

A New Priority-Sort Based Optimization Algorithm for Integrated Process Planning and Scheduling

Ghiath Al Aqel, Muhammad F. Ausaf, Xinyu Li, and Liang Gao

Abstract—Modern manufacturing systems emphasize the need to improve the overall efficiency of the system and achieve global optimality rather than striving for excellence in isolated individual components. Integration of process planning and scheduling, which were previously treated as spate entities, has become an important area of research for accomplishing this goal. This paper presents a novel optimization algorithm for integrated process planning and scheduling (IPPS) problems. The algorithm is based on sorting the operations into different priorities. The experimental results show that the proposed algorithm can effectively solve IPPS problems.

Index Terms—Distributed process planning, Integrated process planning and scheduling, Optimization algorithm, Priority-sort based optimization algorithm.

I. INTRODUCTION

In traditional manufacturing, process planning and scheduling were treated as separated entities, usually performed sequentially with process planning preceding scheduling. This approach generally limits a manufacturing system's flexibility to adapt to dynamic changes, can result in unbalanced resource loads, infeasible process plans or non-optimized process plans and schedules[1]. To rectify these shortcomings and achieve a global improvement for the performance of a manufacturing system, integrated process planning and scheduling (IPPS) is essential[2]. The modern manufacturing environment is characterized by the integration of all its entities into a cohesive unit, without IPPS a true Computer Integrated Manufacturing System (CIMS), which strives to integrate different phases of manufacturing into a single unit, can never be achieved.

Although IPPS is imperative to modern manufacturing systems, the actual task of integrating them together is quiet tedious. The idea of integrating process planning and scheduling together was first presented by [3]. In ideal circumstances merging both process planning and scheduling in a single entity is the objective but as both process planning and scheduling problem are nondeterministic polynomial hard (NP-Hard), their combination not only increases the complexity of the problem but also the search space [4]. So most of the works have been aimed at increasing the information exchange between process planning and scheduling. A lot of works have been done on IPPS and several classification schemes have been proposed for the IPPS problem, the most popular

approach among researchers [5]-[8] etc. is to highlight the three different techniques for solving the IPPS problem as:

Non Linear Process Planning (NLPP) in which a set of pre-determined process plans, ranked in some order, are choose and the scheduling tries to choose the best process plan.

Closed Loop Process Planning (CLPP), a dynamic feedback from the scheduling to the process planning phase is the highlight of the CLPP approach. The process planning phase tries to produce an improved process plan based on this feedback.

Distributed Process Planning (DPP), is a two phase approach. In the preplanning phase jobs are analyzed and corresponding process are determined. In final planning, job operations are matched with available production resources. The result is dynamic process planning and production scheduling which is constrained by real-time events[9].

Because of increasing complexity of products and processes it is believed that traditional exact methods are not effective in solving the IPPS problem[10]. The rapidly growing computational capability of modern computers is fast becoming the focal point for solving the IPPS problem. Heuristic algorithms because of their capability to effectively solve combinatorial optimization problem have been vastly employed in this regard. The proposed algorithm is a heuristic approach to solving the IPPS problem.

Dispatching rules have been used for scheduling problems because of the ease and simplicity of their application. Usually from a group of rules one is chosen and applied in a single pass, with no search, to construct a schedule. Sundaram *et al.* [11] use a dispatching rule approach to prepare an initial schedule for refinement. The application of dispatching rules in the proposed algorithm is slightly different.

In the algorithm presented here the planning and the scheduling are done simultaneously. A combination of two ideas is used for achieving the integration of process planning and scheduling. Different operations are assigned different priorities to establish on order for assigning operations while the selection of a machine to perform that operation is done using a set of dispatching rules. Before assigning any operation to a machine, the job and the current status for the machines are analyzed. According to that analysis, the machine is chosen for the operation. This method deals with the problem as if it was the online problem.

The remainder of this paper is organized as follows. Section 2 presents a description of the IPPS problem and the detail description of the priority-sort based. The

experimental results have been discussed in section 3, while section 4 is the final section containing the concluding remarks and future research directions.

II. PROBLEM DESCRIPTION

A. Problem Statement

The IPPS problem can be stated as [12]:

“Given a set of N parts which are to be processed on machines with operations including alternative manufacturing resources, select suitable manufacturing resources and sequence the operations so as to determine a schedule in which the precedence constraints among operations can be satisfied and the corresponding objectives can be achieved.”

The most common criteria considered for optimization include makespan, job tardiness and balanced machine utilization. Since the IPPS problem is being treated as a single objective function problem in this paper so the criterion considered is the makespan.

The IPPS problem will be subjected to the following assumptions [13]

- All parts are independent of each other and each machine can handle only one part at a time.
- Multiple operations of the same part cannot be performed simultaneously, even on the same machine.
- All parts and machines are available simultaneously at the beginning.
- Transportation time of a part between machines is negligible as compared to the processing time.
- The setup time is included in the processing time.
- Unless bounded by the precedence constraints among operations, different operations on a part can be performed in any sequence.

The objective of minimizing the makespan can be given as [2]:

$$MakeSpan_{min} = \max(c_i) \quad (1)$$

where c_i is the earliest completion time of part ‘ i ’ $\{\forall i \in [1, N]\}$ and N is the total number of parts.

B. Priority-Sort Based Optimization Algorithm (PSBO)

The presented algorithm consists of three distinct actions:

- Classifying the operations into different priority levels
- Generate an order of operations for each priority level
- Starting from the highest priority level, assign these ordered operations to machines using different dispatching rules.

Fig. 1 shows the working of the priority sort algorithm.

1) Concept of priority value:

A priority value is an indication of how important a particular operation is compared to other. Higher priority values indicate operations with higher priority. The least important operations are assigned a priority value of 0 and the remaining operations are prioritized with reference to this value. An operation with a higher priority level is always assigned to a machine prior to an operation with

lower priority value.

The objective of assigning priorities is to accelerate the optimization process. Instead of searching less likely regions the search is directed to areas with higher probability of finding the optimum. Generally, when the complexity of a problem increases the priorities will ensure that less time is required to reach the optimum.

The priorities are assigned using the following guidelines.

Last operations (which are not needed for any other operations) have a priority value of 0.

Independent Operations (which are not involved in any constraint relationship) will have a random priority value between 0 and the maximum priority value for that part.

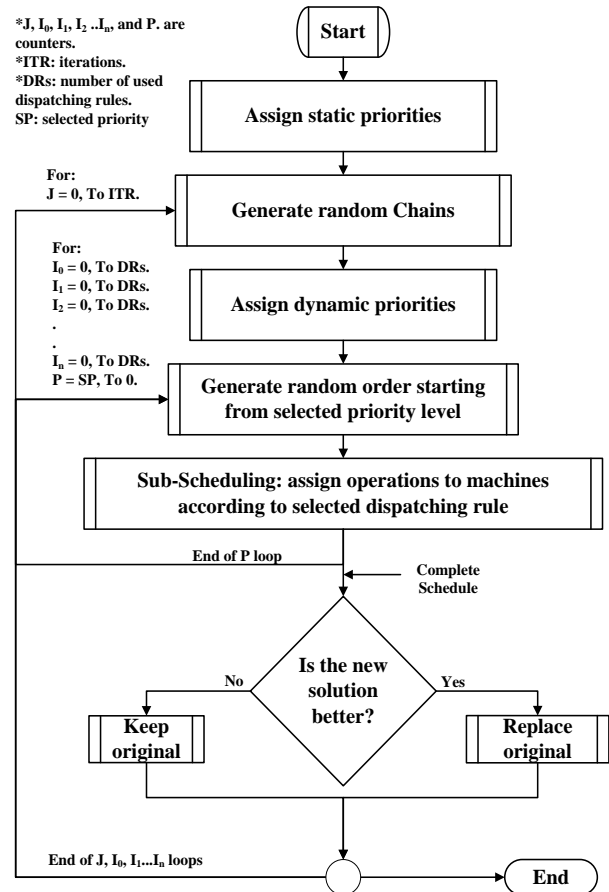


Fig. 1. Flowchart for the priority-based optimization algorithm.

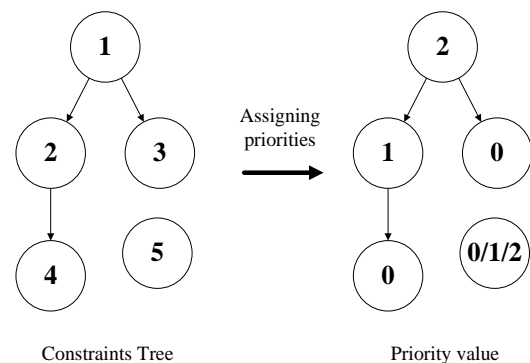


Fig. 2. Assigning priorities to a job.

All other operations will have static priority values one greater than the maximum priority value for their following

operations.

Fig. 2 shows an example of how priorities are assigned with reference to a constraint tree. Since operations 3 and 4 are the last operations in the constraint tree so both of them will be assigned a priority value of 0. Operation 1 is followed by operations 2 and 3, since operation 2 has the higher priority so the priority for operation 1 is one plus the priority of operation 2, hence a priority of 2. Since operation 5 is an independent operation so it will have a varying priority between 0 – 2.

In the algorithm two kinds of priorities are assigned to each operation, namely static and dynamic priorities. Both static and dynamic priority values are assigned to operations according to the rules above. Static priorities are assigned only once, before generating the chains for the first time, while dynamic priorities are assigned after each random generation of order for every job. Since each part will have a single continuous chain, so every operation for that part will have a unique dynamic priority value.

2) Building the solution

Generate chains:

For each part, based on static priorities we randomly generate a sequence of operations called as a chain. In this algorithm the random generation is bounded by the condition that an operation with a priority value 'n' cannot come before another operation with a priority value of 'n+1'. Two possible chains for the part in fig. 2 are shown below in Table I.

These chains are then used to assign dynamic priorities for operations in every part.

Generate operations orders:

After dynamic priorities have been assigned the chains are then sorted into different sets(operations that have the same dynamic priority value are placed in the same set). Next a random order of the operations in each priority set is generated.

Planning & Scheduling:

Starting with the highest priority order the operations are assigned to machines one by one using the dispatching rules.

By scheduling the last operation in the lowest dynamic priority order, a complete process plan and schedule will be obtained.

Intensifying local search:

New solutions are obtained by generating new random orders for each priority level and simultaneously switching the dispatching rules along the population.

Before the next random generating for each priority level P_n , there will be DR times (DR is the number of dispatching rules) random generating for the priority level P_{n-1} , i.e. for the set with the highest priority level, the random generation is redone after generating DR times of the lower priority level. So if there are 'n' priority levels, and a random order is generated for the ' n^{th} ' level for DR times, then the '0' level set is randomly changed for DR^n times. The Table II shows how many times a random order is generated for each subsequent priority level as well.

The presented way of generating random orders is meant to search locally enough and as quick as possible before moving to another location in the solution space.

3) Dispatching rules used:

Once an operation has been selected, the selection of the machine can be done at random but to make the selection procedure more intelligent a set of dispatching rules is used to aid the machine selection procedure for each operation. In this paper, five different dispatching rules are used.

- Choose the machine that achieves the Earliest Possible Start (EPS) for the process.
- Choose the machine that achieves the Earliest Possible Finish (EPF) for the process.
- Choose the machine that achieves the Shortest Processing Time (SPT).
- Choose the machine by which we can have the Shortest Idle Time (SIT).
- Choose the machine by which we can best Balance the Machining Time (BMT) among the machines.

An example of using the aforementioned dispatching rules is shown in Fig. 3. Operation 2 is dependent only on operation 1. Operations 1, 3, 4, 5 and 6 have been already assigned to the machines M1, M2, M3, M4 and M5 respectively. Operation 2 has four possible process plans which are shown dotted in Fig. 3. These process plans are sorted according to each dispatching rule in Table III.

TABLE I: THE POSSIBLE CHAINS FOR THE JOB IN FIG. 1

Chain 1:	1	2	3	4
Chain 2:	1	2	4	3

TABLE II: TIMES OF RANDOM GENERATING FOR EACH PRIORITY LEVEL

Priority level:	P_n	P_{n-1}	...	2	1	0
Times of random generating	DR^1	DR^2	...	DR^{n-2}	DR^{n-1}	DR^n

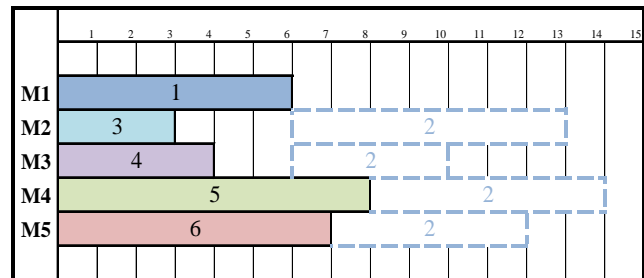


Fig. 2. An example of using dispatching rules to assign operation 2 to a machine.

TABLE III: POSSIBLE PROCESS PLANS FOR OPERATION 2 SORTED ACCORDING TO EACH DISPATCHING RULE (STARTING BY THE BEST MACHINE)

EPS	EPF	SPT	SIT	BMT
M2	M3	M3	M5	M3
M3	M5	M5	M4	M2
M5	M2	M4	M3	M5
M4	M4	M2	M2	M4

The selection procedure of the dispatching rules is done systematically by dividing the population into DR groups. Each dispatching rule will be applied only to a single group of the population for one generation. In the next generations, the dispatching rules will be switched among the population groups.

This selection procedure guarantees that:

- All the rules are applied for each level of dynamic priority values.
- All the rules are applied along the whole population.

III. EXPERIMENTAL STUDIES AND DISCUSSION

A. Test Problems and Results:

In this paper, PSBO has been implemented on five different problems. The objective in each experiment is to achieve the smallest makespan. All jobs in the presented problems have no alternative operations.

1) Problem 1:

This problem is adopted from Sundaram *et al.* [11] with 5 jobs, 4 operations for each, and to be processed on 5 machines. The best makespan obtained is 33.

The Gantt chart for the best solution is shown in fig. 4.

2) Problem 2:

Problem 2 has 5 jobs, 5 machines and 13 operations. Adopted from Moon *et al.* [14] The best solution obtained is 14. The Gantt chart for the best solution is shown in fig. 5.

3) Problem 3:

Adopted from Shao *et al.* [15] problem 3 has 5 jobs, 21 operations, and 6 machines. The best solution obtained is 28. Gantt chart is in fig. 6.

4) Problem 4:

In this problem, 8 jobs, 20 operations and 6 machines. Adopted from Lee *et al.* [16] The best makespan obtained is 23. Gantt chart is shown in fig. 7.

5) Problem 5:

The problem is adopted from Li *et al.* [1] It has 6 jobs, 18 operations for each, and to be processed on 5 machines. Best solution is 27. As shown in fig. 8.

B. Discussion:

The PSBO algorithm puts emphasis on focusing more on the critical regions of the search space (indicated by the priorities). The results, which are listed in table IV, show that in spite of eliminating parts of the solution space at two stages of the algorithm, PSBO was able to reach the optimal solution (makespan) in most problems. For some other problems, it has reached near optimal solutions, which are still the same as the best obtained by using popular heuristic algorithms.

Reducing the solution space has effectively enabled the algorithm to accelerate the optimizing process.

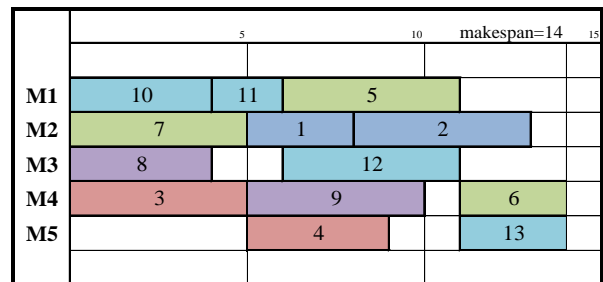


Fig. 5. Gantt chart for the solution for problem 2.

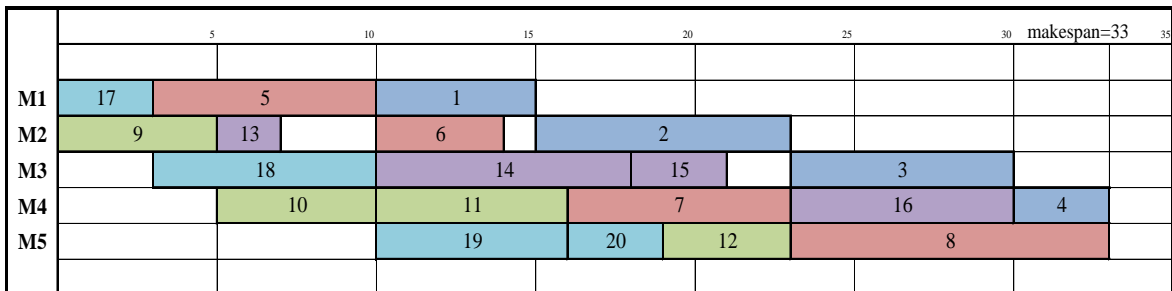


Fig. 4. Gantt chart for the solution for problem 1.

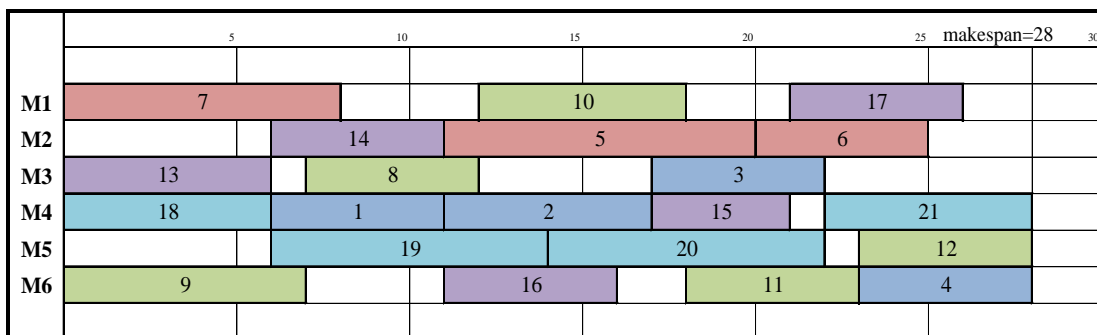


Fig. 6. Gantt chart for the solution for problem 3.

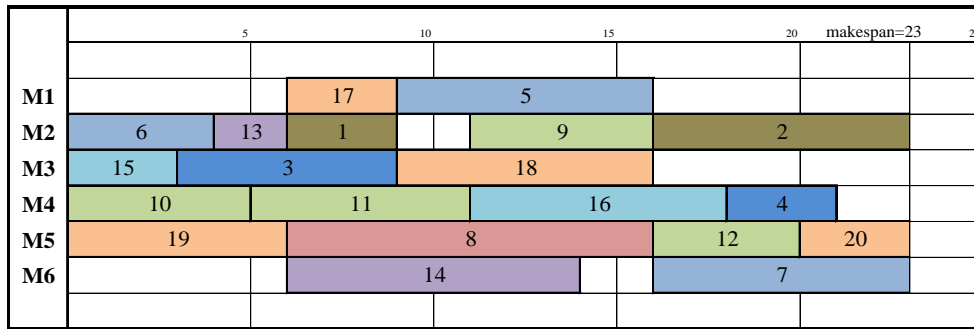


Fig. 7. Gantt chart for the solution for problem 4.

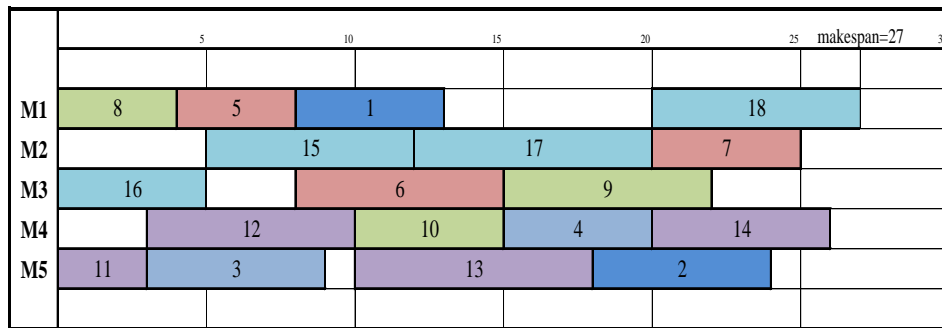


Fig. 8. Gantt chart for the solution for problem 5.

TABLE IV: BEST SOLUTION (MAKESPAN) FOR EACH PROBLEM

Exp.	M/C	Jobs	Opr	Makespan(Solution)		
				In Source	Best in Literature	PSBO
1	5	5	20	38	33	33
2	5	5	13	16	14	14
3	6	5	21	28	28	28
4	6	8	20	34	23	23
5	5	6	18	27	27	27

IV. CONCLUSION AND FUTURE WORK

In this paper, a novel optimization algorithm, which is based on priority sorting for the operation, is presented for integrated process planning and scheduling problem.

Five problems from the literature were taken to evaluate the performance of the proposed algorithm, the results have shown that the proposed algorithm was able to give optimal or near optimal solutions.

In the future research, more benchmark problems will be used to demonstrate the efficiency of this algorithm. In addition, there would be many potential enhancements to be applied for the proposed algorithm. First, is to involve the whole solution space which would enhance the quality of the solutions. Second, is to use a hybrid algorithm in order to accelerate the optimization process. Eventually, the proposed algorithm has also good potential to be used to solve multi-objective IPPS problems.

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous referees whose comments have helped a lot in improving this paper.

REFERENCES

- [1] X. Li, C. Zhang, L. Gao, W. Li, and X. Shao, "An agent-based approach for integrated process planning and scheduling," *Expert Systems with Applications*, vol. 37, pp. 1256-1264, 2010.
- [2] Q. Lihong and L. Shengping, "An improved genetic algorithm for integrated process planning and scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 58, pp. 727-740, 2012.
- [3] G. Chryssolouris, S. Chan, and N. P. Suh, "An Integrated Approach to Process Planning and Scheduling," *CIRP Annals - Manufacturing Technology*, vol. 34, pp. 413-417, 1985.
- [4] N. Cai, L. Wang, and H. Y. Feng, "GA-based adaptive setup planning toward process planning and scheduling integration," *International Journal of Production Research*, vol. 47, pp. 2745-2766, 2009.
- [5] H. C. Zhang and M. E. Merchant, "IPPM - A Prototype to Integrate Process Planning and Job Shop Scheduling Functions," *CIRP Annals - Manufacturing Technology*, vol. 42, pp. 513-518, 1993.
- [6] S. H. Huang, H. C. Zhang, and M. L. Smith, "A progressive approach for the integration of process planning and scheduling," *IIE Transactions*, vol. 27, pp. 456-464, 2013/01/09 1995.
- [7] N. N. Gindy, S. M. Saad, and Y. Yue, "Manufacturing responsiveness through integrated process planning and scheduling," *International Journal of Production Research*, vol. 37, pp. 2399-2418, 1999.
- [8] A. Baykasoglu and L. Ozbakir, "A grammatical optimization approach for integrated process planning and scheduling," *Journal of Intelligent Manufacturing*, vol. 20, pp. 211-221, 2009.
- [9] R. K. Phanden, A. Jain, and R. Verma, "Integration of process planning and scheduling: a state-of-the-art review," *International Journal of Computer Integrated Manufacturing*, vol. 24, pp. 517-534, 2011.
- [10] G. Jezic, M. Kusek, N. T. Nguyen, R. Howlett, L. Jain, L. Zhang, T. N. Wong, and R. K. Fung, "A Multi-Agent System for Dynamic Integrated Process Planning and Scheduling Using Heuristics," in *Agent and Multi-Agent Systems Technologies and Applications*, vol. 7327, Springer Berlin Heidelberg, 2010, pp. 309-318.
- [11] R. M. Sundaram and S. S. Fu, "Process Planning and Scheduling-A Method of Integration for productivity Improvement," *Computers in Industrial Engineering*, vol. 15, pp. 296-301, 1988.
- [12] Y. W. Guo, W. D. Li, A. R. Mileham, and G. W. Owen, "Applications of particle swarm optimisation in integrated process planning and scheduling," *Robotics and Computer-Integrated Manufacturing*, vol. 25, pp. 280-288, 2009.

- [13] X. Y. Li, X. Y. Shao, and L. Gao, "Optimization of flexible process planning by genetic programming," *The International Journal of Advanced Manufacturing Technology*, vol. 38, pp. 143-153, 2008.
- [14] C. Moon, Y. H. Lee, C. S. Jeong, and Y. Yun, "Integrated process planning and scheduling in a supply chain," *Computers & Industrial Engineering*, vol. 54, pp. 1048-1061, 2008.
- [15] X. Shao, X. Li, L. Gao, and C. Zhang, "Integration of process planning and scheduling—A modified genetic algorithm-based approach," *Computers & Operations Research*, vol. 36, pp. 2082-2096, 2009.
- [16] Y. H. Lee, C. S. Jeong, and C. Moon, "Advanced planning and scheduling with outsourcing in manufacturing supply chain," *Computers & Industrial Engineering*, vol. 43, pp. 351-374, 2002.



Ghiath Al Aqel received his B.Sc. degree in mechanical design and production from Damascus University Damascus, Syria, 2008. He worked as planning engineer at Elsewedy Electric, Damascus, Syria. Currently he is a Master's degree student at Huazhong University of Science and Technology (HUST). His research interests include, Optimization, intelligent algorithms and scheduling.



Muhammad Farhan Ausaf received his BS degree in mechanical engineering from University of Engineering & Technology Peshawar, Pakistan. He was a lecturer in the Department of Mechanical Engineering at HITEC University, Pakistan. Currently he is a PhD student at Huazhong University of Science and Technology (HUST). His research interests include, Optimization, intelligent algorithms and scheduling.



Xinyu Li received his Ph.D. degree in Industrial Engineering from Huazhong University of Science and Technology (HUST), China, 2009. He is now a Lecture in the Department of Industrial & Manufacturing Systems Engineering, School of Mechanical Science and Engineering, HUST. His research interests include integrated process planning and scheduling, flexible job shop scheduling and intelligent algorithm.



Liang Gao received his B.S. degree in mechatronic engineering from Xi'dian University, China, in 1996, and the Ph.D. degree in Mechatronic Engineering from Huazhong University of Science and Technology, China, 2002. Now, as a Professor and Head of the Department of Industrial & Manufacturing System Engineering, Huazhong University of Science and Technology, his research interests include operations research and optimization, scheduling, etc. He had published in more than 40 journal papers, including Computer & Operations Research, Expert Systems with Applications, Computers & Industrial Engineering, and so on.