.

The results of the experiment II are summarized in Table 2. The same patterns are exhibited by these tables. HGA and GA achieve superior performance improvements over algorithm H2 for all of the tested conditions. HGA performs very well, especially in the case of Z=15. It implies that HGA is suitable for the tested condition where the transportation is a scarce resource in the system. Alternatively, there is no significant difference between HGA and GA when Z=20 and Z=25. We also observe that although both GA and HGA perform comparably as the ratio Z increases, HGA can find even or better solution in a shorter period of time than the regular GA. Thus, the proposed HGA should be the scheduling approach of choice.

VI. Conclusions

In this study, an investigation of the genetic algorithms for scheduling in a production-distribution system with one supplier and one or more customers located in close proximity to each other (defined as a one customer area) with a consideration of different amount of space for storage of orders in the delivery truck has been undertaken. For the problem in which orders are processed by either one of two parallel machines and delivered by a single delivery truck to one customer area. This scheduling problem involves three decisions, order sequence and order-to-machine assignment and order-to-batch assignment. Our goal is to optimize a coordinated schedule that considered makespan.

The regular GA requires a results representation in N×3 matrix. This results representation greatly increases the computing efforts. Thereby, this paper presents a hybrid GA approach, which incorporates a parallel scheduling procedure (PSP) to assign orders to one of two parallel machines, so as to reduce the results representation to N×2 matrix. Using PSP, all orders have been assigned to either one of two parallel machines promptly. The makespan yielded by the

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

HGA and GA was compared to the algorithm H2 at different problem sizes, transportation time on the delivery truck, and the capacity of the delivery truck. From the experimental results, both HGA and GA are superior to the algorithm H2 in terms of the solution quality. The HGA performs better than the GA significantly when the transportation is a scarce resource in the system (i.e., as the T increases and/or Z decreases). Otherwise, there is no significant difference between HGA and GA. Although both GA and HGA perform comparably as the ratio Z decreases and/or T increases, HGA can find even or better solution in a shorter period of time than the regular GA. Thus, the proposed HGA should be the scheduling approach of choice.

Regarding to the future research work, the author would like to suggest the following issue: the first will investigate extending the GA solution approaches to more complicated machine environments. Another future research may consider problems with multiple delivery trucks. Moreover, an obvious area for future research is to solve the scheduling problem of a production-distribution system where more than one customer areas are involved in system. Such a problem would include routing decisions for each shipment. New algorithms/heuristics would be needed to solve such a problem.

Table 1. Makespan average for experiments when Z=20

N	T	Approach	C_{max}	Imporved %	Converged Generation
30	15	H2	126.1	--	
		HGA	118.2	6.26	168
		GA	119.9	4.92	182
	10	H2	100.2	--	
		HGA	89.4	10.78	57
		GA	90.4	9.78	140
	5	H2	87.5	--	
		HGA	77.2	11.77	5
		GA	77.2	11.77	29
20	15	H2	92.0	--	
		HGA	85.0	7.61	93
		GA	85.8	6.74	116
	10	H2	72.4	--	
		HGA	63.4	12.43	63
		GA	63.8	11.88	169
	5	H2	66.7	--	
		HGA	58.7	11.99	4
		GA	58.7	11.99	11
10	15	H2	58.2	--	
		HGA	48.8	16.15	53
		GA	49.4	15.12	84
	10	H2	43.1	--	
		HGA	36.7	14.85	14
		GA	37.3	13.46	34
	5	H2	33.1	--	
		HGA	27.2	17.82	3
		GA	27.3	17.52	62

Table 2. Makespan average for experiments when T=10

N	Z	Approach	C_{max}	Imporved %	Converged Generation
30	15	H2	109.8	--	
		HGA	103.7	5.56	234
		GA	107.7	1.91	231
	20	H2	100.2	--	
		HGA	89.4	10.78	57
		GA	90.4	9.78	140
	25	H2	96.7	--	
		HGA	84.7	12.41	16
		GA	84.8	12.31	111
20	15	H2	74.7	--	
		HGA	70.7	5.35	126
		GA	72.0	3.61	192
	20	H2	72.4	--	
		HGA	63.4	12.43	63
		GA	63.8	11.88	169
	25	H2	68.7	--	
		HGA	59.6	13.25	7
		GA	59.6	13.25	35
10	15	H2	46.4	--	
		HGA	40.5	12.72	27
		GA	41.4	10.78	23
	20	H2	43.1	--	
		HGA	36.7	14.85	14
		GA	37.3	13.46	34
	25	H2	43.1	--	
		HGA	33.7	21.81	6
		GA	34.0	21.11	10

REFERENCES

[1] Y.-C. Chang, and C.-Y. Lee. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*. *158*. pp. 470-487.

[2] Z.-L. Chen, and G. L. Vairaktarakis. (2005). Integrated scheduling of production and distribution operations. *Management Science*. *51(4)*. pp. 614-628.

[3] J. Holland, *Adaptation in natural and artificial systems*. MI: The University of Michigan Press, 1975.

[4] C.-Y. Lee, and Z.-L. Chen, (2001). Machine scheduling with transportation considerations. *Journal of Scheduling*. *4*. pp. 3-24.

[5] K.-P. Li, V. K. Ganesan, and A. I. Sivakumar. (2005). Synchronized scheduling of assembly and multi-destination air transportation in a consumer electronics supply chain. *International Journal of Production Research*. *43(13)*. pp. 2671-2685.

[6] T.-F. Liang. (2008). Integrating production-transportation planning decision with fuzzy multiple goals in supply chains. *International Journal of Production Research*. *46(6)*. pp. 1477-1494.

[7] P. Qu, and S. J. Mason. (2005). Metaheuristic scheduling of 300-mm lots containing multiple orders. *IEEE Transactions on Semiconductor Manufacturing*. *18(4)*. pp. 633-643.

[8] D. R. Sule, *Industrial scheduling*. PWS publishing company, 1996, pp. 81-82.

[9] G. Wang, and T. C. E. Cheng. (2000). Parallel machine scheduling with batch delivery costs. *International Journal of Production Economics*. *68*. pp. 177-183.

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)