Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 4 Issue 4, April - 2017

A Scheduling and Rescheduling Algorithm for Integrated Process Planning and Scheduling Problem

A. Reha Botsalı
Department of Industrial Engineering
Necmettin Erbakan University,
Konya, Turkey
rbotsali@konya.edu.tr

Alper Şeker
Department of Industrial Engineering
Yıldırım Beyazıt University
Ankara, Turkey
alper.seker@tubitak.gov.tr

***Abstract*— Decreasing product life cycles and increasing product variety force manufacturing companies to improve the flexibility and efficiency of their production systems. Although the flexibility of production systems is more likely to be an investment issue, the efficiency side is closely related to production planning. In this study, we focus on the integrated problem of product routing (A.K.A process planning) and machine scheduling that is a well-known problem of flexible manufacturing. Studies have shown that analyzing process planning and machine scheduling separately may result in solutions that are not efficient for the integrated system. We propose a genetic algorithm based solution methodology for the integrated problem that reduces the complexity of the solution space by eliminating non-promising solutions at the beginning. Also we test our algorithm on a benchmark data with modified problem characteristics and get promising results.**

Keywords*— *Process Planning; scheduling; genetic algorithms; optimization

I. INTRODUCTION

Process planning (PP) and machine scheduling are critical operations that determine how and when to produce with respect to available resources and constraints. Generally these two activities are considered separately and done by different departments in factory settings in expense of delay and extra used machine hours during production. This results in a strong motivation to decrease the inefficiency of the overall production system by integrating scheduling and process planning operations. In this study, we propose an integrated system that does process planning and scheduling simultaneously.

The integrated problem is a more complex version of classical jobshop scheduling and it is an NP-Complete problem. This problem can be described as follows:

- There is a set of jobs N and each job $i \in N$ has a set of operations O_i to be completed,
- Any operation j ($j \in O_i$) of job i ($i \in N$) can be processed by a machine k of machine set M_{ij} with process time t_{ijk},
- One machine can only process one operation of a job at a time and no preemption is allowed,
- The precedence relationship between the operations of a job should be satisfied.
- The objective is minimizing the completion time of the last operation (makespan) of the schedule.

By introducing the following additional variables and parameters, this problem can be modeled as below:

Q : *Set of all machines*

P_{ij}: *Set of operations that preceeds operation j of job i* $(i \in N, j \in O\_i, P\_ij \subset O\_i)$

$$x_{ijk} = \begin{cases} 1, & \text{if operation j of job i is processed by machine k} \\ & i \in \mathrm{N}, j \in \mathrm{O_i}, k \in \mathrm{M_{ij}}, \mathrm{M_{ij}} \in \mathrm{Q} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ijlm} = \begin{cases} 1, & \text{if operation j of job i starts before operation } m \text{ of job } l \\ & i, l \in \mathrm{N}, \mathrm{j} \in \mathrm{O_i}, \mathrm{m} \in O_l) \\ 0 & \text{otherwise.} \end{cases}$$

S_{ij} : *Start time of operation j of job i* $\left(i \in N, j \in O_i,\ S_{ij} \geq 0\right)$

C_{ij} : *Completion time of operation j of job i* $\left(i \in N, j \in O_i, C_{ij} > 0\right)$

C_{max}: *Makespan* $(C_{max} > 0)$

M : *A very big number*

Objective function: Minimize C_{max} (1)

Subject to:

$$\sum_{k \in M_{ij}} x_{ijk} = 1 \qquad \forall i \in N, \forall j \in O_i \tag{2}$$

$$C_{ij} = S_{ij} + \sum_{k \in M_{ij}} x_{ijk} t_{ijk} \qquad \forall i \in N, \forall j \in O_i \tag{3}$$

$$S_{ij} \geq C_{im} \qquad \forall i \in N, \forall j \in O_i, \forall m \in P_{ij} \tag{4}$$

$$S_{lm} - C_{ij} \geq (y_{ijlm} - 1) \times M \qquad \forall k \in Q, i, l \in \mathrm{N}, \mathrm{j} \in \mathrm{O_i}, \mathrm{m} \in O_l : k \in M_{ij} \cap M_{lm} \tag{5}$$

$$y_{ijlm} + y_{lmij} = 1 \qquad \forall k \in Q, i, l \in \mathrm{N}, \mathrm{j} \in \mathrm{O_i}, \mathrm{m} \in O_l : k \in M_{ij} \cap M_{lm} \tag{6}$$

$$x_{ijk} + x_{lmk} - y_{ijlm} - y_{lmij} \leq 1 \qquad \forall k \in Q, i, l \in \mathrm{N}, \mathrm{j} \in \mathrm{O_i}, \mathrm{m} \in O_l : k \in M_{ij} \cap M_{lm} \tag{7}$$

$$C_{max} \geq C_{ij} \qquad \forall i \in N, \forall j \in O_i \tag{8}$$

In the above model, the objective is minimizing the makespan that equals to the finish time of the operation completed latest as stated by constraint set (8). Constraint set (2) ensures that each operation is assigned to a machine. Constraint set (3) defines that the completion time of an operation is equal to the sum of its start time and processing time. Constraint set (4) states the precedence relationships. Finally constraint sets (5), (6), and (7) guarantee that operations processed by the same machine cannot be processed simultaneously. Clearly, it is not feasible to solve this model in a dynamic floor shop environment under time pressure. For this reason, we propose a genetic algorithm that operates on a limited search space. The organization of the paper is as follows: In the next section, we briefly go over the related literature. In the third section we introduce our genetic algorithm model. In section 4, we present our computational results along with our discussion. Finally, in section 5, we conclude our paper with future research directions.

II. Related Work

Due to the complexity of the integrated process planning problem, the literature mostly focus on heuristic techniques. ([1], [2], [3], [4], [5]).

[6] proposed an integrated process planning and scheduling (IPPS) approach in a batch manufacturing environment. A heuristic approach is created for minimizing time delay and cost of process plan involved in adjustment of process plan. [7] proposed an IPPS approach using ant colony optimization (ACO) algorithm to minimize makespan for fluctuating job shop environment. In that study Processing flexibility of alternative routes and machines is considered. Shrestha et al. [8] developed an IPPS system for HMS (Holonoic Manufacturing Systems) using Dynamic Programming. A GA-based method is used for picking a combination of process plans. [9] proposed a multi-machine setup planning approach using GA to solve IPPS problem. A tool accessibility examination approach was used for adaptive setup planning (ASP), and it was extended to solve multi machines setups planning problem. Authors decided that GA-based ASP is capable to quickly respond in changing shop floor situations. In [10], an IPPS approach is proposed for job shop machining operations via a two-step ASP using GAs. It consists of generic setup planning (step one) and adaptive setup merging (step two) in order to optimize cost, quality, makespan and machines utilization. Authors concluded that proposed approach can generate setup plans adaptively based on machines availability and capability. [11] proposed a GA-based IPPS system in which process route was selected on the basis of balanced level of machines utilization, minimum processing cost and shortest processing time (SPT) dispatching rules. [12] proposed a framework and a combined dynamic rescheduling model for IPPS with three typical types of situations normally encountered in a production system which are new jobs, breakdown of machine, and cancellation of order. Meanwhile, an improved evolutionary algorithm for the IPPS problem to generate an optimal initial scheduling plan is created.. To improve the performance of the algorithm for IPPS new genetic representation for the scheduling plan is developed.

In [13], for each job two alternative process plans were used and in the same direction for each production stage these process plans were used with alternative operations. As a result of this study, they showed that using alternative process plans, operations, and machines had an important positive effect on meeting delivery deadlines in a dynamic production environment. In this study, we also use two alternative process plans for each job to take advantage of this fact.

[14] showed that using genetic algorithms in process planning stage instead of random planning has an effect of 20% decrease in production time. Similarly, [15] showed that in terms of computational time, using genetic algorithm instead of tabu search during scheduling has better results as the problem size gets larger. They find that population size and number of operations were two factors that affect the performance of genetic algorithm.

[16] developed an integrated methodology that uses multiple process plans to overcome the disturbances in production. In their study, using multiple process plans allows the scheduling process to be predictive and reactive to the changes. The study concludes that using more than one process plan on hand increases the flexibility and makes the rescheduling more efficient.

As a result of the literature analysis, we see that previous studies mostly try the integration of process planning and scheduling within the framework of an optimization algorithm. Yet, it is observed that as the solution space and search space expand, the computational time also increases. Considering this, in this study, we propose a hybrid optimization algorithm that is time efficient and also generates effective solutions. Our hybrid algorithm consists of a genetic algorithm module and a simulated annealing algorithm based rescheduling module. Certain studies in the literature oversee the fluctuations in the production environment during manufacturing stage and they neglect the situation where feasible schedules may become infeasible in a dynamic production environment. In our study, we allow two way information flow between scheduling and process planning stages. We use a rescheduling algorithm to response changes in the production environment. This rescheduling algorithm gives us feasible modified schedules ensuring that the original schedule is used for the operations until the time of change in the production environment. By this way, our solution procedure handles abrupt changes in the production environment on a real time basis.

Another point we consider in this study is the assembly production nature of the integrated process planning and scheduling problem. Most studies analyzes this integrated problem in terms of a classical job scheduling perspective. In this study, we assume that as long as there is no precedence relationship between the production operations, these operations can be processed simultaneously on different workstations.

III. ALGORITHM DESIGN

We used a modified version of the genetic algorithm (GA) given in [17]. A feasible schedule is represented by a chromosome as shown in Fig. 1. Each chromosome contains a number of genes that is equal to the number of operations in the respective schedule. Each gene has nine fields and each field carries a predetermined type of information as shown in Fig. 2.

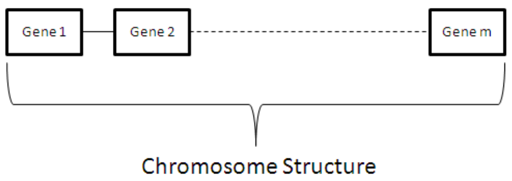

Fig. 1. A chromosome representing a schedule with m operations

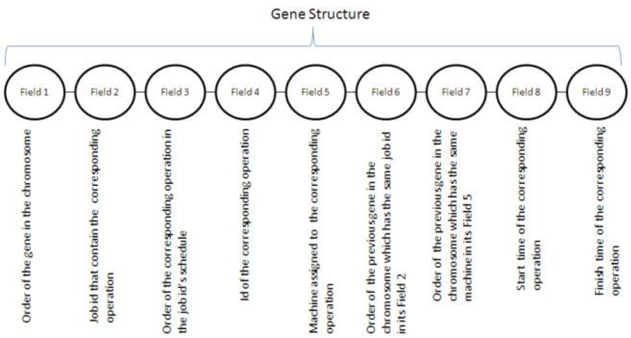

Fig. 2. A gene with its respective fields

For each job, the operations are put in a order depending on the remaining minimum possible process time. For two operations *i* and *j* belonging to the same job, operation *i* comes before operation *j* if the sum of the minimum possible process times of operation *i*'s successor operations are larger than the sum of the minimum possible process times of operation *j*'s successors. Since an operation is more likely to be processed by the machine that has the minimum process time – assuming that the machine is available – minimum possible process times of operations are used in this ordering. By this way, operations of a job are represented by a one dimensional ordered set. In Fig. 2, Field 3 in a gene shows the order number of the operation (Field 4) belonging to the corresponding job (Field 2) in the ordered set.

The start and finish times of operations are stored in Field 8 and Field 9, respectively. Initially all start and finish times are assigned to -1. Starting from the first gene in the chromosome structure, start time and finish time of each operation is calculated using Equations 9 and 10. It should be noted that the selected machine information for each operation is stored in Field 5.

Start Time = Max{0, Finish Time of the Preeceding Operation, Finish Time of the Last Operation Processed by the Selected Machine} (9)

Finish Time = Start Time + Processing Time of the Selected Machine (10)

The fitness value of a feasible schedule is its makespan that is equal to the maximum finish time of its operations.

In GA, the crossover operation between two individual schedules is done on the basis of jobs that have the same route/operation assignment in both schedules. The positions of the genes belonging to such jobs are kept constant in the children, and the remaining genes are swapped between two chromosomes preserving the order they have in the parent chromosomes. Fig. 3 summarizes the crossover operation. Although it is unlikely, if the parents do not have common jobs with same route assignment, then the children are same as parents after crossover operation.

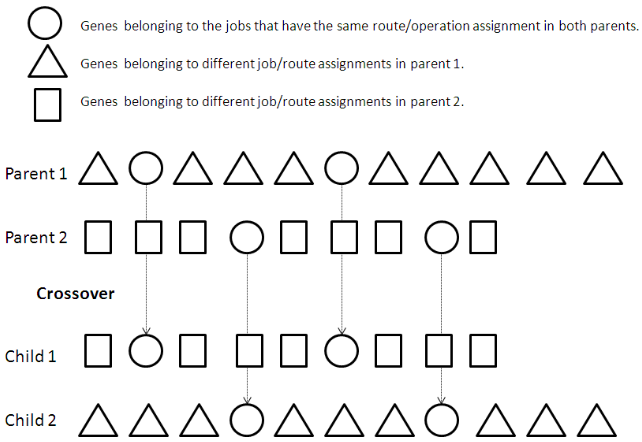

Fig. 3. Crossover operation between two individuals

In production environment, it is likely to have machine breakdowns. These unexpected events require sudden and efficient modifications in production schedule. For example if a machine breaks down at time *t*, then the operation that is processed by this machine at time t and also the other operations that are planned to be processed by the broken machine after time *t* should be rescheduled. Considering this, we also developed a simulated annealing algorithm (SA) that modifies the existing production schedule for a given machine breakdown and respective time. This algorithm uses the same chromosome structure used in GA. In fact, the process of SA is very similar to the mutation operations of GA with the following exceptions.

- Any gene representing an operation whose finish time is less than the break down time *t* cannot be modified.
- The broken machine cannot be assigned to operations whose finish time is greater than the break down time *t*.
- During iterations, any new solution with a shorter makespan is accepted. However, a new solution with a longer makespan is accepted with probability $e^{(makespan\ old - makespan\ new)/T}$, where makespan old and new are the makespan values of old and new solutions, respectively. *T* shows the temperature parameter of SA and it gets smaller as more iterations are done.

IV. COMPUTATIONAL RESULTS AND DISCUSSION

We test our algorithm using the instance set provided by [18]. The genetic algorithm population size and number of iterations is set to 100 and 150, respectively. In order to eliminate non-promising solutions in the search space, we test our algorithm by allowing each job to have only two possible routes available. The available two routes are selected among available routes for each job using two different scenarios. In the first scenario, the two routes that allow the job to be completed at the earliest time are selected assuming that each process of any route is completed by the fastest machine alternative for that process. This scenario is denoted by "Min. Two Routes". In the second scenario, again two routes among all available route alternatives are selected for each job. However during route selection process, the completion time of the route is calculated by using the average processing time of available machines for each process. This assumption takes into consideration the fact that the fastest machine for a process may not be available at all times. The best two routes that are completed in the shortest time using average processing times are taken for the solution algorithm. This scenario is denoted by "Avg. Two. Routes". Other than these two scenarios for route selection process, another alternative is allowing all possible routes in the solution algorithm. This alternative is denoted by "All Routes". As we compare the scenarios that define available routes, we did not notice a significant superiority of one strategy over other in terms of makespan values as shown in Fig. 4. However, in terms of CPU time, including all available routes in the solution process requires more computing time as seen in Fig. 5. For this reason, we decided to continue our analysis with "Avg. Two. Routes" strategy.

[18] suggest a Symbiotic Evolutionary algorithm (SEA) and compare the performance of their algorithm with two other approaches which are a hierarchical approach and a cooperative coevolutionary genetic algorithm (CCGA). They show that SEA outperforms the other two approaches and obtain an improvement rate between 5% and 17% in terms of minimum makespan and mean flow time at the end of 10 different computer runs for each problem instance. Similar to [18], in our algorithm, we consider the precedence constraints and does not allow starting a job's operation before its predecessor operations are completed. However, we assume that two operations of a same job can be processed at the same time if there is no precedence relationship between them. this assumption allows the genetic algorithm proposed in our study to further improve the results found by SEA and provides an improvement rate between 14% and 51% for the same test instances and same number of computer runs in terms of minimum makespan time as shown in Table 1. In Table 2, the comparison of SEA and our genetic algorithm is provided in terms of mean flow times. Again our algorithm provides shorter mean flow times than SEA. However, the improvement rates of the solutions get lower as the instance number (complexity) of the problem increases. This can be explained as follows: The mean flow times provided by our algorithm are the same solutions found by our algorithm with the objective of makespan minimization. On the other hand mean flow time solutions provided by SEA are found with the objective of minimizing mean flow time. For small sized problems there is not much difference in the solutions for minimum makespan and minimum mean flow time. But, as the problem size gets larger, the optimal solutions for minimum makespan and minimum mean flow time may quite differ from each other. CPU time is another dimension for the comparison of algorithms. As given by [18] SEA is run on a IBM Pentium PC (CPU 700 MHz). Our algorithm is run on a computer with Intel Core i3-2100CPU@3.10 Ghz with 2GB Memory. Although, SEA is run on a less powerful computer, its CPU time requirement is less than our algorithm for small sized problems. On the other hand, as problem size increases, the CPU time requirement of our algorithm increases on a lower rate compared to SEA.

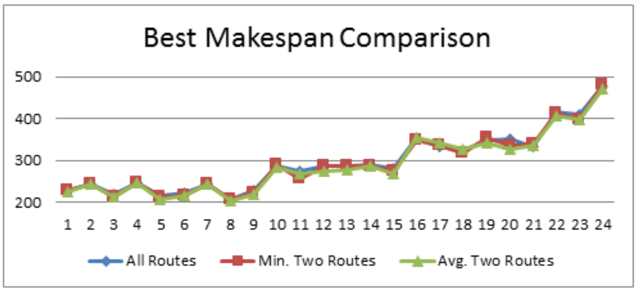

Fig. 4. Best makespan value averages of our genetic algorithm for different route selection strategies and for different instances (1..24)

As changes that affect the current schedule occur during production process, the rescheduling module generates a new schedule in seconds for the remaining operations taking into account the new constraints. In Table 3, a schedule generated for instance 1 of Kim et al.'s [18] data set is shown on the left part. It is assumed that at time 120, machine 10 breaks down. For the new case, the right part of Table 3 shows the new schedule generated by the rescheduling module. As noticed, in the new schedule, only some of the operations that are scheduled after time 120 originally are rescheduled. Due to rescheduled operations, makespan increases from 259 to 279.

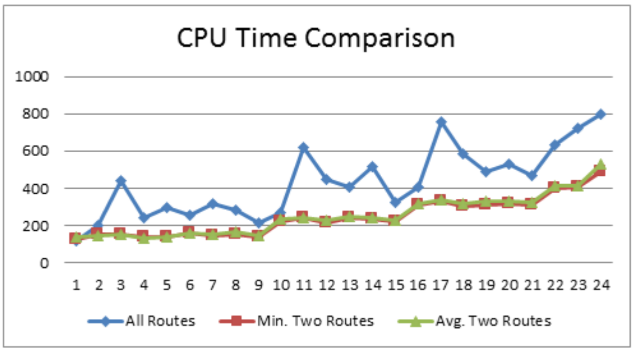

Fig. 5. CPU time averages of different route selection strategies for instances 1to 24

V. Conclusion and Future Research Directions

In this study, we analyze the integrated process planning and scheduling problem, a frequently occurring problem in manufacturing facilities. We developed a genetic algorithm to solve this integrated problem. By allowing two different operations of a job to be processed at the same as long as there is no precedence relationship between each other, we showed that it is possible to improve the existing solutions. Although our assumption relaxes the integrated process routing and scheduling problem studied in the literature a little bit, this assumption also allows us the assembly type production and job scheduling to be combined in production planning. Also we show that by eliminating non-promising solutions during search process, it is possible to greatly shorten computational time without sacrificing solution quality.

For future studies, by using the Multi Agent Systems, process planning and scheduling can be performed in an integrated way within the dynamic production environment including the instant customer demands or changing shop floor conditions (such as machine breakdown). We may also consider some studies such as genetic algorithm or other heuristic and meta-heuristic algorithms are applied to Multi Agent System (MAS) as a supervisor agent which provides additional flexibility and configurability. By this way we may create a dynamic Multi Agent Production System which is designed to produce new schedules rapidly in changing production conditions.

Another future study can be designed based on the RFID-embedded MAS architecture, an agent interaction protocol for dynamic manufacturing control can be proposed to utilize the real-time RFID information, machine and order status to guide agents' cooperative behavior in the manufacturing system.

References

[1] G. J. Palmer, "A simulated annealing approach to integrated production scheduling", Journal of Intelligent Manufacturing, 7(1996): 163–176

[2] A. Bensmaine, M. Dahane, and L. Benyoucef. "A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems."International Journal of Production Research, 52.12 (2014): 3583-3594.

[3] JF. Wang, XL. Fan, CW. Zhang, and ST. Wan. "A Graph-based Ant Colony Optimization Approach for Integrated Process Planning and Scheduling." Chinese Journal of Chemical Engineering, 22.7 (2014): 748-753.

[4] P. Mohapatra, L. Benyoucef, and M. K. Tiwari. "Integration of process planning and scheduling through adaptive setup planning: a multi-objective approach." International Journal of Production Research, 51. 23-24(2013): 7190-7208.

[5] R. K. Phanden, A. Jain, and R. Verma. "An approach for integration of process planning and scheduling." International Journal of Computer Integrated Manufacturing 26.4 (2013): 284-302.

[6] J. Wang, Y.F. Zhang, A.Y.C. Nee, Y.F. Wang, and J.Y.H. Fuh. "Reducing tardy jobs by integrating process planning and scheduling functions", International Journal of Production Research, 47.21(2009): 6069–6084.

[7] C.W. Leung, T.N. Wong, K. L. Mak, and R.Y.K. Fung. "Integrated process planning and scheduling by an agent-based ant colony optimization" Computers & Industrial Engineering, 59.1(2010): 166–180.

[8] R. Shrestha, T. Takemoto, K. Ichinose, and N. Sugimura. "A study on integration of process planning and scheduling system for holonic manufacturing with modification of process plans." International Journal of Manufacturing Technology and Management, 14.3 (2008): 359-378.

[9] N. Cai, W. Lihui, and F. Hsi-Yung. "GA-based adaptive setup planning toward process planning and scheduling integration." International Journal of Production Research 47.10 (2009): 2745-2766.

[10] L.H. Wang, N. Cai, H.Y. Feng, and J. Ma. "ASP: An adaptive setup planning approach for dynamic machine assignments", IEEE Transactions on Automation Science and Engineering, 7.1(2010): 2-14.

[11] W. Zhanjie and T. Ju. "The research about integration of process planning and production scheduling based on genetic algorithm" CSSE, International conference on computer science and software engineering, 1(2008): 9–12.

[12] S. Lv and Q. Lihong. "Process planning and scheduling integration with optimal rescheduling

strategies." International Journal of Computer Integrated Manufacturing, 27.7 (2014): 638-655.

[13] A. Weintraub, D. Cormier, T. Hodgson, R. King, J. Wilson, and A. Zozom. “Scheduling with alternatives: A link between process planning and scheduling”, IIE Transactions, 31.11(1999): 1093-1102

[14] H. Lee and S. Kim, “Integration of process planning and scheduling using simulation based genetic algorithms” International Journal of Advanced Manufacturing Technology, 18.8 (2001): 586–590.

[15] C. Moon,J. Kim, and S. Hur, “Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain”, Computers and Industrial Engineering, 43.1 (2002), 331–349.

[16] C. Grabowik, K. Kalinowski, and Z. Monica. “Integration of the CAD/CAPP/PPC systems”, Journal of Materials Processing Technology, 164–165. 2(2005): 1358–1368.

[17] A. Seker, S. Erol, and R. Botsali. “A neuro-fuzzy model for a new hybrid integrated Process Planning and Scheduling system”. Expert Systems with Applications, 40.13 (2013): 5341-5351.

[18] , Y.K. Kim, K. Park, and J. Ko. “A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling.” Computers & Operations Research, 30.8(2003): 1151–1171.

TABLE I. COMPARISON OF SYMBIOTIC EVOLUTIONARY ALGORITHM AND OUR ALGORITHM IN TERMS OF BEST MAKESPANS

Instance No	Symbiotic Approach				Our Algorithm				Improvement %	
	Best Makespan	Makespan Average of 10 Runs	Standard Deviation	CPU Time	Best Makespan	Makespan Average of 10 Runs	Standard Deviation	CPU Time	Best Mean Flow Time	Mean Flow Time Average of 10 Runs
1	428	437.6	10.9	60.5	225	232.9	7.49	136.44	47.43%	46.78%
2	343	349.7	5.9	68.9	244	245.4	1.80	145.03	28.86%	29.83%
3	347	355.2	7.4	81.7	214	219.6	4.18	151.88	38.33%	38.18%
4	306	306.2	0.4	65.6	247	249.7	1.85	132.81	19.28%	18.45%
5	319	323.7	3.6	63.5	206	219.8	6.00	136.03	35.42%	32.10%
6	438	443.8	5	73.3	215	226.3	5.80	156.66	50.91%	49.01%
7	372	372.4	1.3	69	244	246.5	3.04	151.81	34.41%	33.81%
8	343	348.3	5.7	67.3	202	207.1	4.78	163.39	41.11%	40.54%
9	428	434.9	9.8	73.2	219	229	7.72	145.89	48.83%	47.34%
10	443	456.5	10.8	136	284	292.6	5.80	237.14	35.89%	35.90%
11	369	378.9	5.1	165.8	269	282.4	9.43	243.51	27.10%	25.47%
12	328	332.8	3.4	143.4	275	292.4	9.35	225.25	16.16%	12.14%
13	452	469	10.7	161.2	278	285.9	6.76	245.50	38.50%	39.04%
14	381	402.4	10.6	150.8	286	293.8	5.29	243.85	24.93%	26.99%
15	434	445.2	11	156	267	280.3	8.60	230.58	38.48%	37.04%
16	454	478.8	12	333.6	354	361.7	6.08	315.71	22.03%	24.46%
17	431	448.9	8.7	435.2	342	349.1	4.30	335.21	20.65%	22.23%
18	379	389.6	7.5	357	326	335.5	8.71	319.91	13.98%	13.89%
19	490	508.1	10	417.8	342	356.1	8.79	332.84	30.20%	29.92%
20	447	453.8	5.2	384	328	354.1	11.71	328.31	26.62%	21.97%
21	477	483.2	6.8	392.4	336	348.2	9.16	327.19	29.56%	27.94%
22	534	548.3	6.9	1033.3	408	424.4	10.97	413.78	23.60%	22.60%

23	498	507.5	8.3	1016.6	398	413.3	11.46	409.90	20.08%	18.56%
24	587	602.2	7.1	1622.7	471	490.9	11.64	526.71	19.76%	18.48%

TABLE II. COMPARISON OF SYMBIOTIC EVOLUTIONARY ALGORITHM AND OUR ALGORITHM IN TERMS OF BEST MEAN FLOW TIMES

Instance No	Symbiotic Approach				Our Algorithm				Improvement %	
	Best Mean Flow Time	Mean Flow Time Average of 10 Runs	Standard Deviation	CPU Time	Best Mean Flow Time	Mean Flow Time Average of 10 Runs	Standard Deviation	CPU Time	Best Mean Flow Time	Mean Flow Time Average of 10 Runs
1	313.3	318.9	3.7	57.5	189.7	206.4	10.98	136.44	39.46%	35.29%
2	281.2	287.7	4.7	68	187.7	205.7	11.48	145.03	33.26%	28.49%
3	295.8	304.8	4.3	81.3	192.0	203.4	7.44	151.88	35.09%	33.26%
4	247.2	251.3	4.8	65.7	217.3	224.6	5.01	132.81	12.08%	10.64%
5	275.7	280.3	3.2	66.8	188.8	197.8	4.83	136.03	31.51%	29.43%
6	374.2	384.7	5.7	75.4	177.0	203.8	10.64	156.66	52.70%	47.02%
7	310.5	314.1	2.6	68.1	207.0	218.1	6.66	151.81	33.33%	30.58%
8	288.5	295.2	5	67	183.7	191.8	6.62	163.39	36.34%	35.04%
9	292.8	298.9	7	71.8	192.8	209.9	8.57	145.89	34.14%	29.77%
10	338.9	349.2	6.1	133.8	254.3	265.5	8.56	237.14	24.95%	23.98%
11	303.4	312.9	7.6	163.3	234.4	255.1	14.17	243.51	22.73%	18.46%
12	271.7	279.6	4.7	150.5	239.7	257.4	6.76	225.25	11.79%	7.92%
13	375.9	387	7.1	156	241.7	258.0	7.20	245.50	35.71%	33.32%
14	330	346.9	8.5	149.5	248.1	262.1	7.52	243.85	24.81%	24.45%
15	305.1	316.1	6.2	155.9	239.3	248.7	7.75	230.58	21.56%	21.33%
16	352.4	359.7	4.3	339.1	300.8	319.2	11.79	315.71	14.66%	11.27%
17	359	364.7	4.7	438.7	290.8	311.9	8.71	335.21	19.01%	14.47%
18	313.5	322.5	6.4	355.6	279.8	299.7	13.68	319.91	10.77%	7.06%
19	400.4	406.4	4.6	416.6	305.4	318.9	10.73	332.84	23.72%	21.53%
20	361.3	372	5.7	337.7	292.1	310.6	8.54	328.31	19.16%	16.51%
21	350.9	365.4	8.2	364.7	298.5	312.4	10.84	327.19	14.93%	14.51%
22	411.5	417.8	5.8	1007.6	359.6	383.0	13.65	413.78	12.61%	8.32%
23	396.3	404.7	5.1	999.3	345.3	370.9	11.94	409.90	12.86%	8.35%
24	435.9	452.9	7.5	1597.2	416.5	443.1	15.89	526.71	4.45%	2.17%

TABLE III. ORIGINAL SCHEDULE AND REGENERATED SCHEDULE FOR INSTANCE 1 OF KIM ET AL. * DENOTES RESCHEDULED OPERATIONS IN THE NEW SCHEDULE.

Order	Job #	Operation #	Machine #	Start Time	End time	Job #	Operation #	Machine #	Start Time	End Time	
0	6	1	11	0	29	6	1	11	0	29	
1	3	7	1	0	50	3	7	1	0	50	
2	5	1	6	0	30	5	1	6	0	30	
3	4	3	13	0	24	4	3	13	0	24	
4	6	8	12	0	20	6	8	12	0	20	
5	5	6	2	0	28	5	6	2	0	28	
6	2	1	5	0	10	2	1	5	0	10	
7	1	1	14	0	10	1	1	14	0	10	
8	2	2	15	10	17	2	2	15	10	17	
9	6	9	14	20	27	6	9	14	20	27	
10	4	4	10	24	46	4	4	10	24	46	
11	5	8	2	28	34	5	8	2	28	34	
12	5	7	14	28	70	5	7	14	28	70	
13	6	2	8	29	75	6	2	8	29	75	
14	1	2	11	29	53	1	2	11	29	53	
15	6	5	15	29	53	6	5	15	29	53	
16	2	7	6	30	55	2	7	6	30	55	
17	6	13	2	34	60	6	13	2	34	60	
18	3	16	10	46	95	3	16	10	46	95	
19	4	5	4	46	83	4	5	4	46	83	
20	3	8	1	50	56	3	8	1	50	56	
21	3	1	11	53	87	3	1	11	53	87	
22	1	3	15	53	96	1	3	15	53	96	
23	3	9	13	56	95	3	9	13	56	95	
24	5	2	8	75	115	5	2	8	75	115	
25	4	6	9	83	95	4	6	9	83	95	
26	6	10	4	83	101	6	10	4	83	101	
27	6	3	11	87	98	6	3	11	87	98	
28	3	10	7	95	136	3	10	7	95	136	
29	2	8	10	95	136	4	9	2	95	118	
30	4	9	2	95	118	2	4	9	95	105	
31	2	4	9	95	105	1	5	13	95	125	
32	1	5	13	95	125	3	13	15	96	131	
33	3	13	15	96	131	1	4	12	96	139	
34	1	4	12	96	139	2	3	4	101	141	
35	2	3	4	101	141	2	5	1	105	138	
36	2	5	1	105	138	5	3	3	115	126	
37	5	3	3	115	126	6	6	2	118	160	
38	6	6	2	118	160	3	17	13	125	164	*
39	3	2	15	131	164	5	4	8	126	155	*
40	2	12	8	136	162	2	8	15	131	175	*
41	3	17	10	136	156	6	11	7	136	175	
42	6	11	7	136	175	6	14	1	138	143	*
43	6	4	12	139	180	4	10	12	139	184	*
44	1	6	4	141	173	1	6	4	141	173	
45	2	6	5	141	179	6	15	9	143	182	*
46	3	18	13	156	189	4	1	1	143	177	*
47	3	11	2	160	189	3	2	2	160	195	*
48	5	4	8	162	191	3	18	13	164	197	*
49	3	3	1	164	175	1	7	5	173	222	*
50	3	14	4	173	185	2	12	4	175	200	*
51	1	7	11	173	212	4	2	6	177	204	*
52	2	13	7	175	214	6	18	9	182	204	*
53	6	12	6	175	188	6	4	12	184	225	*
54	5	9	5	179	219	4	11	14	184	201	*
55	4	10	12	180	225	3	11	2	195	224	*
56	3	15	14	185	201	5	9	13	197	233	*
57	4	1	4	185	218	2	13	7	200	239	*
58	6	14	13	189	198	3	14	6	204	218	*
59	3	5	2	189	208	3	15	14	218	234	*
60	5	5	6	191	220	5	5	6	218	247	*
61	6	15	9	198	237	2	6	5	222	260	*
62	3	19	14	201	235	1	8	8	222	269	*
63	3	12	1	201	220	3	5	2	224	243	*
64	1	8	8	212	259	6	7	11	225	240	*
65	6	7	11	212	227	3	19	14	234	268	*
66	2	14	10	214	251	3	12	1	234	253	*
67	4	2	15	218	238	3	3	7	239	247	*
68	3	6	12	225	234	2	14	12	239	279	*
69	3	4	3	234	254	3	6	7	247	255	*
70	4	11	14	235	252	6	12	6	247	260	*
71	6	18	13	237	256	3	4	2	255	273	*