ATLANTIS PRESS
Advances in Economics, Business and Management Research, volume 33
Second International Conference on Economic and Business Management (FEBM 2017)

Exploring a two-criterion order scheduling problem by using five heuristics

Win-Chin Lin[a], Carol Yu[b], Shuenn-Ren Cheng[c], Tzu-Yun Lin[a], Yuan-Po Chao[d], Shang-Chia Liu[e], and Chin-Chia Wu[a]*
[a]Department of Statistics, Feng Chia University,Taichung, 40724, Taiwan
[b]Fundraising office, Fu Jen Catholic University Hospital, New Taipei City, 24205, Taiwan
[c]Graduate Institute of Business Administration, Cheng Shiu University, Kaohsiung, Taiwan
[d]Department of Business Administration, Cheng Shiu University, Kaohsiung, Taiwan
[e]Department of Business Administration, Fu Jen Catholic University,New Taipei City,Taiwan
[*]Corresponding author: Chin-Chia Wu, cchwu@fcu.edu.tw

Abstract
Recently the order scheduling (OS) problem is concerned by the research community. However, the OS study with more than one criterion is only few. In view of this limitation, we address an OS problem in which the objective is to find a schedule to minimize the sum of total flowtime and the maximum tardiness. The complexity of this problem is very difficult. Thus, we use five heuristics including three modified heuristics, an iterated greedy (IG) method, and a particle swarm colony (PSO) algorithm for finding approximately solutions. Finally, the statistical results and comparison performances of all five heuristics are reported.
Key words: *order scheduling; particle swarm optimization; iterated greedy; total flowtime; maximum tardiness*

1 Introduction
In the recent days, the issue of customer order scheduling has grown a hot toptic of research. Some applications of the OS model have existed in the production of integrated circuits and in the manufacturing of semifinished lenses (see Ahmadi *et al.*[1]).
The OS literature on minimizing the total completion time criterion, readers can refer to Leung *et al*.[2,3], Wagneur and Sriskandarajah[4], Wang and Cheng[5], and Sung and Yoon[6]; The OS works on minimizing the total weighted order completion time, readers may refer to focus on discussing the complexity of the problem on two machines by Sung and Yoon[6] and Ahmadi and Bagchi[7], and to on analysing some approximation algorithms and worst bounds by Leung *et al*.[8-11], Wang and Cheng[5], and Chen and Hall[12], etc. As the OS literature involving due dates, we refer readers to Blocher *et al*.[13], Erel and Ghosh[14], Hsu and Liu[15], Lee[16] used a branch-and-bound and several heuristics to solve a mean tardiness OS problem. Leung *et al*.[17], Yang[18], and Xu *et al*.[19] studied a OS model with a learning effect to minimize the total tardiness. They develop a branch-and-bound algorithm, simulated annealing, particle swarm optimization, and order-scheduling MDD algorithms for it. More recently, Framinan and Perez-Gonzalez[20] propose a greedy search algorithm for the customer order scheduling problem to minimize the total completion time. They propose a greedy search algorithm and compared the proposals with existing approximate algorithms. Lin *et al.*[21] considered a two-agent multi-facility order scheduling with ready times. The objective is to minimize the total

completion time of the orders of one agent, with the restriction that the total completion time of the orders of the other agent cannot exceed a given limit.

In light of the above OS literature focuses on single criterion, however, there are common encountered more than two criteria in many real situations. This motivates us to explore minimizing the sum of the total flowtime and maximum tardiness as the objective function. Minimizing the total completion time means that the system can yield an efficient task planning to reduce carrying costs, while, minimizing the maximum tardiness of all jobs means that managers reduce the penalty costs from outside customers. To be best our knowledge, this problem has not been explored until now. The remainder of this study is organized as follows: In Section 2, we define the proposed model. In Section 3, we introduce the details of IG and PSO algorithms and several heuristics. In Section 4, we provide observations of all the proposed algorithms. We draw the conclusions and offer suggestions in the last section.

2. Problem statement

The study can be formally described as follows. Consider *n* orders which are operated on *m* different machines. Those machines are designed in parallel. Pre-emption and machine breakdown are not allowed. Let t_{ik} be the processing time on machine M_k and let d_i be the due date for order *i*. All the *n* orders are ready at time zero. The objective function of this study is to find a schedule to minimize the sum of the total flowtime and maximum tardiness, i.e., minimize $\sum C_i(S) + T_{\max}(S)$. This problem is also NP-hard because the total flowtime minimization OS problem has been shown as NP-hard by Ahmadi *et al.*[1]. Therefore, we propose some heuristics, an IG algorithm, and a PSO for near-optimal solutions.

3. Methods

Due to the fact that this problem is not easily to find the optimal solution in a short **time,** we then propose three heuristics which are modified from Smith[20] and Van Wassenhove and Golders[22]. The main idea of three heuristics is that we replace the processing time by the maximum value, the mean value, and the minimum value among m machines, respectively. Then we apply Van Wassenhove and Golders's algorithm to find all possible feasible solutions, at last, we choose the one with the minimum objective solution among all the feasible solutions. They are recorded as VW-H1, VW-H2, and VW-H3. For more details of Van Wassenhove and Golders's algorithm, readers may refer to Van Wassenhove and Golders [22]. In what follows, we then propose an IG and a PSO to find near-optimal solutions. For the details of IG method, readers may refer to Ruiz and Stützle[24], while for the details of PSO, readers may refer to Shi and Eberhart[25] and Eberhart and Kennedy[26]. According our preliminary instances, the parameters adopted in the IG algorithm, namely the controlling temperature parameter (*T*), the number of neighborhood improvements (*M*), and the number of destructions (*d*), are set to T=0.75, M=600, and d=n/2. Meanwhile, based on the experimental pretests of Xu *et al.*[19], the parameters (w, B1, B2, N, NN, ITRN) = (0.5, 0.5, 0.5, N, NN/2, 600) were used in the PSO method.

4. The tested results

In this section we test and report a computational experiment to evaluate the performances of all proposed five methods over the small and big numbers of orders, respectively. The processing times of orders are generated from a uniform distribution U (1, 100) based on the designed in Leung *et al.*[17], Lee[16] and Xu *et al.*[19], while the due dates of orders are generated from another uniform distribution $TPT \times U(1-\tau-R/2, 1-\tau+R/2)$, where $TPT=\sum_{i=1}^{m}\sum_{j=1}^{n} t_{ij}/m$, τ is noted the tardiness, and R is noted the range of the due dates. The values of (τ, R) are set at (0.25, 0.25), (0.5, 0.25), (0.5, 0.5), (0.5, 0.75), (0.25, 0.5), and (0.25, 0.75). One hundred instances are tested for each case.

In what follows, we will evaluate the performance levels of the branch-and-bound and five heuristics for small number of orders. We examine two order sizes at n = 12 and 14 and three machine sizes at m = 2, 3, and 4. One hundred instances were tested for each case. A total of 3600 instances were tested for small numbers of orders. For each heuristic method, we report the average gap (AGP), where, Vi is the objective function yielded by each heuristic, and V^* is the smallest value of the objective function among the five proposed algorithms. The results are summarized in Tables 1-3.

As shown in Tables 1 and 3, the AGPs of five heuristics increase as the value of m or R increases. On the other hand, Table 2 reports that the AGPs of five heuristics decrease as the value of τ increases.

Table 1 –The gap performance of five methods when m changes (n=12, 14)

n	m	IG	PSO	VW-H1	VW-H2	VW-H3
12	2	430.158	433.063	448.380	445.497	482.285
	3	480.050	483.260	510.327	503.060	552.025
	4	515.533	518.222	551.198	543.567	595.918
14	2	501.358	506.058	522.447	518.072	560.253
	3	550.527	554.758	581.258	575.482	630.257
	4	615.090	619.518	654.092	641.600	718.783

Table 2 –The gap performance of five methods when τ changes (n=12, 14)

N	τ	IG	PSO	VW-H1	VW-H2	VW-H3
12	0.25	519.856	523.209	551.297	544.838	598.812
	0.50	430.639	433.154	455.307	449.911	488.007
14	0.25	608.300	613.099	641.024	632.486	701.367
	0.50	503.017	507.124	530.840	524.283	571.496

Table 3 –The gap performance of five methods when R changes (n=12, 14)

n	R	IG	PSO	VW-H1	VW-H2	VW-H3
12	0.25	432.817	435.345	460.325	455.560	494.698
	0.50	481.127	483.773	510.223	502.233	550.662
	0.75	511.798	515.427	539.357	534.330	584.868
14	0.25	496.822	499.990	524.252	519.253	567.717
	0.50	555.977	560.245	587.543	578.235	634.287
	0.75	614.177	620.100	646.002	637.665	707.290

Next, we examine the performances of all five proposed algorithms for large numbers of orders. We set the order sizes at n = 50 and 100 and the machine sizes at m = 5, 10, and 20. We design the values of (τ, R) at (0.25, 0.25), (0.5, 0.25), (0.5, 0.5), (0.5, 0.75), (0.25, 0.5), and (0.25, 0.75). One hundred instances are examined for each case. Therefore, in total, 3600 instances are tested. The related results are summarized in Tables 4-6.

Tables 4 and 6 release that the AGPs of five heuristics increase as the value of *m* or *R* increases. Meanwhile, the AGPs of five heuristics decrease as the value of τ increases from Table 5.

Table 4 –The gap performance of five methods when m changes (n=50, 100)

n	m	IG	PSO	VW-H1	VW-H2	VW-H3
50	5	2095.873	2191.715	2227.650	2145.530	2302.747
	10	2247.800	2328.432	2426.975	2332.222	2484.847
	20	2374.215	2443.653	2564.323	2483.867	2614.037
100	5	4149.487	4407.415	4342.553	4151.592	4441.107
	10	4391.645	4590.623	4675.648	4468.648	4740.100
	20	4579.588	4747.620	4890.538	4704.155	4939.408

Table 5 –The gap performance of five methods when τ changes (n=50, 100)

n	τ	IG	PSO	VW-H1	VW-H2	VW-H3
50	0.25	2440.314	2530.111	2622.483	2528.909	2689.804
	0.50	2038.278	2112.422	2190.149	2112.170	2244.616
100	0.25	4766.616	4993.281	5053.203	4839.690	5129.959
	0.50	3980.531	4170.491	4219.290	4043.240	4283.784

Table 6 –The gap performance of five methods when R changes (n=50, 100)

n	R	IG	PSO	VW-H1	VW-H2	VW-H3
50	0.25	2039.253	2111.595	2193.762	2111.738	2243.737
	0.50	2275.077	2359.678	2442.798	2357.447	2508.227
	0.75	2403.558	2492.527	2582.388	2492.433	2649.667
100	0.25	3988.890	4176.543	4224.778	4050.163	4289.105
	0.50	4447.842	4660.033	4714.237	4516.082	4786.802
	0.75	4683.988	4909.082	4969.725	4758.150	5044.708

Finally, we investigate the statistical differences in the performance levels of the five algorithms. We apply one-way analysis of variance for small and large orders. The results are provided in Tables 7 and 8. In view of the fact that the *p*-values are less than 0.05, one can refer to the row "Algorithm" under "Source of Variation" in Tables 7 and 8, to confirm that the performance differences of the proposed five algorithms are significant at the 0.05 level, regardless of the order size.

Table 7 –The ANOVA of five methods (n=12, 14)

Source of Variation	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	1626761.48	162676.14	346.49	<.0001
Algorithm	4	127635.46	31908.86	67.96	<.0001
Size	1	316014.82	316014.82	673.10	<.0001
τ	1	487196.43	487196.43	1037.71	<.0001
R	2	315541.99	157770.99	336.05	<.0001
Machine	2	380372.75	190186.37	405.09	<.0001
Error	169	79344.02	469.49		
Corrected Total	179	1706105.51			

Table 8 –The ANOVA of five methods (n=50, 100)

Source of Variation	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	250411836.5	25041183.6	1030.97	<.0001
Algorithm	4	1772423.1	443105.8	18.24	<.0001
Size	1	217222794.9	217222784.9	8943.24	<.0001
τ	1	17294652.2	17294652.2	712.03	<.0001
R	2	9559010.9	4779505.4	196.78	<.0001
Machine	2	2562955.4	2281477.7	93.93	<.0001
Error	169	4104850.7	24289.1		
Corrected Total	179	254516687.2			

To further compare the quality of solutions among the five algorithms, SAS (version 9.4) was used to execute Fisher's least significant difference tests. The results are provided in Tables 9 and 10. As shown in Table 9, the overall averages (for small orders) of AGPs, in decreasing order, are 589.920, 544.617, 537.879, 519.147, and 515.453 for VW-H3, VW-H1, VW-H2, PSO, and IG, respectively. All pairwise comparisons between the five algorithms are significant at the 0.05 level for small orders, except those of the IG vs. PSO and VW-H1 vs. VW-H2. As shown in Table 10, the overall averages (for big orders) of AGPs, in decreasing order, are 3587.04, 3521.28, 3451.58, 3381.00, and 3306.43 for VW-H3, VW-H1, PSO, VW-H2, and IG, respectively. All pairwise comparisons between the five algorithms are significant at the 0.05 level for large orders, except those of the PSO vs. VW-H1, PSO vs. VW-H1 and VW-H1vs. VW-H3.

Table 9 –The Fisher' s LSD of five methods (n=12, 14)

Pairwise Comparison Between Algorithms	Pairwise Mean Difference $\lvert\overline{GAP_i}-\overline{GAP_j}\rvert$	LSD(α=0.05)=10.082 Difference > LSD?
IG vs. PSO	$\lvert 515.453-519.147\rvert$	No
IG vs. VW-H1	$\lvert 515.453-544.617\rvert$	Yes
IG vs. VW-H2	$\lvert 515.453-537.879\rvert$	Yes
IG vs. VW-H3	$\lvert 515.453-589.920\rvert$	Yes
PSO vs. VW-H1	$\lvert 519.147-544.617\rvert$	Yes
PSO vs. VW-H2	$\lvert 519.147-537.879\rvert$	Yes
PSO vs. VW-H3	$\lvert 519.147-589.920\rvert$	Yes
VW-H1 vs. VW-H2	$\lvert 544.617-537.879\rvert$	No
VW-H1 vs. VW-H3	$\lvert 544.617-589.920\rvert$	Yes
VW-H2 vs. VW-H3	$\lvert 537.879-589.920\rvert$	Yes

These results report that the mean AGP for small orders or for large orders of the IG algorithm is the smallest (best), whereas VW-H3 is the largest (worst) for both of small and big orders, at the 0.05 significance level. Additionally, the IG has the least dispersion of the five algorithms, as shown in Figs 1 and 2. These results show that the solutions obtained from the proposed IG have both high accuracy and high stability.

Table 10 –The Fisher' s LSD of five methods (n=50, 100)

Pairwise Comparison Between Algorithms	Pairwise Mean Difference $\lvert\overline{GAP_i} - \overline{GAP_j}\rvert$	LSD(α=0.05)=72.517 Difference > LSD?
IG vs. PSO	$\lvert 3306.43 - 3451.58\rvert$	Yes
IG vs. VW-H1	$\lvert 3306.43 - 3521.28\rvert$	Yes
IG vs. VW-H2	$\lvert 3306.43 - 3381.00\rvert$	Yes
IG vs. VW-H3	$\lvert 3306.43 - 3587.04\rvert$	Yes
PSO vs. VW-H1	$\lvert 3451.58 - 3521.28\rvert$	No
PSO vs. VW-H2	$\lvert 3451.58 - 3381.00\rvert$	No
PSO vs. VW-H3	$\lvert 3451.58 - 3587.04\rvert$	Yes
VW-H1 vs. VW-H2	$\lvert 3521.28 - 3381.00\rvert$	Yes
VW-H1 vs. VW-H3	$\lvert 3521.28 - 3587.04\rvert$	No
VW-H2 vs. VW-H3	$\lvert 3381.00 - 3587.04\rvert$	Yes

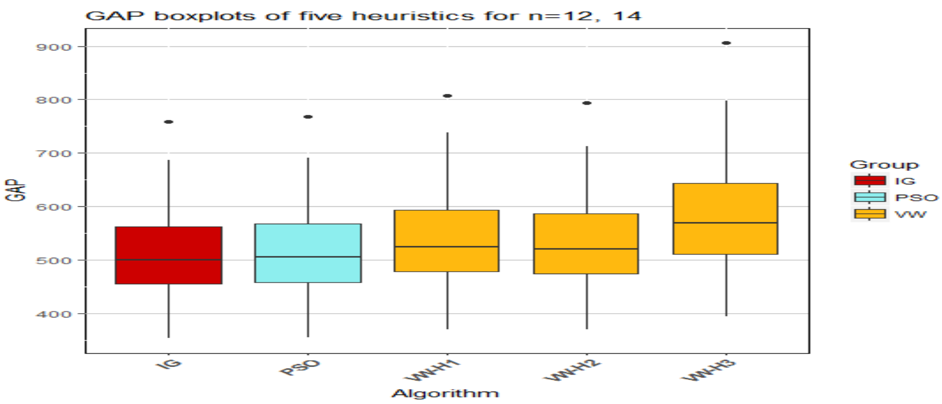

Fig. 1 –The boxplots of five methods (n=12, 14)

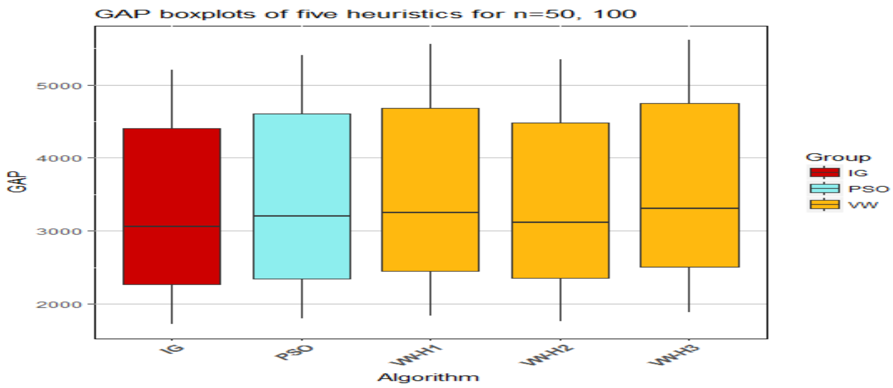

Fig. 2–The boxplots of five methods (n=50, 100)

5. Conclusions and suggestions

The literature releases that OS problems exist in numerous manufacturing and service environments. Production managers commonly should consider optimizing multiple objectives instead of a single objective in many real-life situations. Thus, we address an order scheduling model to minimize the sum of the total flowtime and the maximum tardiness. Different from existed papers assuming that the orders are paid attention on single criterion, this paper addresses the bi-criteria objective function. We then propose five heuristics such as IG, PSO, VW-H1, VW-H2, and VW-H3 methods to find near-optimal solutions. It is noted that IG and PSO are two metaheuristics instead of heuristics. Sometimes, they are not easily to construct. The test results confirm that IG can find a good quality of solutions in terms of their high accuracy and stability for both small and large numbers of orders.

Acknowledgement
This paper was supported by the Ministry of Science Technology (MOST) of Taiwan under grant numbers MOST 105-2221-E-035-053-MY3.

References

1. *R.H. Ahmadi, U. Bagchi, T.A. Roemer*, Coordinated scheduling of customer orders for quick response, Naval Research Logistics **52** (2005) 493-512.
2. *J.Y.T. Leung, H. Li, M. Pinedo,* Order scheduling in an environment with dedicated resources in parallel. Journal of Scheduling **8** (2005) 355-386.
3. *J.Y.T. Leung, H. Li, M. Pinedo,* Approximation algorithms for minimizing total weighted completion time of orders on identical machines in parallel. Naval Research Logistics **53** (2006a) 243-260.
4. *E. Wagneur, C. Sriskandarajah*, Open shops with jobs overlap, European Journal of Operational Research **71** (1993) 366-378.
5. *G. Wang, , T.C.E. Cheng*, Customer order scheduling to minimize total weighted completion time, Proceedings of the 1st Multidisciplinary Conference on Scheduling Theory and Applications (2003) 409-416.
6. *C.S. Sung, S.H. Yoon,* Minimizing total weighted completion time at a pre- assembly stage composed of two feeding machines. International Journal of Production Economics **54** (1998) 247-255.
7. *R.H. Ahmadi, U, Bagchi*, Coordinated scheduling of customer orders, Working paper, John E. Anderson Graduate School of Management, University of California, Los Angeles, 1993.
8. *J.Y.T. Leung, H. Li, M. Pinedo,* Scheduling orders for multiple product types to minimize total weighted completion time, Discrete Applied Mathematics **155** (2007a) 945-970.
9. *J.Y.T. Leung, H. Li, M. Pinedo, J. Zhang,* Minimizing total weighted completion time when scheduling orders in a flexible environment with uniform machines. Information Processing Letters **103** (2007b) 119-129.
10. *J.Y.T. Leung, C.Y. Lee, C.W. Ng, G.H. Young*, Preemptive multiprocessor order scheduling to minimize total weighted flowtime. European Journal of Operational Research **190** (2008a) 40-51.
11. *J.Y.T. Leung, H. Li, M. Pinedo*, Scheduling orders on either dedicated or flexible machines in parallel to minimize total weighted completion time. Annals of Operations Research **159** (2008b) 107-123.
12. *Z.L. Chen, N.G. Hall*, Supply chain scheduling: assembly systems, Working Paper, Department of Systems Engineering, University of Pennsylvania (2001).
13. *J.D. Blocher, D, Chhajed, M, Leung*, Customer order scheduling in a general job shop environment, Decision Sciences **29**(4) (1998) 951-981.

14. *E. Erel, J.B. Ghosh*, Customer order scheduling on a single machine with family setup times: Complexity and algorithms, Applied Mathematics and Computation **185** (2007) 11-18.
15. *S.Y. Hsu, C.H. Liu*, Improving the delivery efficiency of the customer order scheduling problem in a job shop, Computers & Industrial Engineering 57 (2009) 856-866.
16. *I.S. Lee*, Minimizing total tardiness for the order scheduling problem, International Journal of Production Economics **144** (2013) 128-134.
17. *J.Y.T. Leung, H. Li, M. Pinedo*, Scheduling orders for multiple product types with due date related objectives. European Journal of Operational Research **168** (2006b) 370-389.
18. *J. Yang*, The complexity of customer order scheduling problems on parallel machines, Computers & Operations Research **32** (2005) 1921-1939.
19. *J. Xu, C.-C. Wu, , Y. Yin, C.L. Zhao, Y.T. Chiou, W.C. Lin*, An order scheduling problem with position-based learning effect, Computers & Operations Research **74** (2016) 175-186.
20. *J.M. Framinan, P. Perez-Gonzalez*, New approximate algorithms for the customer order scheduling problem with total completion time objective. Computers & Operations Research **78** (2017) 181-192.
21. *W.C. Lin, Y. Yin, S.R. Cheng, T.C.E. Cheng, C.H. Wu, C.-C. Wu*, Particle swarm optimization and opposite-based particle swarm optimization for two-agent multi-facility customer order scheduling with ready times, Applied Soft Computing, **52** (2017) 877-884.
22. *W.E. Smith*, Various optimizers for single state production. Naval Research Logistic Quarterly **3** (1956) 59-66.
23. *L.N. Van Wassenhove, F. Gelders*, Solving a bicriterion scheduling problem. European Journal of Operational Research **4** (1980) 42-48.
24. *R. Ruiz, T. Stützle*, An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. European Journal of Operational Research **187 (3)** (2008) 1143-1159.
25. *Y. Shi, R.C. Eberhart*, A modified particle swarm optimizer. Proceedings of IEEE International Conference on Evolutionary Computation (1998) 69-73.
26. *R.C. Eberhart, J. Kennedy*, A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, (1995) 39-43. Piscataway, NJ: IEEE Service Center.