A Decomposition Method for Single-Machine Scheduling With Dynamic Arrivals

Pei-Chann Chang[†], *and Shih-Hsin Chen* [††]

Department of Industrial Engineering and Management, Yuan-Ze University, Nei-Li, Tao Yuan, Taiwan, R.O.C., 32026

Abstract

This research develops a new decomposition method for single machine scheduling problem with dynamic arrivals in minimizing the total weighted completion times. We investigated the two decomposition methods proposed by earlier researchers. Different prospects of the approaches are analyzed. We proofed that the new decomposition method is feasible and can find the optimal solution for the original problems by combining the solutions from each subsets. This new method use the WSPT schedule as a decomposition procedure instead of a B&B schedule by other researcher, thus the total computational time is greatly reduced and the algorithm is much easier to be implemented. A branch and bound procedure using the decomposition procedure is developed to further test the efficiency of different decomposition approaches. The experimental results show that the new approach is much more efficient when compared with other decomposition methods.

1. Introduction

This paper considers the single-machine scheduling problem with dynamic arrivals and the objective is to minimize the total weighted completion time. A formal definition of the problem is that there are n jobs to be sequenced in the set of S and each job i has a processing time p_i and an arrival time r_i and the objective is to minimize the total weighted completion time, i.e., $\sum_i w_i c_i$. The problem is very popular in the factory since the orders arrive dynamically in the factory and these orders have to be finished in their earliest completion times. If we take a look one step at a time, the order will pass through the shop floor one after the other. For each machine, the order will arrive at the machine dynamically and the shop floor supervisor wants to complete these jobs at their earliest completion times.

This problem has raised a lot of attentions recently from the researchers and the industrial practitioners. Different lower bounds have been developed by researchers, such as Chandra [1], Bianco and Ricciardelli [2], Dessouky and Deogun [3], Hariri and Potts [4], Chu [5], and Chand et. Al [6]. Also, a series of heuristic approaches, such as Hariri and Potts [4], Liu and MacCarthy [7], Braglia and Melloni [8], Chand et. Al. [9] and Reeves [10] are provided to deal with problems in practical sizes. In addition, a decomposition approach first proposed by Deogun [6], and later modified by Chand et. Al. is very efficient in terms of reducing the computational times. In their approach, they consider the processing time of each job first and then decompose the set of jobs into subsets when the arrival time of the next job is greater than the completion time of the previous job. However, there is a minor mistake in their approach, we give a counter example to demonstrate this flaw and we propose a different decomposition approach in which we consider the arrival time of each job first and then decompose the set of jobs S into subsets S_i . Properties and theorems are provided to prove our approach and experimental results are also shown for comparison between these two approaches.

2. The Old Decomposition Procedure

2.1 **Decomposition by Deogun**

In Deogun's approach, if a sequence S, define a block b

A case example by Deogun can best describe the approach:

i	r_i	p_i	t_i	c_i
1	0	3	0	3
2	2	5	3	8
3	6	7	8	15
4	16	8	16	24
5	17	11	24	35
6	73	13	73	86
7	60	18	86	104
8	40	20	104	124

9	37	29	124	153
10	90	35	153	188
11	190	40	190	230
12	107	43	230	273
13	212	57	273	330
14	337	60	337	397
15	218	68	397	465
16	467	80	467	547
17	470	93	547	640
18	550	93	640	733
19	551	94	733	827
20	586	98	827	925

In this example, he decomposesthe set of jobs S into {1,2,3}, {4,5}, {6,7,8,9,10}, {11,12,13}, {14,15}, {16,17,18,19,20}, i.e., six subsets. The optimal solutions for these six subsets are (1,2,3), (4,5), (8,7,6,9,10), (11,12,13), (15,14), (16,17,18,19,20), and if we combine these subsets into S and we will derive an optimal solution for S, and that is (1, 2, 3, 4, 5, 8, 7, 6, 9, 10 , 11, 12, 13, 15, 14, 16, 17, 18, 19, 20). However, if we consider the following example:

i	r_i	p_i	t_i	c_i
1	0	3	0	3
2	2	5	3	8
3	23	8	16	24
4	17	11	24	35
5	73	13	73	86
6	8	14	86	100

Using Deogun's decomposition approach, the problem can be divided into the following subsets, i.e., {1,2}, {3,4}, {5,6}. Solve the problem for each subsets and recombine the solutions. The final solution from Deogun's decomposition is (1,2,3,4,5,6). But if we take a close look at the problem, we can find out that actually job 6 can be inserted into the idle time between job 2 and job 3. The optimal solution for this problem is (1,2,6,3,4,5) which is different to the solution by Deogun's. Therefore, we can conclude that the decomposition procedure is not correct and leave some rooms to be improved.

Moreover, if we have an early job with large processing time, i.e., 600, since all the jobs are dense (i.e., the arrival time of each job is very close to each other). The Deogun's decomposition can not proceed easily.

2.2 **Decomposition by Chand et. Al.**

Chand et. Al. also propose a different decomposition approach. First they use a Potential blcok finder (partitions jobs into p potential blocks)

Set i = 1 and p = 0.

Use the SPT dispatching rule to find the smallest j $\geq$ i such that jobs $\{i, i+1, ..., j\}$ form a potential block, Set p = p + 1 and then k_p = j.

Stop if $j = n$. Else, set $i = j + 1$ and go to step 2.

and then a block finder procedure is further applied to finds the first m potential blocks from the original *p* blocks.

i Set m=1 and k = k_j

ii Find $\sigma^*(1,k)$,

iii Stop if m=p or if C($\sigma^*(1,k)$) $\leq$ r_{k+1}; jobs {1,…,k} form the first block. Else, set

m = m +1 , k=km, and go to Step 2.

The procedure is based on the arrival time of each jobs and a non-delay schedule. However, in order to find a block there will have m times of branch and bound procedure to be tested in the worst cases. The procedure is very time consuming and not easy to be implemented in the searching procedure.

3. A Decomposition Approach

In this research, we will propose a new decomposition approach which will take the arrival time of each job into consideration first. This new decomposition approach is described in the following:

Theory 1. For a set of jobs ,i.e., N, If ω represents the WSPT schedule and π^* is the optimal schedle for the set of jobs N, then $C(\omega) \geq C(\pi^*)$。

Proof：If ω and π^* are the same then the theory is true.

Otherwise, let k represent the largest integer that satisfy

$\frac{P_{\pi^*(k)}}{\omega_{\pi^*(k)}} > \frac{P_{\pi^*(k+1)}}{\omega_{\pi^*(k+1)}}$, and let $x = \pi^*(k-1)$, $y = \pi^*(k)$, $z = \pi^*(k+1)$. Then, we know that $r_y < r_z$ and $C_x^{\pi^*} < r_z$, that is $r_z > \max\{C_x^{\pi^*}, r_y\}$ （If that is not the case then one can switch y and z to derive a better schedule）.

If ω' means that the set of jobs, i.e., $\{y, \pi^*(k+1), \cdots, \pi^*(n)\}$ in the WSPT sequence. Then, within the ω' , job y must be sequenced within $\pi^*(k+1), \cdots, \pi^*(n)$. If we assume that job y is sequenced after $\pi^*(k+m)$, which means ,

$$\omega' = (\pi^*(k+1), \cdots, \pi^*(k+m), y, \pi^*(k+m+1), \cdots, \pi^*(n))$$

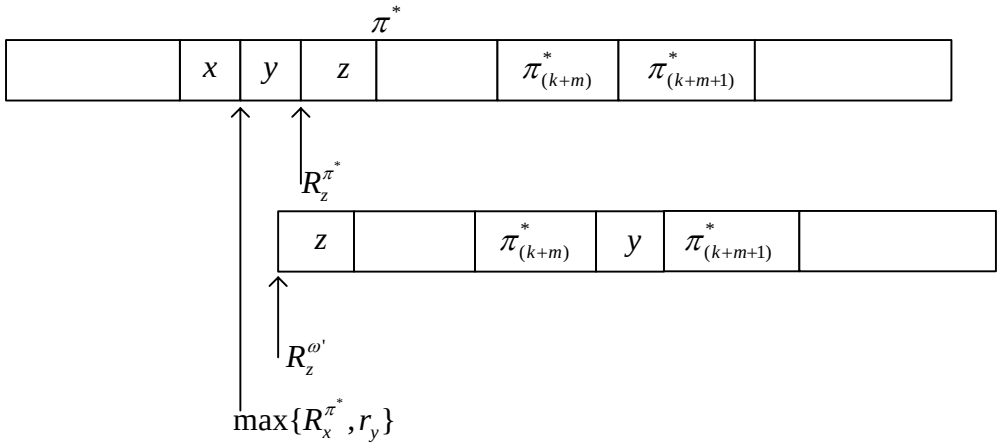

Fig. 1 The figure title

And, $R_z^{\omega'} = r_z$, $R_z^{\pi^*} = \max\{C_y^{\pi^*}, r_z\}$, therefore

$$R_z^{\pi^*} - R_z^{\omega'} \le \max\{C_y^{\pi^*}, r_z\} - r_z$$

$$= \max\{C_y^{\pi^*} - r_z, 0\}$$

If $C_y^{\pi^*} - r_z < 0$, then $R_z^{\pi^*} - R_z^{\omega'} = 0$; Otherwise

$$R_z^{\pi^*} - R_z^{\omega'} = C_y^{\pi^*} - r_z$$

$$< C_y^{\pi^*} - \max\{C_x^{\pi^*}, r_y\} = p_y$$

So, in any cases, $R_z^{\pi^*} - R_z^{\omega'} < p_y$ is true. Therfore, $C_{\pi^*(k+1)}^{\pi^*} - C_{\pi^*(k+1)}^{\omega'} \le p_y$, $C_{\pi^*(k+2)}^{\pi^*} - C_{\pi^*(k+2)}^{\omega'} \le p_y$, $\cdots$, $C_{\pi^*(k+m)}^{\pi^*} - C_{\pi^*(k+m)}^{\omega'} \le p_y$ .That is $C_{\pi^*(k+m)}^{\pi^*} \le p_y + C_{\pi^*(k+m)}^{\omega'}$, and $R_{\pi^*(k+m+1)}^{\omega'} \ge R_{\pi^*(k+m+1)}^{\pi^*}$. We can infer that $C_{\pi^*(n)}^{\pi^*} \ge C_{\pi^*(n)}^{\pi^*}$. Finally, for all jobs in the WSPT sequence , it is very obviously that $C_{\omega(n)}^{\omega} \ge C_{\pi^*(n)}^{\omega'}$, therefore $C_{\omega(n)}^{\omega} \ge C_{\pi^*(n)}^{\pi^*}$, i.e., $C(\omega) \ge C(\pi^*)$.

According to the theory above, we know that the following theory is also true.

Theory 2. For a set of jobs N and its two subsets, i.e., $\{N_1, N_2\}$. If the WSPT sequence of N_1 ,i.e., ω satisfy $C(\omega) \le \min_{i \in N_2}\{r_i\}$, then the optimal schedule of N_1, i.e., π_1^* and the optimal schedule of N_2, i.e., π_2^* are combined, i.e., (π_1^*, π_2^*) must be the optimal schedule for job set N.

Proof：According to theory3, that schedule π_1^* and π_2^* are not overlapped together, therefore schedule (π_1^*, π_2^*) must be a feasible schedule and

$$\varphi((\pi_1^*, \pi_2^*)) = \varphi(\pi_1^*) + \varphi(\pi_2^*)$$

From theory 2, we know that $\varphi(\pi_1^*) + \varphi(\pi_2^*)$ less than the optimal schedule of job set N.

orollary 3. For a set of jobs N and a possible partition, i.e., $\{N_1, N_2, \cdots, N_k\}$, if schedule ω_i and schedule π_i^* represent the WSPT schedule and the optimal schedule for job set N_i and

$$\mathrm{C}(\omega_i) \le \min_{j \in N_{i+1}} \{r_j\} \quad ,$$

$\forall i = 1,2,...,k-1$

then the optimal schedule for N is $(\pi_1^*, \pi_2^*, ..., \pi_k^*)$.

According to the theory above, then the procedure of the partitioning is described as follows:

A set of jobs *N* is divided into k subsets, i.e., {N_1, N_2 ,…, N_k}, according to the following procedure, i.e. DC(*N*):

Let k=1, $N = \{1,2,...,n\}$;

Let $t = \min_{j \in N}(r_j + p_j)$ and $Q = \{j \mid r_j \le t, j \in N\}$;

Reassign $t = \min_{j \in Q} r_j + \sum_{j \in Q} p_j$ then $N_k = \{j \mid r_j \le t, j \in N\}, k = k+1$;

$N = N \setminus N_k$, if $N = \phi$, stop, otherwise go to 2.

Then, we can have a definition of the partition of *N*.

Definition.

For a set *N* and a family of set {N_1, N_2 ,…, N_k}, if

$N_i \cap N_j = \varnothing$, $i \neq j$;

$\bigcup_{i=1}^{k} N_i = N$

then {N_1, N_2 ,…, N_k} is the partition of *N*.

Let us define $Z(\pi) = \sum_{j=1}^{n} w_{\pi(j)} c_{\pi(j)}$,

where $c_{\pi(j)} = \max\{c_{\pi(j-1)}, r_{\pi(j)}\} + p_{\pi(j)}$.

Theorem 1

If {N_1, N_2} is the partition of a given set *N*, then $Z^*(\pi_{N_1}) + Z^*(\pi_{N_2}) \le Z^*(\pi_N)$.

The proof is given in the Appendix.

It is apparent that the following properties are true.

Property 1.

If {N_1, N_2,… N_k } is the partition of a given set *N*, then $\sum_{i=1}^{k} Z^*(\pi_{N_i}) \le Z^*(\pi_N)$.

Property 1 states that the optimal objective function solution of the set of jobs N is the upper bound of its subsets.

According to Property 1, LB_1 can be improved by the following *LB* procedure.

***LB** algorithm*

t = 0, b= 0, *U*=*N*; call DC(*N*) and generate k subsets, i.e., $N_1, N_2, .., N_k$, i=1;

$b = b + LB_1(N_i)$, $i = i+1$;

if *i* > k then exit; otherwise go to 2.

Then, the value of the *LB* algorithm, which serves as a decision index, is adopted to assign the next job in sequence for Heuristic CS2 and it is outlined as follows:

Heuristic CS2

t=0, $S = \phi, U = N, O = 0$;

Find job k such that $BF_k = \min_{j \in U}(LB(U))$;

Schedule job k, i.e., $S = S \cup \{k\}$, $t_k = \max(r_k, t) + p_k, O = O + w_k t_k$, *U*=*U*\{k};

If *U* = ∅ stop and print *O* as a heuristic solution, otherwise go to step 2.

4. The Branch and Bound Procedures

The efficiency of the branch and bound algorithm depends upon the selection of lower bound and dominance properties, which in turn establishes the breadth of the search tree. The following dominance theorem eliminates many unnecessary nodes in a branching tree.

Theorem 2

Two partial sequences σ_1 and σ_2 include the same set of jobs, If

$C(\sigma_1) \le C(\sigma_2)$

$B(\sigma_1) \le B(\sigma_2)$

then σ_1 dominates σ_2; i.e., the sequence initiated with

partial sequence σ_2 will not be included in the optimal sequence.

Denote all the scheduled jobs in an open node as σ_2. If σ_2 can be rescheduled as σ_1 according to some rules or a simple heuristic and satisfies with Theorem 3, then this open node can be eliminated from further consideration. In this study, the HP heuristic will be applied to generate the σ_1. Then, two branch and bound algorithms, *BAB* 1 and *BAB* 2, are developed in this section. They have the same iterative procedures, except that the dominance property is applied in *BAB* 2 procedures.

Let σ represent a partial sequence of a job set *N*; let *heu*(σ) represent the value of a sequence initiated with partial sequence σ to which a heuristic algorithm is applied; and let LB(σ) represents the lower bound value of an open node initiated with σ. The BAB1 algorithm is as follows:

1. $\sigma = ( )$, $OPEN = \{\sigma\}$, $\alpha = heu\ (\sigma)$;
2. if $OPEN = \varnothing$ then exit;
3. take arbitrarily $\sigma \in OPEN$, $OPEN = OPEN - \{\sigma\}$;
4. get a heuristic solution $\alpha' = heu\ (\sigma)$;
5. calculate the lower bound $L = LB(\sigma)$;
6. if $B(\pi) \le L$, then go to 2;
7. if $B(\alpha') = L$ then $\pi = \pi'$, and go to 2
8. if $B(\alpha') < B(\alpha)$ then $\pi = \pi'$;
9. let *U* denote the set containing all the jobs in σ, and $\bar{U} = N - U$;
10. $t = \min_{i \in \bar{K}} \{\max(C(\sigma), r_i) + p_i\}$, $Q = \{i \mid i \in \bar{U}, r_i < t\}$;
11. let $SUCC = \bigcup_{j \in Q} \{(\sigma, j)\}$;
12. set $OPEN \leftarrow OPEN \cup SUCC$;
13. go to 2.

In BAB2, the scheduled jobs represented as σ in each open node is rescheduled according to HP heuristic. If the rescheduled sequence represented as σ_1 is better than σ, then the node can be eliminated from further consideration. The steps from 1 to 9 in BAB2 are the same as those of BAB1 and the rest steps are as follows:

14. apply HP heuristic on *K* to get σ_1;
15. if $B(\sigma_1) < B(\sigma)$ then go to 2;
16. $C = \min_{i \in \bar{K}} \{\max(C(\sigma), r_i) + p_i\}$, $Q = \{i \mid i \in \bar{K}, r_i < C\}$;
17. let $SUCC = \bigcup_{j \in Q} \{(\sigma, j)\}$;
18. set $OPEN \leftarrow OPEN \cup SUCC$;
19. go to 2.

5. Experimental results

All algorithms were coded in C language and run on a Pentium III PC. To evaluate the various schemes, the experiments were conducted on a series of problems reported by Hariri and Potts [5]. The algorithms, including Heuristic HP, Heuristic CS1, Heuristic CS2, BAB1 and BAB2, were tested on problems with dimensions 20,30,40 and 50. For each job *i*, an integer processing time p_i from the uniform distribution [1,100] and an integer weight w_i from the uniform distribution [1,10] were generated. Since the range of release dates is likely to influence the effectiveness of the algorithms, an integer release date for each job *i* was generated from the uniform distribution [0, $50 \cdot 5nR$], where *R* controls the range of the distribution. The value 50.5*n* measures the expected total processing time. For each selected value of *n*, 20 problems were generated for each of the *R* values 0.2, 0.6, 0.8, 1.25, 1.5, and 3.0, thereby producing 120 problems for each value of *n*.

Table 3. Summary results for Different Decompositions

# of jobs n	R value	B&B 1			B&BDC1			B&BDC2			B&BDC3		
		Average Branch nodes	Average CPU sec	# of unsolved nodes	Average Branch nodes	Average CPU sec	# of unsolved nodes	Average Branch nodes	Average CPU sec	# of unsolved nodes	Average Branch nodes	Average CPU sec	# of unsolved nodes
20	0.2	189.80	0.01	0	90.80	0.00	0	189.80	0.02	0	90.80	0.00	0
	0.6	1135.20	0.08	0	447.00	0.03	0	1135.20	0.08	0	447.00	0.03	0
	0.8	1477.00	0.08	0	601.80	0.03	0	1477.00	0.09	0	601.80	0.05	0
	1.25	2096.20	0.13	0	349.40	0.02	0	2096.20	0.12	0	349.40	0.03	0
	1.5	312.20	0.02	0	129.60	0.00	0	312.20	0.01	0	129.60	0.02	0
	3.0	38.20	0.00	0	32.80	0.00	0	38.20	0.00	0	32.80	0.01	0
30	0.2	24126.80	2.99	0	809.40	0.14	0	24126.80	2.95	0	809.40	0.13	0
	0.6	45250.20	4.69	0	1790.60	0.24	0	45250.20	4.66	0	1790.60	0.26	0
	0.8	177365.40	17.83	0	5094.20	0.65	0	177365.40	17.83	0	5094.20	0.64	0
	1.25	20086.40	1.35	0	287.40	0.03	0	20086.40	1.35	0	287.40	0.02	0
	1.5	2359.00	0.18	0	693.80	0.03	0	2359.00	0.16	0	693.80	0.07	0
	3.0	32.20	0.00	0	30.60	0.00	0	32.20	0.00	0	30.60	0.00	0
40	0.2	110400.40	23.21	1	3554.60	0.88	0	110398.60	23.22	1	3554.60	0.86	0
	0.6	612488.40	82.51	3	114616.40	22.73	1	612383.40	82.50	3	115095.00	22.73	1
	0.8	676896.00	89.67	4	376205.80	61.70	3	676909.40	89.68	4	376404.00	61.69	3
	1.25	269434.40	27.50	0	8077.60	1.29	0	269434.40	27.44	0	8077.60	1.26	0
	1.5	75613.80	9.16	0	1796.40	0.22	0	75613.80	9.17	0	1796.40	0.21	0
	3.0	46.00	0.00	0	45.80	0.00	0	46.00	0.00	0	45.80	0.00	0
50	0.2	258942.60	107.10	0	46765.20	13.72	0	258942.60	91.62	0	46765.20	17.30	0
	0.6	2737110.00	755.97	3	943146.80	212.42	0	3642433.00	721.25	3	975529.00	257.51	0
	0.8	4659832.80	1000.00	5	2983272.80	799.98	3	4489304.40	1000.00	5	3119898.60	801.29	3
	1.25	4161644.00	751.15	3	1565724.80	384.96	1	3944916.80	761.23	3	1414276.20	373.87	1
	1.5	1153377.40	116.37	0	60212.20	7.82	0	1153377.40	116.36	0	60212.20	7.83	0
	3.0	384.00	0.03	0	134.20	0.02	0	384.00	0.03	0	134.20	0.02	0

* a lower bound on the average because of some unsolved problems within 1500 seconds.

For both branch and bound methods, the problems with small *R* and large R values are the best, and the most difficult problems occur when *R*=0.6 to *R*=1.25. The average number of branching nodes for B&BCS and B&BHP is below 35% compared to those of B&B1 and B&BBR. In addition, the average computation time for B&BCS and B&BHP is below 45% compared to those of BAB1, thus the effectiveness of Theorem 3 is very clear.

6. Conclusion

In this study, a single machine sequencing problem with different ready times in minimizing the total weighted completion time has been considered. Two heuristics have been developed for a dynamic scheduling environment. Both procedures, Heuristic CS1 and Heuristic CS2, were extensively evaluated and compared with the only existing heuristic, Heuristic HP. Also, an efficient decomposition procedure was presented and included in the heuristic and branch and bound computations. Finally, a dominance property is provided to further improve the efficiency of the branch and bound procedure..

Computational testing has shown that the decision indices are effective and sensitive for both heuristics in which good, in many cases optimal, sequences are generated. Both heuristic outperform the only existing heuristic, Heuristic HP, and Heuristic CS2 outperform Heuristic CS1 in all instances. One explanation of the better performance of Heuristic CS1 and Heuristic CS2 is that they not only include the WSPT measure on which Heuristic 1 is based but also explicitly treat the

remaining unscheduled jobs. Testing has also shown that the performance of the heuristics is not significantly affected by the problem structure.

In contrast to the dominance properties in most branch and bound methods, which eliminate a node with an associated lower bound when it is worse than the upper bound or a branching node with the follow-up job dominated by certain rules. The dominance property in this study eliminated a node in which its partially scheduled sequence was dominated by a sequence obtained by any heuristic or priority rules. Computational results have shown that the dominance property eliminates 65% of the branching nodes and, thus successfully limits the size of the search tree.

References

1. Lenstra, J.K, A. H. G. Rinnooy Kan and P. Brucker, "Complexity of machine scheduling problem," *Annals of Discrete Mathematics*, **1**, 343-362 (1977).

2. Dessouky, M. I. and J.S. Deogun, "Sequencing jobs with unequal ready times to minimize mean flow time," *SIAM Journal of Computing*, **10**, 192-202 (1981).

3. Deogun, J.S., "On scheduling with ready times to minimize mean flow time," *The Computer Journal*, **26,** 320-328 (1983).

4. Bianco, L. and S. Ricciardelli, "Scheduling of a single machine to minimize total weighted completion time subject to release dates," *Naval Research Logistics Quarterly*, **29**, 151-167 (1982).

5. Bianco, L.,S. Ricciardelli, Giovanni Rinaldi, and Antonio Sassano "Scheduling Tasks with Sequence-Dependent Processing Times," *Naval Research Logistics Quarterly*, 35, 177-184 (1988).

6. Chu, C., "A Branch and Bound Algorithm to Minimize Total Flow Time with Unequal Release Dates," *Naval Research Logistics Quarterly*, 39, 859-875 (1992).

7. Chand, S., Rodney Traub, and Reha Uzsoy, "An Iterative Heuristic for the Single Machine Dynamic Total Completion Time Scheduling Problem," *Computers and Operations Research*, 23, 641-651, (1996).

8. Hariri, A. M. A. and C. N. Potts, "An algorithm for single machine sequencing with release times to minimize total weighted completion time," *Discrete Applied Mathematics*, **5**, 99-109 (1983).

9. Posner, M. E., "Minimizing weighted completion times with deadlines," *Operations Research*, **33**, 562-574 (1985).

10. Liu, J. and B. L. MacCarthy, "Effective heuristics for the single machine sequencing problem with ready times," *International Journal of Production Research*, **29**, 1521-1533 (1991).

11. Chand, S., Rodney Traub, and Reha Uzsoy, "Single-Machine Scheduling with Dynamic Arrivals: Decomposition Results and an Improved Algorithm," *Naval Research Logistics Quartly*, 43, 709-719, (1996).

12. B.L. Golden, W.R. Stewart, Empirical analysis of heuristics, The Traveling Salesman Problem, John Wiley and Sons, New York, 1990, pp.207-249.

Dr. P. C. Chang received his M.S. and Ph.D. degrees from the department of Industrial Engineering at Lehigh University in 1985 and 1989. He is a professor of Yuan Ze University in Taiwan. His research interests include Production Scheduling, Sales Forecasting, Case Based Reasoning, ERP, Global Logistics, and Applications of Soft Computing. He has published his research works in several SCI Journals, such as, Decision Support Systems, Expert Systems with Applications, European Journal of Operational Research, International Journal of Production Economics, Applied Soft Computing, Journal of Intelligent Manufacturing, Computers and Operations Research, etc.

Mr. S.H. Chen is a PhD student of the department of Industrial Engineering at Yuan Ze University in Taiwan. His research interests are the Multi-Objective Problems in Production Scheduling, Soft Computing applied in Manufacturing problems and the Development of Genetic Algorithm.