Integrated multi-objective process planning and flexible job shop scheduling considering precedence constraints

Elahe Shokouhi
Department of Industrial Engineering, Shomal University, Amol, Iran

ABSTRACT
Process planning and scheduling decisions are independently performed in a manufacturing system in traditional approaches. Integration of these two decisions provides significant benefits such as enhancing the productivity of manufacturing resources, lead time reduction, and decreasing total production costs. An approximately 10% improvement has been created in the small problems by this integration. In this research, a mathematical model is proposed to simultaneously make these decisions. Flexible job shop, as a prevalent configuration in production systems, is the supposed configuration in the study in which the feasible process plans are recognized based on the precedence relations between operations. Due to the NP-Hardness of the problem, a solving procedure based on the genetic algorithm is devised so that all the alternative process plans could be considered implicitly. Makespan, critical machine workload, and machines total workload are considered as objective function. These has been combined in a weighted-sum to form an objective function and solved in pareto space. A comparison of the exact solutions with the proposed algorithms (WGA & NSGA II) results confirms the efficiency and the effectiveness of the proposed algorithms in obtaining the final solutions.

1. Introduction
Considering the flexibility of different aspects alongside the multiple objectives may tend to resolving and filling a gap between theoretical planning and real circumstances in production systems programming. This paper puts forth contemplating the concurrency of process planning and operation scheduling in the job-shop structure. Process planning looks forward to transform an engineered prototype into a final product through the mass production system according to which it requires specific information about the sequences of machineries, facilities, and jigs and fixtures (Guo et al., 2009). Process planning and scheduling are carried out as independent tasks in traditional approaches toward the production systems. In other words, those process plans should be completed that pose necessities in scheduling decisions. The machineries and utilities required for operations and sequences are some of...
these necessities that are inclined towards considerable reduction in makespan. Scheduling
and process planning related decisions integration lead to the significant improvements in
manufacturing stations according to which the flexibility could be accounted as one of its
accompanied results (Li et al., 2010).

Since 1984 that Chryssolouris, Chan, and Cobb (1984) contributed Integrated Process
Planning and Scheduling (IPPS), numerous practitioners pursued their novelties (Larsen
& Alting, 1992) as classified IPPS into three categories:

1. Flexible/nonlinear process planning that is deployed by Jain, Jain, and Singh (2006),
Kumar and Rajotia (2003), and Leung et al. (2010) through such an approach is that
all the potential process plans would be considered and archived. The process with
higher priority could be put into practice and providing that the objective function
is satisfied, it could be petrified. Otherwise, the next priority will be chosen.

2. Closed loop process planning is another approach which is used by Wang (2009)
and Zhang, Saravanan, and Fuh (2003). The process plan in this kind of approach
is produced regarding the resources status and relying on the dynamic feedback
from the workshop.

3. The latest approach is the distributed process plan including simultaneous sched-
uling and planning. Here, one would face with two phases of preliminary and final
planning. This one needs ad-hoc software capabilities accompanied by a reliable
hardware configuration (Sen & Jain, xxx).

Many of the IPPS works considered the production system configuration as job-shop
ones because of the similarity of these models to the real world circumstances. Kim, Park,
and Ko (2003) proposed a symbiotic evolutionary algorithm for the integration of process
planning and job shop scheduling (Tarimoradi et al., 2015). In recent years, some research
has focused on the multi-objective problem according to which it was an efficient approach
to reduce the gap between theories and applied programming. Mohapatra et al. (2015)
deployed the Non-Dominated Sorting Genetic Algorithm (NSGA) with controlled elitist
for the mentioned integration. Likewise, Shao et al. (2009) offered integration of process
planning and scheduling by the Genetic Algorithm and to improve the algorithm efficiency,
they used the heuristic method for operators and solution representation. In their considered
problem, sequences and operations are considered simultaneously. Zhang and Gen (2010)
used a multi-objective GA for solving process planning and scheduling problem through
a distributed production system. Zhao et al. (2015) used a multi-objective ant colony for
job-shop scheduling with multiple process scenarios. Li, Gao, and Li (2012) used Nash
Equilibrium to integrate scheduling and multi-objective process planning by relying on a
game theory approach. In Sobeyko and Mönch (2017), appropriate neighborhood structures
for VNS are proposed.

In a comprehensive review conducted by Ausaf, Li, and Gao (2014) they looked through
more than 50 works published in this context and categorized these IPPS into four main
classes as Tabu Search, GA, Ant Colony, and hybrid algorithms with regard to their used
algorithms.

A look inside the previous works reveals despite the fact that practitioners in this field
of research area contributed different novelties manipulating technical issues, one in which
this paper has contemplated, i.e. multi-objective job-shop scheduling with respect to mul-
tiple possible process plans is an unprecedented innovation. Through the contribution
of this paper, the FJSP configuration problem is modeled and an algorithm is proposed in which makespan, machine load, and total load on machineries are the simultaneous considered goals. Likewise, in previous works, the potential process plans are specifically produced, and in large scale, problems with many potential candidates exponentially make the problem more complicated. Whilst the proposed contribution herein, there is no need to provide a complete list of feasible plans, so that considering the predecessor and successor relations between operations through the algorithms’ operators are necessary for the optimum solutions.

The rest of this paper is organized as follows: the contribution of this paper as an integration of scheduling and process planning and its embodiment in a mathematical format is provided in Section 2. Afterward, the proposed methods for solving the problem relying on GA are expounded in Section 3. To evaluate the proposed GA-based algorithms, some problems in different scales are designed and their results by the algorithms are checked in Section 4. Finally, concluding remarks and some insinuations for future works are indexed in Section 4.

2. The background

As mentioned, scheduling was carried out after the process planning. This method had an obstacle in improving productivity and production system response as mentioned in Saygin and Kilic (1999) and Kumar and Rajotia (2003):

(A) Through practical production, the process plan devises the job separately. For each job, production resources in the workshop are budgeted regardless of the competition for dedicating them (Usher & Fernandes, 1996), and this could result in recurrent selection. Hence, the process plan would be far ahead of reality and be hard to reach in the workshop for a class of jobs (Lee & Kim, 2001). Thus, the obtained optimum process plan would be unfeasible.

(B) Scheduling was usually carried out after the process planning. In the scheduling step, considering the process plan was inevitable. A fixed process plan might lead the scheduling into surplus bottleneck or unbalanced resource loads.

(C) Even if process planners consider the current resources constraint in the shop, the emerging constraint of the process plan stage might be affected as a result of tardiness between devise and implementation stages and change. The recent studies revealed that between 20 and 30% of process plans in a given period for adapting with dynamic changes in the production environment must be rescheduled (Kühnle, Braun, & Bühring, 1994).

(D) In most cases, for both the scheduling and process plan, a single criterion optimization method is used while in a real production environment, more than an objective should be taken into account (Kumar & Rajotia, 2003). Process planning highlights the technical requirements while scheduling points out the time and resources dedication importance. Thus, process plan and scheduling might have opposite objectives while IPPS provides a possibility for having these seemingly contradicting criteria concurrently. This is while such integration effects the production cost reduction, bottleneck limitation in the production line and improvement in facility productivity which is ostensibly clear. Overcoming these problems, a
comprehensive study and using systemic integration for scheduling and process planning are necessary.

The concept of integration of process planning and scheduling was initially developed by Chen and Khoshnevis (1993). They described the basic issues involved in the integration of process planning and scheduling through a heuristic algorithm. The approach was adopted and extended in Tan and Khoshnevis (2000). Based on the concept of distributed/progressive approach, Brandimarte and Calderini (1995) proposed a solution methodology for a joint process selection and job shop scheduling problem taking both operation cost and makespan into account within a multi-objective framework.

A progressive approach is developed in the form of a three-phase method (preplanning, pairing planning and final planning) in which the interaction between process planning and scheduling reduces the complexity in the real-manufacturing environment (Huang, Zhang, & Smith, 1995). Saygin and Kilic (1999) presented a framework that integrates flexible process plans with offline (predictive) scheduling in FMS. The framework consisted of four integrated stages such as machine tool selection, process plan selection, scheduling and rescheduling in modules to reduce the completion time. Likewise, a dissimilarity maximization method was presented for selecting the appropriate process plan for a part mix where the parts have alternative process plans. Dolgui, Guschinsky, and Levin (2009) discussed a special case of transfer machine (machine with rotary table), to carry out an optimal assignment of operations to spindle heads and working positions. The simplified way of explaining this problem is to partition the setups of all machining operations into subsets leading to minimization of the number of working positions and spindle heads. They reduced the puzzle to the constrained shortest path search problem. The proposed approach is demonstrated in their work for all optimal solutions while respecting production rate, precedence and compatibility constraints. Dolgui, Gordon, and Strusevich (2012) presented an approach to solve the single machine scheduling problem (minimization of makespan) with positional-dependent processing time under the precedence constraint. A framework for a similar problem was developed Jain et al. (2006) which can quickly integrate both functions and can be implemented in a company without dismantling and reorganizing the existing process planning and scheduling department. An effective model for integration of process planning and scheduling in the batch manufacturing environment was developed Zhang et al. (2003). The integration problem was modeled at two levels: process planning and scheduling which are linked by an intelligent facilitator. The uniqueness of this approach was characterized by the process planning strategy and the also intelligent facilitator which makes the full use of plan solution space intuitively to reach the satisfactory schedule. To meet the requirement, due to the rapid change in capacity and functionality, a new methodology was developed by Wang, Feng, and Cai (2003) for distributed process planning. Wang et al. (2009) presented a dynamic approach to reduce tardy jobs through integration of process planning and scheduling in the batch manufacturing environment. Delorme, Dolgui, and Kovalyov (2012) developed a combinatorial approach to solve the problem for transfer and assembly lines. Lian et al. (2012) presented an algorithm namely the Imperialist competitive algorithm which minimized the makespan for an IPPS problem in which the operation flexibility, sequencing flexibility, processing flexibility of each job are taken into account simultaneously. Li et al. (2012) proposed a game theory-based approach to deal with multi-objectives in the IPPS problem. Rajkumar et al. (2010) and Zarandi
et al. (2015, 2015) proposed a greedy randomized adaptive search procedure algorithm for the integration of process planning and scheduling in a flexible job-shop environment. Min, Li, and Shensheng (2004) proposed integration of computer-aided process planning (CAPP) and production planning and scheduling (PPS)-based on distributed and dynamic process planning (ICAPPS) which includes the designing, part planning, shop planning and scheduling layer for single-piece, small-lot and made-to-order production. Leung et al. (2010) developed an ant colony-based optimization algorithm in an agent-based system to integrate process planning and scheduling. Lihong and Shengping (2012) developed a mathematical model and improved GA to integrate process planning and scheduling. Zhang and Fujimura (2012) proposed an improved VEGA with archive (iVEGA-A) to deal with multi-objective IPPS problems, with consideration of being given to the minimization of both makespan and machine workload variation. From the above research, it is found that setup planning plays an important role for the integration of process planning and scheduling, because it is closely related to process plan generation and machine selection by considering the machining flexibility, sequencing flexibility and machine availability. There are two major constraints in setup planning that come from design specifications and manufacturing resources. Generally, most of the setup plans satisfy the first constraint. The concept of hybrid graph theory, which transferred the indirect graph to directed graph by changing the two-way edge into one way edge, which is effectively used in setup planning, was introduced by Alavidoost, Tarimoradi, and Zarandi (2015a), Alavidoost et al. (2014), Tarimoradi, Alavidoost, and Zarandi (2015), Zhang and Lin (1999), and Zhang et al. (2001). Here, the tolerance relations were used as the critical constraints of setup planning (Zhang et al., 2001) applied to the tolerance decomposition, fixture design and manufacturing capability in setup planning to determine the number and sequences of setups, select the datum, machining features and operations in each setup. A hybrid approach to setup planning optimization using GA, simulated annealing (SA) and a precedence relationship matrix (PRM) was proposed (Ong, Ding, & Nee, 2002). Here PRM acted as the main constraint of setup planning optimization. NSGA-II is used in this paper. A large number of variants of NSGA are presented in literature. An efficient NSGA is discussed in Alavidoost, Tarimoradi, and Zarandi (2015b) and Tran et al. (2013). The controlled elitist NSGA-II is introduced by Deb and Goel (2001). A fast elitism-based NSGA-II is discussed by Deb et al. (2002). NSGA-II is used by Chaube, Benyoucef, and Tiwari (2012) for reconfigurable manufacturing systems. Mohammadi, Karampouraghhghi, and Samaei (2012) similarly used the multi-objective optimization model based on SA for process planning and scheduling. A PSO-based approach is used for this integration as discussed in Mohapatra et al. (2015). Application of the modified version of PSO in process planning and scheduling is discussed in Guo et al. (2009). An improved vector evaluated Genetic algorithm is used in Zhang and Fujimura (2012). For an automated setup planning integration, the integration of CAPP and computer-aided design was solved based on geometric analysis, precedence constraints analysis, kinematic analysis, tolerance analysis and force analysis (Huang & Xu, 2003). Good process plans were generated by considering machine processing and fixture constraints of the system given in Gologlu (2004). It formed the setups by grouping the features to be machined taking into account any precedence relations between the features. As considered, despite the fact that there are a number of related works to what is intended herein as Brandimarte and Calderini (1995), Moslehi and Mahnam (2011), Xia and Wu (2005), and Zhang et al. (2009), related to what that is pronounced here as integration of
MO-process planning and Flexible Job-shop scheduling regarding precedence relations is unprecedented.

3. Problem statement

The considered problem here is generalized FJSP as a Multi Objective Flexible Job Shop Scheduling Problem with Alternative Process Plans (MOFJSP-APP). This problem is a concurrent consideration of multi-objective flexible job shop and process plan per se. In a job shop platform, each operation with a pre-identified sequence could be processed on a specific machine. Whilst in a flexible kind of job shop different machines are possible for each operation and as a result in an FJSP alongside determining the job sequences, their allocation to the feasible machines is another issue. This point results in FJSP complexity even higher than JSP which is categorized as the NP-Hard problem per se (Brandimarte, 1993).

As mentioned in the considered problem, the flexibility is embedded in process plans by predecessor and successor relations between operations configuring the whole job. Table 1 represents 3-jobs each of which their constituent operations’ inter-connections are declared. In case of the MOFJSP-APP problem, it is specified that for job No 1, there are two possible processes according to the graph. Thus, the decision on three sub-problems is the point here: selecting a possible process plan for each job (regarding their predecessor and successor relations), allocating the machines to each operation (i.e. which operation should be processed on which machines), and determining the operation sequences (i.e. in what sequence should the allocated operations to a machine be arranged). It is with respect to these three concurrent decisions that integration will provide better and qualified results.

By solving the problem with the proposed algorithm, the selected ones from possible operations are circled. The corresponding Gantt charts for the problem final result are exposed in Gantt chart (1). The integration model is illustrated in Figure 1 and described as follows:

First, the feature extraction and CAPP generate all the possible process plans regardless of the shop resources. These process plans are saved in the database and the precedence constraint on each operation for each job and machining time of all the alternative machines for all operations. Then machine selection, operation sequencing, and scheduling result in optimal integration of machine scheduling with their process plans.

Demir and Kürşat İşleyen (2013) divided the model of FJSP due to the sequence sub-problem to sequence-position, precedence, and time-indexed variable in which the use of the precedence variable has caused integration process planning and scheduling in FJSP. Based

| Problem | Integration WGA | Integration CPU Time | No Integration WGA | No Integration CPU Time | Improvement percent due to the integration |
|---------|----------------|----------------------|--------------------|------------------------|------------------------------------------|
| P1      | 118.9          | 3.520                | 140.54             | 4.111                  | 15.4                                     |
| P6      | 50.1           | 6.598                | 56.93              | 7.231                  | 12                                       |
| P7      | 89.9           | 5.222                | 105.7              | 5.89                   | 15                                       |
| P2      | 140.6          | 4.220                | 167.38             | 5.158                  | 16                                       |
| P3      | 166.1          | 4.898                | 214.322            | 5.887                  | 22.5                                     |
| P4      | 152.8          | 4.907285             | 190.761            | 5.087                  | 19.9                                     |
| P5      | 430.3          | 26.68365             | 576.8              | 28.4555                | 25.4                                     |
| P9      | 20.5           | 3.777212             | 22.8               | 4.0525                 | 10.11                                    |
| P10     | 21.3           | 5.61404              | 23.93              | 6.0233                 | 11                                        |
| P11     | 21.3           | 4.055                | 23.61              | 4.5352                 | 9.8                                      |
on the results of the research model M2, that are required from the categories precedence variables has integrated scheduling and planning of the process as the most efficient model in terms of computation time and results optimization objective function (makespan), was selected.

For transparency and other reason, that why has used this integration in this research. I decided to consider small problem with assuming integration and no integration. In the no integration, one of the possible process planning can be chosen at random. The precedence relations in the selected process planning have been considered as the input data in the proposed algorithm. The results of WGA have obtained in following table. As you can approximately see improvement of 10%, increases in the problem with bigger-scale. In the scale of the problem which complexity is low, the existence of the integration shows its performance. In large-scale that complexity of the problem increases, Integrations existence in the improvement of the results is impressive and in the production systems will more efficient.

4. Mathematical formulation

Manne (1960) proposed a model for FJSP. Here, their model is developed to consider the multi-objective and flexibility in production plans according to the predecessor relations and finally results in MOFJSP-APP.

Assumptions:
The MOFJSP-APP according to which this paper’s mathematical problem is based on is as follows:

- Each machine could be engaged in processing one of the products.
- Operation processing on a machine continues and the break is not authorized.
- There are no predecessor relations between operations from different jobs.
- Set up times are independent and included in processing times.
- The required times for preparing operations are included in processing times.
- Machines are independent.

A problem including 3 jobs each of which with 4, 2, and 3 operations, respectively are presented in Table 2. In processing each of the operations there is a set of feasible machines
Table 2. Data for a sample FJSP-APP problem.

| Operation | Processing time on alternative machines | Precedence graph | Feasible process plans |
|-----------|----------------------------------------|------------------|------------------------|
|           | M1 | M2 | M3 | M4 |                | O11 | O12 | O13 | O14 | O11 | O12 | O13 | O14 |
| Job 1     | O11 | 10 | –  | 8  | 12            |     |     |     |     | O11 | O12 | O13 | O14 |
|           | O12 | 5  | 6  | 9  | –             | O12 |     |     |     |     |     |     |     |
|           | O13 | 10 | 15 | –  | 8             |     | O13 |     |     | O12 |     |     |     |
|           | O14 | –  | 3  | 5  | –             |     |     | O12 | O13 |     |     |     |     |
| Job 2     | O21 | 8  | –  | 10 | 6             | O21 | O22 |     |     | O21 | O22 |     |     |
|           | O22 | 12 | 7  | 5  | 9             |     |     | O21 | O22 |     |     |     |     |
| Job 2     | O31 | 10 | –  | 12 | –             | O31 | O32 |     |     | O31 | O32 |     |     |
|           | O32 | 5  | 6  | 12 | 6             |     |     | O31 | O32 |     |     |     |     |
|           | O33 | 8  | 9  | –  | 9             |     |     | O31 | O32 |     |     |     |     |

and one of them could be selected for allocation. The 4th column represents precedence relations between the operations in each job and after that possible processing plans are listed. This problem considers the sequence of operations on possible machines in order to meet the objectives.

The indices, parameters, and decision variables of the mixed-integer mathematical model are expounded as follows:

Indices

$k, k'$: Operation index

$j, j'$: Job index

$i, i'$: Machine index

Parameters

N: Set of jobs

M: Set of machines

$O_j$: Set of operations in $j$th job

$o_{jk}$: The $k$th operation in $j$th job

$p_{sjk}$: Processing time of $o_{jk}$ operation on $i$th machine

$M_{jk}$: Set of possible machines for $o_{jk}$; $M_{jk} \in M$

$Q_{jk}$: Set of ordered pairs including operations from $j$th job with no interplay, i.e. $<k, k'> \in Q_j$ if operation $k$ has no precedence relation with $k'$

$P_{jk}$: Set of ordered pair clearing interplay between operations from $j$th job, i.e. $<k, k'> \in P_j$ if operation $k$ should precede operation $k'$

L: A big number

Decision variables

$y_{ijk}$: 1 if operation $o_{jk}$ completed on $i$th machine, Otherwise 0

$st_{ijk}$: Starting time for operation $o_{jk}$ on $i$th machine

$ft_{ijk}$: Finishing time for operation $o_{jk}$ on $i$th machine

$x_{ijk}$: 1 if operation $k'$ of job $j'$ is scheduled after operation $k$ of job $j$ where both operations are processed on machine $i$, otherwise 0

$z_{jkl}$: 1 if operation $o_{jkl}$ carried out after $o_{jkl}$, otherwise 0

Goals

$C_{\text{Max}}$: Makespan

CWL: Critical machine workload

TWL: Machines total workload
The objective function and constraints are as follows:

\[
\begin{align*}
\text{Min } Z &= \alpha_1 C_{\text{max}} + \alpha_2 \text{CWL} + \alpha_3 \text{TWL} \\
C_{\text{max}} &\geq f_{ijk} \forall j \in N, \forall k \in O_j \\
\text{CWL} &\geq \sum_{j=1}^{n} \sum_{k \in O_j} y_{ijk} p_{ijk} \forall i \in M \\
\text{TWL} &= \sum_{j=1}^{n} \sum_{k \in O_j} \sum_{i \in M_j} y_{ijk} p_{ijk} \\
\text{st}_{ijk} - f_{ijk} + L \cdot x_{ijk} &\geq 0 \forall j, j' \in N, j' < j, \forall k \in O_j, \forall k' \in O_{j'}, \forall i \in M_{jk} \cap M_{jk'} \\
\text{st}_{ijk'} - f_{ijk'} + L \left(1 - x_{ijk} x_{ijk'}\right) &\geq 0 \forall j, j' \in N, j' < j, \forall k \in O_j, \forall k' \in O_{j'}, \forall i \in M_{jk} \cap M_{jk'} \\ &\sum_{\forall i \in M_{jk}} \text{st}_{ijk} - \sum_{\forall i \in M_{jk'}} f_{ijk'} + L \cdot z_{jk} &\geq 0, \forall j \in N, \forall (k, k') \in Q_j \\
\sum_{\forall i \in M_{jk'}} \text{st}_{ijk'} - \sum_{\forall i \in M_{jk}} f_{ijk} + L \left(1 - z_{jk'}\right) &\geq 0, \forall j \in N, \forall (k, k') \in Q_j \\
\sum_{\forall i \in M_{jk}} \text{st}_{ijk} - \sum_{\forall i' \in M_{jk'}} f_{ijk'} &\geq 0, \forall j \in N, \forall (k, k') \in P_j 
\end{align*}
\]
As Equation (1) presents, the objective function is considered as a weighted sum each of which are defined in Equations (2)–(4). The first one is after reducing the total job processing time (identified as makespan in literature) and is the most common criterion in the scheduling problem. The second one and the third are machines critical load and machines total load, respectively and their minimization tends towards proper distribution of jobs on machines and reduces the bottleneck on a specific resource. The triple goals weights are determined by other practitioners through their empirical experiences. According to Xing,
Chen, and Yang (2009), makespan, machines critical load, and machines total load priorities are considered as high, average, and small with digits of 0.5, 0.3, and 0.2:

\[ Z = 0.5 C_{\text{max}} + 0.3 \text{CWL} + 0.2 \text{TWL} \]

Constraints (5) and (6) are the impediment time overlap of two operations allocated to a machine. Constraints (7) and (8) declare that two operations related to a job have no overlap. Constraint (9) arranges operation \( k' \) after operation \( k \) providing that \( k \) is predecessor for \( k' \). Constraint (10) guarantees each machine to do one operation. Constraint (11) nulls the non-allocated machines’ operation time. Constraint (12) explains the relation between starting time and finishing time of the operations. Constraints 13–17 are devised for Boolean and non-negative variables.

5. Proposed algorithms and solving procedure

Regarding the NP-Hard type of the problem, for solving it in a large-scale, GA-based algorithms are proposed as considered herein. Regarding the fact that multiple operation sequences considering precedence constraints that are connotationally deployed result in extended searching space, the hallmark which is fitted with genetic algorithm characteristics in parallel and random searching provides capabilities for obtaining the optimum results. In a recently ad-hock review on the corresponding field of study conducted by Ausaf et al. (2014), about 50 works were considered and concluded that the genetic algorithm is the most efficient and popular algorithm for the IPPS problems.

**Procedure: Typical GA**

**Begin**

- **Step 1.** Determine population size (nPop), crossover rate (pc), mutation rate (mu), and iteration numbers (max-it).
- **Step 2.** Randomly generate first population.
- **Step 3.** Calculate their fitness function and sort individuals.
- **Step 4.** Update the best answer.
- **Step 5.** Consider termination conditions.
  - **5.1.** IF the condition is satisfied, OTHERWISE generation = 1 + generation.
- **Step 6.** Deploy crossover and mutation operators on parents for generating offspring.
- **Step 7.** Calculate fitness function of parents and offspring and sort them.
- **Step 8.** Back to step 4.

**END**

The motivation of such an approach is stemmed in its capabilities for searching and reaching the global optimum point or a point near to this.

**The Proposed GA**

**Begin**

- **Step 1.** Put \( l = 0 \) and set \( \text{SO}(l) \) with all of the operations without predecessor.
- **Step 2.** Determine \( f^* := \min \{ f \mid o_{m} \in \text{SO}(l) \} \), if \( o_{m}^* \) is the corresponding operation with \( f^* \), the dedicated machine and related job to the operation \( o_{m}^* \) are symbolized as \( m^* \) and \( j^* \).
- **Step 3.** Set the \( \text{CO}(l) \) as \( \text{CO}(l) := \{ o_{p} \in \text{SO}(l) \mid \{ s_{p} < f^* \} \land (m_{l} = m^* \lor j = j^*) \} \).
- **Step 4.** Select \( o_{m}^* \) which has the most priority (according to the chromosome 1st strain) from \( \text{CO}(l) \), and plan its starting in \( s_{m}^* \).
- **Step 5.** Set the \( \text{SO}(l+1) \) with eliminating \( o_{m}^* \) from \( \text{SO}(l) \) and adding successors to \( o_{m}^* \).
- **Step 6.** Put \( l:=l+1 \), if \( \text{SO}(l) = \emptyset \) go to step 2, otherwise halt.

**END**
Where:

\[ \text{SO}(l) \] set of schedulable operation at 1st step

\[ \text{CO}(l) \] set of candidate operation for scheduling at 1st step

\[ s_{ojk} \] The soonest possible for starting operation \( o_{jk} \in \text{SO}(l) \)

\[ f_{ojk} \] The soonest possible for finishing operation \( o_{jk} \in \text{SO}(l) \)

\[ m_{jk} \] Dedicated machine to operation \( o_{jk} \) (according to 2nd strain of the chromosome)

### 5.1. Characteristics of the algorithms

In this section, common characteristics for deployed algorithms as their Initial Population Generation, Selection, Crossover, Mutation, Termination Conditions, and Parameters Calibration are going to be considered.

#### 5.1.1. Solution representation and encoding

An impetus point in GA-based algorithms efficiency is the proper encoding system for the solution representation cause of interplay between encoding the solutions into the genotype environment and genetic operators. Here, complying with the problem semantics, a two strain chromosome for representation is deployed. The length of each strain is equal to the Total Number of Operations (TNO). Job number 1 to job number \( n \) are numbered from 1 to \( n \). The problem declared as Figure 2, 5 delegates operation 1 from job 2. As a case, the 2nd operation from the 1st job which ought to process on machine 1 has the highest priority in scheduling and the 3st operation from the 3nd job on 2nd machine caught the next priority.

#### 5.1.2. First generation

In fact, this is the first seed of the algorithm. GA starts searching in the feasible area relying on these solutions. Generally, in meta-heuristic algorithms, the first generation as a starter point is important about how they are provided. Herein, they are produced randomly to prevent the convergence ahead of the expectations.

#### 5.1.3. Selection operator

The most common and efficient method for selecting the next generation in GA is considered as Roulette Wheel and because of its efficiency in the previous related works Rabiee, Zandieh, and Ramezani (2012), the proposed algorithm uses this as its selection operator.

#### 5.1.4. Crossover

Two kinds of single-point and two-point crossover are randomly used on parent chromosomes. Figure 3 represents a case of two-point crossover.

![Figure 2. Solution representation.](image)
5.1.5. **Chromosomes’ repairing**

After using crossover, chromosomes may repair again and doing so the genes would be considered from the 1st to the last part and would substitute in cases of repetition or infeasible ones. If the devoted machine is infeasible for the substitute operation, one of them would randomly be dedicated. Figure 4 shows a case of chromosome repairing.

5.1.6. **Mutation**

Using this operator, some of the solutions could be selected randomly and according to the mutation rate, and then one of the mutation methods as complete converser, partial converser, or replacement could be done on the selected chromosomes. Figures 5–7 describe each of the mentioned methods, respectively.

5.1.7. **Chromosomes’ repairing**

In the mutation operator, because of genes are just replaced as a result only precedence relationships condition between operations have done for chromosomes’ repairing.

5.1.8. **Decoding**

A decoding procedure is developed to convert each chromosome into a unique active schedule. This procedure is an extension of the Giffler and Thompson algorithm which was originally proposed to generate active schedules for a JSP (Baker & Trietsch, 2013). Table 3 represents the output of the algorithm.

![Figure 3. A case of crossover.](image)

![Figure 4. Chromosomes repairing after crossover.](image)

![Figure 5. Complete inverse mutation.](image)
The following Gantt chart represents operation decoding through 9 iterations. As exposed in the first iteration, the second operation from the first job is scheduled and continued to the ninth operation (Figure 8).

The completed Gantt chart of problem number 1 obtained final result is represented in Figure 9. The selected sequence for job number 1 is in the 2nd order and for job number 3, it is in the 3rd order (makespan = 24, Objective function = 29.8).

The disjunctive graph can be used to represent the feasible solutions of scheduling problems. $G = (N, A, E)$ is composed by nodes (N), conjunctive arcs (A), and disjunctive arcs (E).
Conjunctive arcs connect two consecutive operations of the same job based on the processing route of the job, and disjunctive arcs define the sequence of operations on each machine. The weight of each node is displayed by the processing time of the corresponding operation. In the following figure, disjunctive graph of the mentioned problem is displayed.

5.1.9. Fitness function

Fitness functions are criteria according to which their optimum values are an active set of timely program (Baker & Trietsch, 2013). Thus, in order to determine the solutions fitness values, transforming information from two strain chromosomes and then the final objective function value could be calculated accordingly. According to the solution relocation in terms of two-strain chromosomes, the different chromosomes after running the mentioned algorithm may result in the same schedule. To reduce the genetic operator computations, after evaluating the fitness for each solution, the priority of each operation will change according to the starting time and the operation in active plan which started sooner would have the higher priority.

Procedure: Fitness Evaluation

Input data:

- Let the $SPT_i (i = 1, \ldots, m)$ be the sum of processing time of resource $i$;
- Let the $Stij$ be the start time of the operation $k$ of order $j$ on resource $i$;
- Let the $Ko$ be the total number of operations of TO;
- Let the $Ftijk$ be the end time of operation $k$ of order $j$ on resource $i$;
- Let the $PTOijk$ be the processing time of operation $k$ of order $j$ on resource $i$;

Begin

$SPT_i = 0$ for all $i$;
$Stij = 0$ for all $i, j, k$;
For $j = 1$ to $I$
  For $k = 1$ to $Ko$
    $i \leftarrow$ resource number selected by operation $k$;
    $Stij = SPT_i$;
    $Ftijk = Stij + PTOijk$;
    $SPT_i = Ftijk$
  End
End

$Makespan = \max_{i=1, \ldots, M} (Ft_{ij})$;
Machines total workload $= \sum (SPT_i)$;
Critical machine workload $= \max (SPT_i)$;
Return $Makespan$, Machines total workload, Critical machine workload;

END
5.1.10. **Termination condition**

As it is clear by its name, it determines the circumstance for halting the iteration. For applied algorithms, the termination condition in this paper is a specific number of iterations dictated by the calibration method explained in next section.

5.1.11. **Parameter calibration**

There are various methods to calibrate the meta-heuristic algorithm parameters, some of which are full factorial design, i.e. they examine all possible combinations (Montgomery, 2008; Ruiz, Maroto, & Alcaraz, 2006) that are intrinsically time and cost consuming. The Taguchi method (1986) uses a special design of orthogonal arrays to study the entire parameters space with a small number of experiments.

The Taguchi method clusters factors into two main groups: controllable and noise factors (uncontrollable). Since noises of factors are uncontrollable, their elimination is unpractical and almost impossible, and the Taguchi method tries to reach the best controllable factors level from a robustness point of view. In addition to determine the best factors level, Taguchi establishes the relative importance of each factor with respect to its main impacts on the objective function (Naderi et al., 2009). To analyze the experimental data and find optimal factor combination, the Taguchi method uses a criterion entitled signal-to-noise (S/N) ratio which is expected to be maximum.

![Disjunctive graph for Table 1](image)

**Figure 10.** Disjunctive graph for Table 1.

| Test | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Iteration-1 | 4546.3 | 4514.95 | 4463.5 | 4462.65 | 4502.15 | 4611.25 | 4514.3 | 4462 | 4530.25 |
| Iteration-2 | 4470.25 | 4506.8 | 4421.8 | 4487.95 | 4561.85 | 4504.7 | 4452.5 | 4477.5 | 4581.75 |
| Iteration-3 | 4508.05 | 4555.4 | 4509.15 | 4556.3 | 4549.6 | 4502.775 | 4428.55 | 4567.55 | 4488.65 |
| Iteration-4 | 4572.2 | 4492.8 | 4471.75 | 4597.2 | 4495.95 | 4512.25 | 4545.25 | 4497.05 | 4536.05 |
| Iteration-5 | 4479.95 | 4464.3 | 4428.55 | 4480.85 | 4574.75 | 4523 | 4517.4 | 4530.9 | 4549.7 |
| Iteration-6 | 4578.9 | 4468.45 | 4451.4 | 4558.5 | 4550.3 | 4546.3 | 4502.45 | 4430.9 | 4359.7 |
| Iteration-7 | 4499.55 | 4564.2 | 4488.3 | 4508.45 | 4474.85 | 4461.75 | 4498.9 | 4489.4 | 4602.7 |
| Iteration-8 | 4510.5 | 4471.8 | 4521.3 | 4561.2 | 4483.85 | 4386.25 | 4500.25 | 4560.4 | 4473.75 |
| Iteration-9 | 4491.3 | 4498.5 | 4469.25 | 4583.25 | 4470.6 | 4570.2 | 4466.3 | 4437.2 | 4488.25 |
| Iteration-10 | 4445.3 | 4401.6 | 4428.7 | 4534 | 4478.55 | 4544.45 | 4369.7 | 4539.2 | 4477.55 |
| Mean | 4510.23 | 4492.08 | 4465.37 | 4533.035 | 4514.245 | 4507.293 | 4479.29 | 4489.21 | 4508.835 |
| Min | 4445.3 | 4401.6 | 4421.8 | 4462.65 | 4470.6 | 4386.25 | 4369.7 | 4430.9 | 4359.7 |
The appropriate values for the proposed algorithm are obtained by the Taguchi method (Candan & Yazgan, 2015). Doing so, according to Table 4 and through the experiments for each operator as the mutation rate, crossover rate, population number, and iteration number 3 levels are considered and the values 0.4, 0.9, 50, and 300 are obtained, respectively.

Implementing the Taguchi method has come in action using Minitab17. The Taguchi experiments for each of these algorithms give the S/N ratio and mean criteria for the proposed algorithms separately. The result of GA are exposed in Figure 11.

Using the exposed diagram and after complimentary experiments, the selected levels for parameters are presented in Table 5 and underline values are optimized.

In this study, NSGAII is used due to get the answer on the Pareto algorithm. In this algorithm, used operator in proposed GA is employed. The results of the problem with large, medium, and small scales have shown in Table 9.

Figure 11. Mean of means and S/N ratio for each parameter.
6. The experiments

In order to evaluate the proposed algorithms in the mentioned MOFJSP-APP problem, the case problems in previous works are deployed. Since the multi-objective circumstance is unprecedented for this problem, the FJSP-APP is deployed as the closest problem to the context of this work deployed. Four numbers of the works according to which their required data were accessible have been used. After modifying them from single-objective to multi-objective and by considering makespan, critical machine workload, and machines total workload, the final obtained results using the proposed algorithm (GA) are presented in Table 6 and compared by the results from LINGO. Reaching an optimum solution, a 15-min time limit is deliberated. The obtained results in this predefined duration prove the efficiency of the algorithm. Likewise, these kinds of problems are solved by GA according to which their results have deviation from the optimum solution through a fraction of a second.

To calculate the proposed GA deviation in comparison with the exact method, the following formula is deployed. Output results in Table 6; narrows to zero convergence in the proposed GA and prove its efficiency. Due to NP-Hardness of problem, lingo software does not solve problems with medium and large scales in an acceptable time so have not been compared. Figures 12 and 13 represent the comparison results between the exact method and the proposed GA regarding processing time and result quality. As declared, the proposed GA is efficient enough from the processing time point of view and is near the exact method from the quality point of view (as obtained in problems P2, P10, P11). The star sign (*) shows that this problem has not result in the acceptable time.

\[
\text{ERR} = \frac{\text{Genetic Result} - \text{Optimum}}{\text{Optimum}}
\]

Considering the algorithm capability in a large-scale, the two aforementioned cases with variation in demand for each order are adopted and eight new problems are created (Mosahar, Fazel Zarandi, & Türksen, 2014). The P5 problem which is constituted from 8 orders and 4 machines, is transferred into 80, 160, 240, and 400 orders, and problem P8 that includes 6 orders with 5 machines is changed into 60, 120, 180, and 300 orders by increasing demands and number of machines is limitation for example a factory that faced with the high volume of orders. Table 7 represents specifications and results of these problems. For each problem, the proposed GA is run ten times and as declared the small deviation of the best solution in comparison with the average of objective function value through these ten runs proves the stability of the proposed algorithm in obtaining an appropriate solution.

---

**Table 5. Parameters and their level via Taguchi calibration method.**

| Level | Maximum iteration | Population size | Crossover rate | Mutation rate |
|-------|-------------------|-----------------|---------------|--------------|
| 1     | 100               | 40              | 0.7           | 0.1          |
| 2     | 300               | 50              | 0.8           | 0.2          |
| 3     | 400               | 60              | 0.9           | 0.4          |

---
Table 6. Obtained solution in small and medium-scale instances.

| Adopted from                  | Problem | Job numbers | Machine numbers | Operation numbers | Variable numbers | Constrain numbers | LINGO Objective Function | Genetic Objective Function | LINGO Processing time | Proposed GA Processing time | Genetic algorithm Error |
|-------------------------------|---------|-------------|-----------------|-------------------|------------------|-------------------|--------------------------|--------------------------|------------------------|---------------------------|--------------------------|
| Mohapatra et al. (2015) P1    | 4       | 4           | 14              | 321               | 554              | 118.5             | 118.9                    | 40                       | 6                      | 0.003376                  |
| Mohapatra et al. (2015) P2    | 2       | 4           | 14              | 289               | 511              | 140.6             | 140.6                    | 45                       | 12.62                  | 0                         |
| Mohapatra et al. (2015) P3    | 2       | 4           | 14              | 257               | 456              | 165.6             | 166.1                    | 60                       | 13.5                   | 0.003019                  |
| Mohapatra et al. (2015) P4    | 2       | 4           | 14              | 283               | 508              | 152.75            | 152.85                   | 180                      | 12.7                   | 0.000655                  |
| Mohapatra et al. (2015) P5    | 8       | 4           | 48              | 3146              | *                | 430.25            | *                        | 12.54                    | *                      | *                         |
| Mohapatra et al. (2015) P6    | 2       | 4           | 6               | 97                | 146              | 49.1              | 50.1                     | 3                        | 3                      | 0.020367                  |
| Mohapatra et al. (2015) P7    | 2       | 4           | 7               | 118               | 183              | 89.6              | 899                      | 7                        | 3.27                   | 0.003348                  |
| Li et al. (2010) P8           | 6       | 5           | 18              | 293               | 474              | *                 | 29.8                     | *                       | 6.254                  | *                         |
| Nasr and Elsayed (1990) P9    | 4       | 6           | 12              | 194               | 309              | 20.4              | 20.5                     | 3                        | 2.55                   | 0.004902                  |
| Li et al. (2010) P10          | 5       | 5           | 20              | 99                | 170              | 21.3              | 21.3                     | 2                        | 5.88                   | 0                         |
| Moon, Lee, and Bae (2008) P11 | 5       | 5           | 13              | 99                | 170              | 21.3              | 21.3                     | 5                        | 2                      | 0                         |
In this section, the problem Moon (P11) with 5 jobs each including 2, 2, 3, 2, and 4 is considered. There are precedence relations between operations of a specific job. Table 8 exposed possible sequences and inter-relations between these operations. Meanwhile, the possible machines for each operation are represented. After running the algorithm, the possible plans pronounced. Figure 14 shows the related Gantt chart for this problem. The sameness of the
The proposed algorithm’s result and exact method result (by LINGO) proves the efficiency of the proposed algorithm. After running the algorithm, the selected processes from the set are pronounced in the figure below. The result of the exact method and proposed algorithm is declared in Tables 6 and 9.

6.2. Second experiment

In this section, the problem P5 (middle scale) includes 8 jobs with 7, 7, 10, 4, 3, 3, and 4 operations respectively, that are to be scheduled on 4 machines. Table 10 described

Table 8. Data set 2 for FJSP-APP problem.

| Operation | M1 | M2 | M3 | M4 | M5 | Precedence graph | Feasible process plans |
|-----------|----|----|----|----|----|------------------|------------------------|
| Job 1     | O11| 5  | 3  | –  | –  | –                | O13 – O12              |
|           | O12| –  | 5  | –  | –  | –                |                        |
| Job 2     | O21| –  | –  | 6  | 5  | –                | O23 – O21              |
|           | O22| –  | –  | –  | –  | 4                |                        |
| Job 3     | O31| 5  | 4  | –  | –  | –                | O33 – O32              |
|           | O32| –  | –  | 2  | 3  | –                |                        |
|           | O33| –  | 5  | –  | –  | –                |                        |
| Job 4     | O41| –  | –  | 4  | –  | –                | O43 – O42              |
|           | O42| –  | –  | –  | 5  | –                |                        |
| Job 5     | O51| 4  | 3  | –  | –  | –                | O53 – O52              |
|           | O52| 2  | 4  | –  | –  | –                |                        |
|           | O53| –  | 5  | –  | –  | –                |                        |
|           | O54| –  | 4  | 3  | –  | –                |                        |

Figure 14. Gantt chart for data set.
the problem characteristics. Figure 15 shows precedent constraint between the operations. The Gantt chart of problem is presented in Figure 16. Disjunctive graph of this data set is displayed in Figure 17. To run the algorithm in large scale with demand variation, the scale of this problem and problem P8 are changed and while only the numbers of problem jobs are ten times bigger, other characteristics (machine numbers) remained the same. Regarding being NP-Hard and its amplification with an increase in problem size, LINGO did not turn a feasible result neither in middle-scale (8 jobs), nor in large-scale (80 jobs).

In the following table, the results of proposed algorithms are displayed so that objectives have been brought separately. The NSGAII approximately is more efficient in each of the goals.

### Table 9. Comparison between WGA & NSGA II.

| Problem | f1   | f2   | f3   | WGA f1 | f2   | f3   |
|---------|------|------|------|--------|------|------|
| P1      | 85   | 84   | 256  | 118.9  | 83   | 255  |
| P2      | 127.5| 89.5 | 250  | 140.6  | 123  | 246  |
| P3      | 176  | 94   | 249.5| 166.1  | 177.5| 250.5|
| P4      | 151  | 88   | 254.5| 152.8  | 152.75| 237.75|
| P5      | 323  | 308  | 882  | 430.3  | 290  | 841.5|
| P6      | 46   | 31   | 89   | 50.1   | 45   | 88   |
| P7      | 92   | 33   | 140  | 89.9   | 82   | 130  |
| P9      | 15   | 10   | 50   | 20.5   | 17   | 10   |
| P10     | 16   | 11   | 50   | 21.3   | 33   | 24   |
| P11     | 14   | 13   | 52   | 21.3   | 14   | 13   |
| EP5-10  | 3520 | 3520 | 9409 | 4697.8 | 2891.25| 2867 |
| EP5-20  | 7340 | 7245 | 19270| 9697.5 | 6464.75| 6420 |
| EP5-30  | 11300| 11090| 28000| 14577  | 9581.5| 9538 |
| EP5-50  | 17370| 17329| 45760| 23035.7| 16495 | 16465|
| EP8-10  | 270  | 265  | 1109 | 436.3  | 223  | 1055 |
| EP8-20  | 447  | 447  | 2151 | 787.8  | 458  | 2114 |
| EP8-30  | 900  | 800  | 3315 | 1353   | 721  | 3189 |
| EP8-50  | 1390 | 1275 | 5905 | 2258.5 | 1252 | 5361 |

### Table 10. Operation information for data set 3.

| Operation number | Precedence | Substitute machines | Processing time |
|------------------|------------|---------------------|-----------------|
| o₁₁              |            | M₁, M₂, M₃, M₄     | 40–40–30        |
| o₁₂              | o₁₁        | M₁, M₂, M₃, M₄     | 40–40–30        |
| o₁₃              |            | M₁, M₂, M₃, M₄     | 20–20–15        |
| o₁₄              | o₁₂, o₁₃   | M₁, M₂, M₃, M₄     | 12–10–10–7      |
| o₂₁              |            | M₁, M₂, M₃, M₄     | 20–10–10–7.5    |
| o₂₂              | o₂₁        | M₁, M₂, M₃, M₄     | 20–20–15        |
| o₂₃              | o₂₁        | M₁, M₂, M₃, M₄     | 18–18–13.5      |
| o₂₄              | o₂₁, o₂₃   | M₁, M₂, M₃, M₄     | 20–20–15–15     |
| o₃₁              |            | M₁, M₂, M₃, M₄     | 20–20–15        |
| o₃₂              | o₃₁, o₃₃   | M₁, M₂, M₃, M₄     | 15–15–11.25     |
| o₃₃              |            | M₁, M₂, M₃, M₄     | 12–15           |
| o₃₄              |            | M₁, M₂, M₃, M₄     | 21–18           |
| o₄₁              | o₄₂        | M₁, M₂, M₃, M₄     | 18–25           |
| o₄₂              | o₄₁        | M₁, M₂, M₃, M₄     | 27–25           |
Figure 15. Precedence relations between the operations for Test Case 3.
Figure 16. Gantt chart for Test Case 3.
Figure 17. Disjunctive graph for Table 8.
7. Concluding remarks and implications for future works

This paper considered the multi-objective flexible job-shop problem with an undeniable application for services as well as the production industries. As mentioned, this problem includes a number of orders each of which were constituted from a bunch of operations. Operations ought to be scheduled on machines, providing that these operations could be processed on multiple machines. The sequence of operation may or may not be linear and has been exposed as a predecessor-successor graph. Thus, it results in flexibility in the production plan beside the issue on machine. Such a fact is considered as integration of the process plan and scheduling problems. The problem is formulated as the concurrent minimization of makespan and machines workload. Then a number of pronounced problems in the literature are embodied in the proposed mathematical model and the obtained optimum solution in a feasible time proves the efficiency of the model.

To solve the proposed problem, the GA-based algorithms are developed according to which theirs operators have adjusted complying with the problem specifications. Through the proposed methods, besides deploying the GA capabilities in global search, the produced generation will remain feasible in mutation from one generation into the next. Performance of the proposed algorithms from the time and quality point of view according to some large-scale problems are evaluated and proven.

Dynamism and contingencies of the workshop circumstances and environmental effects on the production system are indispensable principles of today’s production and services systems. Thus, as an opportunity for future study considering the dynamism or/and contingencies and proposing a solution for them is noteworthy. Variation in the number and quality of the accessible machines, orders with priority, utilities’ malfunction and tardiness, changes in operation sequences, changes of order ingredients combination through the process are other research interests in this area. Likewise, regarding the varieties of entrepreneurs’ requirements, developing optimization methods with other criteria in production planning systems as the number of tardiness, total cost, average of inventories, and number of work-in-process using the appropriate algorithm could be taken into account as other cases to continue this work.

Disclosure statement

No potential conflict of interest was reported by the author.

References

Alavidoost, M. H., Tarimoradi, M., & Zarandi, M. H. F. (2015a). Fuzzy adaptive genetic algorithm for multi-objective assembly line balancing problems. Applied Soft Computing, 34, 655–677.
Alavidoost, M., Tarimoradi, M., & Zarandi, M. F. (2015b). Bi-objective mixed-integer nonlinear programming for multi-commodity tri-echelon supply chain networks. Journal of Intelligent Manufacturing, 1–18.
Alavidoost, M., Zarandi, M. F., Tarimoradi, M., & Nemati, Y. (2014). Modified genetic algorithm for simple straight and U-shaped assembly line balancing with fuzzy processing times. Journal of Intelligent Manufacturing, 28(2), 1–24.
Ausaf, M. F., Li, X., & Gao, L. (2014). Optimization algorithms for integrated process planning and scheduling problem-A survey. In 2014 11th World Congress on Intelligent Control and Automation (WCICA). IEEE.
Baker, K. R., & Trietsch, D. (2013). *Principles of sequencing and scheduling*. Hoboken, NJ: Wiley.

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research, 41*(3), 157–183.

Brandimarte, P., & Calderini, M. (1995). A hierarchical bicriterion approach to integrated process plan selection and job shop scheduling. *The International Journal of Production Research, 33*(1), 161–181.

Candan, G., & Yazgan, H. R. (2015). Genetic algorithm parameter optimisation using Taguchi method for a flexible manufacturing system scheduling problem. *International Journal of Production Research, 53*(3), 897–915.

Chabane, A., Benyoucef, L., & Tiwari, M. K. (2012). An adapted NSGA-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system. *Journal of Intelligent Manufacturing, 23*(4), 1141–1155.

Chen, Q., & Khoshnevis, B. (1993). Scheduling with flexible process plans. *Production Planning & Control, 4*(4), 333–343.

Chryssolouris, G., Chan, S., & Cobb, W. (1984). Decision making on the factory floor: An integrated approach to process planning and scheduling. *Robotics and Computer-Integrated Manufacturing, 1*(3–4), 315–319.

Deb, K., & Goel, T. (2001). *Controlled elitist non-dominated sorting genetic algorithms for better convergence in Evolutionary multi-criterion optimization*. Berlin: Springer.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Delorme, X., Dolgui, A., & Kovalyov, M. Y. (2012). Combinatorial design of a minimum cost transfer line. *Omega, 40*(1), 31–41.

Demir, Y., & Kırsat İşleyen, S. (2013). Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling, 37*(3), 977–988.

Dolgui, A., Gordon, V., & Strusevich, V. (2012). Single machine scheduling with precedence constraints and positionally dependent processing times. *Computers & Operations Research, 39*(6), 1218–1224.

Dolgui, A., Guschinsky, N., & Levin, G. (2009). Graph approach for optimal design of transfer machine with rotary table. *International Journal of Production Research, 47*(2), 321–341.

Giffler, B., & Thompson, G. L. (1960). Algorithms for solving production-scheduling problems. *Operations Research, 8*(4), 487–503.

Gologlu, C. (2004). Machine capability and fixturing constraints-imposed automatic machining set-ups generation. *Journal of Materials Processing Technology, 148*(1), 83–92.

Guo, Y., Li, W. D., Mileham, A. R., & Owen, G. W. (2009). Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing, 25*(2), 280–288.

Huang, S., & Xu, N. (2003). Automatic set-up planning for metal cutting: An integrated methodology. *International Journal of Production Research, 41*(18), 4339–4356.

Huang, S. H., Zhang, H.-C., & Smith, M. L. (1995). A progressive approach for the integration of process planning and scheduling. *IEEE Transactions, 27*(4), 456–464.

Jain, A., Jain, P., & Singh, I. (2006). An integrated scheme for process planning and scheduling in FMS. *The International Journal of Advanced Manufacturing Technology, 30*(11–12), 1111–1118.

Kim, Y. K., Park, K., & Ko, J. (2003). A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers & Operations Research, 30*(8), 1151–1171.

Kühnle, H., Braun, H. J., & Bührling, J. (1994). Integration of CAPP and PPC. *Interfusion Manufacturing Management, 5*(2), 21–27.

Kumar, M., & Rajotia, S. (2003). Integration of scheduling with computer aided process planning. *Journal of Materials Processing Technology, 138*(1–3), 297–300.

Larsen, N. E., & Alting, L. (1992). Dynamic planning enriches concurrent process and production planning. *The International Journal of Production Research, 30*(8), 1861–1876.

Lee, H., & Kim, S. S. (2001). Integration of process planning and scheduling using simulation based genetic algorithms. *The International Journal of Advanced Manufacturing Technology, 18*(8), 586–590.
Leung, C. W., Wong, T. N., Mak, K. L., Mak, K. L., & Fung, R. Y. K. (2010). Integrated process planning and scheduling by an agent-based ant colony optimization. *Computers & Industrial Engineering*, 59(1), 166–180.

Li, X., Gao, L., & Li, W. (2012). Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling. *Expert Systems with Applications*, 39(1), 288–297.

Li, X., Gao L., Zhang, C., & Shao, X. (2010). A review on integrated process planning and scheduling. *International Journal of Manufacturing Research*, 5(2), 161–180.

Lian, K., Zhang, C., & Gao, L. (2012). Integrated process planning and scheduling using an imperialist competitive algorithm. *International Journal of Production Research*, 50(15), 4326–4343.

Lihong, Q., & Shengping, L. (2012). An improved genetic algorithm for integrated process planning and scheduling. *The International Journal of Advanced Manufacturing Technology*, 58(5–8), 727–740.

Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, 8 (2), 219–223.

Min, L., Li, B., & Shensheng, Z. (2004). Modeling integrated CAPP/PPS systems. *Computers & Industrial Engineering*, 46(2), 275–283.

Mohammadi, G., Karampourhaghghi, A., & Samaei, F. (2012). A multi-objective optimisation model to integrating flexible process planning and scheduling based on hybrid multi-objective simulated annealing. *International Journal of Production Research*, 50(18), 5063–5076.

Mohapatra, P., Nayak, A., Kumar, S. K., & Tiwari, M. K. (2015). Multi-objective process planning and scheduling using controlled elitist non-dominated sorting genetic algorithm. *International Journal of Production Research*, 53(6), 1712–1735.

Montgomery, D. C. (2008). *Design and analysis of experiments*. Hoboken, NJ: Wiley.

Moon, I., Lee, S., & Bae, H. (2008). Genetic algorithms for job shop scheduling problems with alternative routings. *International Journal of Production Research*, 46(10), 2695–2705.

Mosahar, T., Fazel Zarandi, M. H., & Türksen, I. B. (2014). Hybrid intelligent agent-based internal analysis architecture for CRM strategy planning. *International Journal of Intelligent Systems and Applications*, 6(6), 1–20.

Moslehi, G., & Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129(1), 14–22.

Naderi, B., Zandieh, M., Balagh, A. K. G., & Roshanaeia, V. (2009). An improved simulated annealing for hybrid workflows with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert Systems with Applications*, 36(6), 9625–9633.

Nasr, N., & Elsayed, E. (1990). Job shop scheduling with alternative machines. *The International Journal of Production Research*, 28(9), 1595–1609.

Ong, S., Ding, J., & Nee, A. (2002). Hybrid GA and SA dynamic set-up planning optimization. *International Journal of Production Research*, 40(18), 4697–4719.

Özgüven, C., Özbakır, L., & Yavuz, Y. (2010). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling*, 34(6), 1539–1548.

Rabiee, M., Zandieh, M., & Ramezani, P. (2012). Bi-objective partial flexible job shop scheduling problem: NSGA-II, NRGA, MOGA and PAES approaches. *International Journal of Production Research*, 50(24), 7327–7342.

Rajkumar, M., Asokan, P., & Page, T. (2010). A GRASP algorithm for the integration of process planning and scheduling in a flexible job-shop. *International Journal of Manufacturing Research*, 5(2), 230–251.

Ruiz, R., Maroto, C., & Alcaraz, J. (2006). Two new robust genetic algorithms for the flowshop scheduling problem. *Omega*, 34(5), 461–476.

Saygin, C., & Kilic, S. E. (1999). Integrating flexible process plans with scheduling in flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 15(4), 268–280.

Sen, S., & Jain, S. (2010). *Integration of process planning and scheduling comparison of models approach*. Shao, X., Li, X., Gao, L., & Zhang C. (2009). Integration of process planning and scheduling – A modified genetic algorithm-based approach. *Computers & Operations Research*, 36(6), 2082–2096.
Sobeyko, O., & Mönch, L. (2017). Integrated process planning and scheduling for large-scale flexible job shops using metaheuristics. *International Journal of Production Research, 55*(2), 392–409.

Taguchi, G. (1986). *Introduction to quality engineering: Designing quality into products and processes*. Tan, W., & Khoshnevis, B. (2000). Integration of process planning and scheduling – A review. *Journal of Intelligent Manufacturing, 11*(1), 51–63.

Tarimoradi, M., Alavidoost, M., & Zarandi, M. F. (2015). Comparative corrigendum note on papers “Fuzzy adaptive GA for multi-objective assembly line balancing” continued “Modified GA for different types of assembly line balancing with fuzzy processing times”: Differences and similarities. *Applied Soft Computing, 35*, 786–788.

Tarimoradi, M., Zarandi, M. H. F., Zaman, H., & Turksan, I. B. (2015). Evolutionary fuzzy intelligent system for multi-objective supply chain network designs: An agent-based optimization state of the art. *Journal of Intelligent Manufacturing, 28*(7), 1–29.

Tran, R., Wu, J., Denison, C., Ackling, T., Wagner, W., & Neumann, F. (2013). Fast and effective multi-objective optimisation of wind turbine placement. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM.

Usher, John M., & Fernandes, K. J. (1996). Dynamic process planning – The static phase. *Journal of Materials Processing Technology, 61*(1–2), 53–58.

Wang, L. (2009). Web-based decision making for collaborative manufacturing. *International Journal of Computer Integrated Manufacturing, 22*(4), 334–344.

Wang, L., Feng, H.-Y., & Cai, N. (2003). Architecture design for distributed process planning. *Journal of Manufacturing Systems, 22*(2), 99.

Wang, J., Zhang, Y. F., Nee, A. Y. C., & Wang, Y. F. (2009). Reducing tardy jobs by integrating process planning and scheduling functions. *International Journal of Production Research*, 47(21), 6069–6084.

Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering, 48*(2), 409–425.

Xing, L.-N., Chen, Y.-W., & Yang, K.-W. (2009). Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. *Applied Soft Computing, 9*(1), 362–376.

Zarandi, M. H. F., & Tarimoradi, M. H. (2015). In *Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American*. IEEE.

Zarandi, M. H. F., & Tarimoradi, M. H. (2015). Fuzzy comparison dashboard for multi-objective evolutionary applications: An implementation in supply chain planning. In *Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American*. IEEE.

Zhang, W., & Fujimura, S. (2012). Multiobjectiv process planning and scheduling using improved vector evaluated genetic algorithm with archive. *IEEJ Transactions on Electrical and Electronic Engineering, 7*(3), 258–267.

Zhang, W., & Gen, M. (2010). Process planning and scheduling in distributed manufacturing system using multiobjective genetic algorithm. *IEEJ Transactions on Electrical and Electronic Engineering, 5*(1), 62–72.

Zhang, H.-C., & Lin, E. (1999). A hybrid-graph approach for automated setup planning in CAPP. *Robotics and Computer-Integrated Manufacturing, 15*(1), 89–100.

Zhang, Y., Saravanan, A., & Fuh, J. (2003). Integration of process planning and scheduling by exploring the flexibility of process planning. *International Journal of Production Research, 41*(3), 611–628.

Zhang, Y., Hu, W., Rong, Y., & Yen, D. W. (2001). Graph-based set-up planning and tolerance decomposition for computer-aided fixture design. *International Journal of Production Research, 39*(14), 3109–3126.

Zhang, G., Shao, X., Li, P., & Gao, L. (2009). An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Computers & Industrial Engineering, 56*(4), 1309–1318.

Zhao, B., Gao, J., Chen, K., & Guo, K. (2015). Two-generation Pareto ant colony algorithm for multi-objective job shop scheduling problem with alternative process plans and unrelated parallel machines. *Journal of Intelligent Manufacturing, 1–16.*