International Journal of
Intelligent Systems and
Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Comparison of Simulated Annealing and Genetic Algorithm Approaches on Integrated Process Routing and Scheduling Problem

Ahmet Reha Botsalı*[1],

Accepted 3rd September 2016

Abstract: Today flexible manufacturing systems are highly popular due to their capability of quick response to customer needs. Although the advantages of flexible manufacturing systems cannot be denied, these systems also bring new issues on production planning side. Especially assigning machines to production operations and scheduling these operations with respect to machine constraints turn out to be an NP-Hard problem. In this study, the integrated process routing and scheduling problem is explained, and the performance of two different meta-heuristic techniques, which are genetic algorithms and simulated annealing, are compared in terms of solution time and quality.

Keywords: *Optimization, Integrated Process Planning and Scheduling, Job Scheduling, Simulated Annealing, Genetic Algorithms.*

1. Introduction

Today, as the flexible manufacturing systems get more popular, scheduling production activities becomes more complex. The production managers have to make decisions such as: "Identifying the production operations that the jobs should go through", "Assigning machines to job operations", and "Identifying the sequence job operations on the machines". Some of these decisions are related to the routing of the jobs on the shop floor and some of them are related to machine scheduling. These multiple aspects of production planning make the decision making much harder. Although it is possible make decisions sequentially, this approach leads to a sub optimal solution. In order to find the best solution that minimizes the completion time of all jobs (makespan), the solution algorithm should take into account all aspects of the problem. This integrated production planning problem is referred as *integrated process routing and scheduling problem* in the literature.

It is possible to solve *integrated process routing and scheduling problem* optimally using the following mathematical programming model given by Botsalı and Şeker [1]. The model is 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
$(i \in N, j \in O_i, P_{ij} \subset O_i)$

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

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

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

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

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

M : *A very big number*

Objective function:

(1) *Minimize* C_{max}

Subject to:

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

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

(4) $S_{ij} \geq C_{im} \quad \forall i \in N, \forall j \in O_i, \forall m \in P_{ij}$

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

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

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

(8) $C_{max} \geq C_{ij} \quad \forall i \in N, \forall j \in O_i$

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.

This problem is NP-Hard and it becomes more and more difficult to solve the problem at the optimal level as the problem size gets larger. Due to this fact, in the literature there are various studies ([2], [3], [4], [5], [6], [7], [8]) to find a good solution in a short time

[1] *Necmettin Erbakan University, Turkey*
** Corresponding Author: Email: rbotsali@konya.edu.tr*
Note: This paper has been presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.

for this problem using heuristic methods. In this study, we compare the performance of two different heuristics based on genetic algorithms and simulated annealing algorithm using the data instances provided by [2]. In the next section, details of these heuristics are provided.

2. Heuristic Algorithms

The genetic algorithm used in this study was developed by Botsalı and Şeker [1]. Basically, this algorithm assigns a gene for each of the job operations that will be processed on the job floor. If there are n jobs and each job i has o_i operations to be completed on a production scenario, then the total number of genes that is carried by the chromosome of the respective individual in the population is equal to $\sum_{i=1}^{n} o_i$.

Each gene in the chromosome has nine fields and each of these fields carries specific information as given below:

- Field 1: Order of the gene in the chromosome,
- Field 2: Job id that contain the corresponding operation
- Field 3: Order of the corresponding operation in the job id's schedule
- Field 4: Id of the corresponding operation
- Field 5: Machine id assigned to the corresponding operation
- Field 6: Order of the previous gene in the chromosome which has the same job id in its Field 2
- Field 7: Order of the previous gene in the chromosome which has the same machine id in its Field 5
- Field 8: Start time of the corresponding operation
- Field 9: End time of the corresponding operation

If two individuals in the population have some jobs in common that share the same assigned operation sequence then these individuals can be crossed over. During crossover, the genes of these individuals are exchanged while keeping the location of the genes that carry the information belonging to common jobs. Other than the crossover operation, there are two types of mutation operation which are:

1. Changing the machine of a specific operation with an alternative machine (Change of field 5 in the gene)
2. Changing the order of genes in the chromosome.

The simulated annealing algorithm uses a data structure that is similar to the one used in the genetic algorithms. Also the modification of the solution during the simulated annealing algorithm's iterations are similar to the mutation operations of the genetic algorithm. During iterations, any new solution with a shorter makespan is accepted. However, a new solution with a longer makespan is accepted with probability:

$$e^{((\text{makespan old-makespan new})/T),}$$

where makespan old and new are the makespan values of old and new solutions, respectively. T shows the temperature parameter of simulated annealing algorithm and it gets smaller as more iterations are done.

The algorithms are tested on a data set provided by Kim et al. [2]. However, there is an assumption we made which is different than the work done by Kim et al. [2]. We assume that two or more operations belonging to a specific job can be processed at the same time by different machines as long as there is no precedence constraint between these operations. By this way, integrated process routing and scheduling problem analysed in this study contains some characteristics of the assembly line scheduling problem. Yet, it is still possible to cancel this assumption and apply our algorithms to the original problem by just modifying the fitness/objective value calculation functions in the coding. Our computational results are given in the next section.

3. Computational Results

The genetic algorithm population size and the number of iterations is set to 100 and 150, respectively. Tournament size is taken as 3 and at each generation of the genetic algorithm 20% of the population is replaced by new individuals. For the simulated annealing algorithm, the maximum and minimum temperatures are set to 300 and 0.001, respectively. The cooling rate is set to 0.9 and at each temperature 10 iterations are done.

Both algorithms are run on a computer with Intel Core i5-5200U CPU@2.20 Ghz with 8GB Memory. For each instance, the algorithms are run 10 times and 10 different solutions are obtained. In Table I, the best solution values and the average of all solution values are provided for each algorithm and instance scenario.

As seen in Table I, the simulated annealing algorithm outperforms the genetic algorithm both in terms of solution quality and solution time when best solutions are analysed. Out of 24 instances, only for instance 11, the best solution of the simulated annealing algorithm is %0.1 higher than the genetic algorithm solution. In two instances, best solutions of the simulated annealing and the genetic algorithm are same and for the rest of the instance runs, the simulated annealing algorithm solutions are better than genetic algorithm solutions.

When the algorithms are compared in terms of average solution quality, except three instances, again solutions of the simulated annealing algorithm are better than solutions of the genetic algorithm. For those three instances, the average of simulated annealing solutions are at most 1.8% larger than the average of genetic algorithm solutions. However, for the remaining instances, the average of the solutions given by the simulated annealing algorithm can be up to 7.1% better than the average of the solutions given by the genetic algorithm.

Finally when two algorithms are compared in terms of solution time, the simulated annealing algorithm's solution time is in general 90% shorter than the solution time of the genetic algorithm. In fact this is something expected since the number of computations is higher in the genetic algorithm.

During genetic algorithm computations, it is observed that the population converges to final solution before the 150th generation, so it may be possible to shorten the solution time of the genetic algorithm a little bit by reducing the number of generation parameter value from 150 to a lower value. Yet, still it is not possible for the genetic algorithm to be as effective as the simulated annealing algorithm in terms of computation time.

Although both algorithms used in this study are heuristic algorithms and they do not guarantee the optimal solution in the end, it is interesting to see that the simulated annealing algorithm gives better results compared to the genetic algorithm in general. There may be several reasons behind this. The author thinks that one of these reasons may be the size of the solution space. Since the number of the variables is very large in an integrated process routing and scheduling problem, it is difficult for a heuristic algorithm to explore the whole space. For this reason, the quality of the final solutions highly depend on the initial solution(s) that the heuristic algorithms start from. As stated in the previous section, the genetic algorithm starts with a set of solutions which contain 100 different solutions (initial population). These 100 solutions are at different quality levels. The tournament selection procedure of the genetic algorithm gives higher chance to good quality solutions to be involved in crossover process. However, it seems like this is not enough to explore the parts of the solution space containing good quality solutions. On the other hand, the simulated annealing algorithm, starts with the best solution chosen out of an initial solution set of 100 solutions. It is seen that searching the solution space starting with a good solution and putting all the computational effort on one promising solution lead the simulated annealing algorithm to be superior to the genetic algorithm.

Table 1. Comparison of makespan solutions

	Simulated Annealing Algorithm		Genetic Algorithm		% Improvement in Genetic Algorithm	
Instance No	Best Makespan	Avg. Makespan	Best Makespan	Avg. Makespan	% Difference in Best Makespan Value	% Difference in Average Makespan Value
1	209	224,9	228	239,4	8,3%	6,1%
2	244	250,2	244	245,8	0,0%	-1,8%
3	207	218,8	213	223,4	2,8%	2,1%
4	247	256,2	247	253,8	0,0%	-0,9%
5	208	215,5	214	219,7	2,8%	1,9%
6	200	216	216	228,5	7,4%	5,5%
7	244	249	244	246,2	0,0%	-1,1%
8	204	209,9	207	215,3	1,4%	2,5%
9	209	218,8	222	229	5,9%	4,5%
10	267	284,7	291	300,2	8,2%	5,2%
11	259	273,6	257	273,9	-0,8%	0,1%
12	273	285	287	298,3	4,9%	4,5%
13	264	276,1	285	294	7,4%	6,1%
14	267	290	285	299,4	6,3%	3,1%
15	252	266,3	273	281,9	7,7%	5,5%
16	332	349	347	360,2	4,3%	3,1%
17	322	334,9	327	344,1	1,5%	2,7%
18	298	318,6	325	335,8	8,3%	5,1%
19	340	347,7	348	364,8	2,3%	4,7%
20	325	340,2	336	356	3,3%	4,4%
21	315	330,9	339	350,8	7,1%	5,7%
22	379	402,4	410	433,2	7,6%	7,1%
23	376	392,1	398	409,9	5,5%	4,3%
24	450	472,9	472	487,1	4,7%	2,9%

Table 2. Comparison of computation time

Instance No	SA Average Time (Seconds)	GA Average Time (Seconds)	% Improvement in Genetic Algorithm Solution Time
1	10,02	92,37	89,2%
2	10,40	102,12	89,8%
3	12,05	107,74	88,8%
4	10,18	92,98	89,0%
5	10,42	96,09	89,2%
6	11,08	108,12	89,8%
7	10,80	101,79	89,4%
8	11,53	101,76	88,7%
9	10,24	94,26	89,1%
10	11,64	147,94	92,1%
11	14,97	159,36	90,6%
12	11,75	143,21	91,8%
13	12,09	159,58	92,4%
14	13,24	158,37	91,6%
15	12,33	146,56	91,6%
16	14,04	202,80	93,1%
17	16,50	215,61	92,3%
18	14,12	202,91	93,0%
19	15,91	208,58	92,4%
20	15,86	209,66	92,4%
21	15,67	203,74	92,3%
22	18,44	262,17	93,0%
23	19,96	263,44	92,4%
24	21,19	314,25	93,3%

4. Conclusion

In this study, two different algorithms are compared for the integrated process routing and scheduling problem. These two algorithms are based on genetic algorithm and simulated annealing algorithm techniques. The algorithms are tested over the problem instances provided by Kim et. al. [2]. It is observed that the simulated annealing algorithm has better performance than the genetic algorithm in general.

It is possible to extend this study in various directions. Different assumptions such as the allowance of pre-emption on shop floor can be considered. Also the performance of different heuristic algorithms such as particle swarm optimization or tabu search can be compared.

References

[1] A. R. Botsalı And A. Şeker, "A Scheduling and Rescheduling Algorithm for Integrated Process Planning and Scheduling Problem", Necmettin Erbakan University, Konya, Turkey, Working paper, 2016.

[2] 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, vol. 30, pp 1151–1171, 2003.

[3] H. Lee and S. Kim, "Integration of process planning and scheduling using simulation based genetic algorithms", International Journal of Advanced Manufacturing Technology, vol. 18, pp 586–590, 2001.

[4] 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, vol. 59, pp 166–180, 2010.
[5] S. Lv and Q. Lihong "Process planning and scheduling integration with optimal rescheduling strategies", International Journal of Computer Integrated Manufacturing, vol. 27, pp 638–655, 2014.
[6] 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, vol. 51, pp 7190–7208, 2013.
[7] 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, vol. 43, pp 331–349, 2002.
[8] 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, vol. 40. pp 5341-5351, 2013.

