J Intell Manuf (2016) 27:375–388
DOI 10.1007/s10845-014-0870-2

Evolutionary resource assignment for workload-based production scheduling

Ilkyeong Moon · Sanghyup Lee · Moonsoo Shin · Kwangyeol Ryu

Received: 9 January 2013 / Accepted: 7 January 2014 / Published online: 21 January 2014

Abstract In this paper, we propose an evolutionary method with a simulation model for scheduling jobs including operations specified in terms of workload rather than processing time. It is suggested that processing times should be determined according to the number of assigned resources rather than the workload. The simulation model is used to estimate the result of resource allocation in a time horizon based on preselected rules. The evolutionary methods improve a production schedule in terms of compliance with due dates by selecting an alternative resource allocation rule and changing timing constraints. The results of computational experiments show that compliance with due dates improved by as much as 30 % under the modified production schedule over the initial schedule.

Keywords Resource assignment · Scheduling · Simulation · Workload · Genetic algorithm · Tabu search

I. Moon
Department of Industrial Engineering, Seoul National University, Gwanak-gu, Seoul 151-744, Korea

S. Lee
Automation Research Department, Industrial Research Institute, Hyundai Heavy Industries Co. LTD., Ulsan 682-792, Korea

M. Shin
Department of Industrial and Management Engineering, Hanbat National University, San 16-1, Duckmyoung-dong, Yuseong-gu, Daejeon 305-719, Korea

K. Ryu (✉)
Department of Industrial Engineering, Pusan National University, Jangjeon-dong, Geumjeong-gu, Busan 690-735, Korea
e-mail: kyryu@pusan.ac.kr

Introduction

In scheduling problems, a processing time of an individual job or an operation is typically a known parameter. In a deterministic problem, the processing time is considered fixed and known *a priori* with certainty. Moreover, probability functions corresponding to processing times are also known factors in stochastic problems. Generally, in a fully automated production system from which human operations are excluded, the processing times can be relatively easily specified. However, human factors often result in more complex and unpredictable behaviors than generated with the inanimate resources typically associated with traditional scheduling models (Lodree et al. 2009). In some past studies, humans were considered to be the most complex element within a system (Bailey 1996; Matthews et al. 2000). Therefore, if an operation is performed by humans, the processing time varies with the number of operators and the working conditions, making it difficult to predict. In addition, two implications follow: (1) processing times are variable and (2) they are dependent on the human operators involved. Simply speaking, the processing time t_i of an operation i that is performed by h_i operators can be expressed as $t_i = f_i(h_i)$, where f_i is the processing time function for operation i. Because h_i varies according to the working conditions of the human operators, scheduling problems are difficult to resolve.

Labor-intensive industries, such as shipbuilding and construction, where human operators constitute a main source of the work force have a tendency to specify individual jobs in terms of man-hours rather than amount of predefined processing times between start and end points. For example, a shipbuilding site employs many human operators as well as machinery. Some operations (cutting, bending, etc.) are performed by machinery, while others (marking, welding, etc.) are carried out by humans. The amount of man-hours produce a more realistic value for use in scheduling problems than do deterministic or stochastic measures, because man-hours account for variations of resource allocation, and thus, in the case of shipbuilding, the welding operation should be specified in terms of man-hours. However, most researchers focused on scheduling problems based their studies on explicit processing times, and most have not fully explored scheduling problems that specify unit operations in terms of workload (or man-hours). In past studies, "workload" is typically the term preferred over "man-hours" (Alfieri 2009); thus, the term *workload* is interchangeably used with *man-hours* in the following discussion.

The values of workloads must be converted to processing times to obtain the solutions for scheduling problems. For the conversion, the number of human operators should be determined. When the number of human operators is fixed for a specific operation, the processing time of that operation can be easily calculated by dividing the total workload for the operation by the number of operators. However, if the number of human operators is not predefined, the processing time of the operation varies with the number of assigned human resources. Actually, even though operations i and j may comprise identical tasks, the processing times p_i and p_j may differ when varying quantities of resources are involved. As a consequence, for scheduling labor-intensive production, management must effectively predict quantities of human resources to assign to individual jobs.

Previous researchers gave limited consideration of practical resource assignment, which has given rise to a widely documented gap between scheduling theory and practice (Graves 1981; Ford et al. 1987; McKay et al. 1988; Dudek et al. 1992; Olhager and Rapp 1995; McKay et al. 2002). To close the "gap" between scheduling theory and practice (MacCarthy and Liu 1993), in this paper, we propose a simulation model to emulate an actual production, whereby a feasible configuration of resource allocation is projected onto a production schedule. In the problem, operations are specified in terms of workload instead of predefined processing times, and the number of resources that can be applied to each operation is considered as a constraint. Specifically, this scheduling problem pertains to effective resource allocations and operation sequences in the shipbuilding industry.

A quantitative study on scheduling research carried out by Reisman et al. (1997) revealed that out of 184 reviewed papers, only five dealt with realistic production settings. Our proposed simulation model is based on a predefined resource assignment rule; thus, selection of the assignment rule affects the quality of the generated production schedule. Therefore, in addition to finding improved start dates for individual products, we also propose evolutionary methods to select the optimal resource-assignment rule that will improve an initial production schedule.

The remainder of this paper is organized as follows. The relevant literature on the subject is reviewed in "Literature review" section. In "Problem description" section, we define a scheduling problem based on the shipbuilding industry. "Simulation model" section illustrates a simulation model including resource assignment procedures. In "Evolutionary methods" section, we present evolutionary methods, based on the simulation model, to improve production schedules. Finally, in "Computational experiments" section, some computational experiments are presented, which are followed by concluding remarks in "Conclusions" section.

Literature review

Scheduling problems

Although much of the literature deals with scheduling issues in the shipbuilding industry, it mainly focuses on efficient operations, including use of quay cranes, transportation vehicles at ports, and so on (Kim and Lee 1997; Kim and Jung 2006; Kim and Lee 2007; Liang et al. 2009; Tavakkoli-Moghaddam et al. 2009). For a better understanding of scheduling problems in the shipbuilding industry, the reader is referred to Lee et al. (1996, 1997); Cho et al. (1998), and Cho et al. (2001).

In various practical systems, processing times may be controlled by allocating additional resources. After 1980, scheduling problems with controllable processing times have been extensively investigated (Vickson 1980; Nowicki and Zdrzalka 1995; Hoogeveen and Woeginger 2002). The scheduling problem illustrates decisions needed to allocate resources and sequence jobs. The solution is measured by two criteria dependent on (1) the job completion time and (2) the resource consumption cost, respectively. According to Shabtay and Steiner (2007), most of researchers who investigated the scheduling problems assumed use of non-renewable resources such as money, fuel, gas, and electricity. In addition, they have generally assumed that the processing time follows a linear function of resource consumption.

Systems in which both machinery and human operators constrain the production capacity are defined as dual resource constrained (DRC) systems (Xu et al. 2011). Scheduling problems for DRC systems are more complicated than those with a single resource constraint because of the additional interaction between job dispatching and assigning human operators. Therefore, according to Xu et al. (2011), analytical approaches to solving the problems related to a DRC systems are neither feasible nor adequate, but meta-heuristic methods such as genetic algorithms and simulated annealing (ElMaraghy et al. 2000; Chaudhry and Drake 2009) are promising approaches. Most scheduling problems for DRC systems assume the availability of limited renewable

Table 1 Characteristics of scheduling problems

Scheduling problem type	Deterministic problem	Problem with controllable processing times	Problem in DRC systems	Problem investigated in this paper
Resource type	Not available	Non-renewable	Renewable	Renewable
Processing time	Fixed	Controllable	Fixed	Controllable
Solution	– Job sequence	– Job sequence	– Job dispatching	– Job schedule
		– Resource allocation	– Resource allocation (reallocation is allowed)	– Resource allocation (reallocation is allowed)

resources and static processing times. That is, after used in one job, renewable resources can be reassigned to other jobs, but the processing times are typically fixed regardless of the number of assigned resources.

Resource-constrained project scheduling problem (RCPSP) is an NP-hard optimization problem which aims at minimizing the total duration or makespan of a project with respect to precedence relations between the activities and the limited renewable resource availability (Blazewicz et al. 1983; Brucker et al. 1999; Demeulemeester and Herroelen 2002; Kolisch and Hartmann 2006). The methods to solve the RCPSP have ranged from exact methods including practical approaches (Zhu et al. 2006; Pantouvakis and Manoliadis 2006) to heuristic (Boctor 1996; Lova et al. 2006) or meta-heuristic [Hartmann 2001 (genetic algorithm); Nonobe and Ibaraki 2002 (tabu search algorithm); Bouleimen and Lecocq 2003 (simulated annealing algorithm); Jarboui et al. 2008 (combinatorial particle swarm optimization algorithm)] methods. Due to the complex nature of the problem, various heuristic methods have been proposed to solve the RCPSP in different context (Shue and Zamani 1999; Lee et al. 2003; Hasgül et al. 2009; Lova et al. 2009; Oddi et al. 2010). The trend of the research has shifted toward the meta-heuristic approaches because of the need for solving large realistic project instances (Thammano and Phu-ang 2012). Up to date, many meta-heuristic algorithms have been proposed in the literature (Debels et al. 2006; Tseng and Chen 2006; Zhang et al. 2006). Multi-mode resource-constrained project scheduling problem (MRCPSP) deals with the selection of a single activity mode from a set of available modes in order to construct a precedence and a (renewable and non-renewable) resource feasible project schedule with a minimal makespan (Coelho and Vanhoucke 2011). There is a clear distinction between algorithms that incorporate both renewable and non-renewable resource constraints and algorithms that limited to projects with only renewable resource constraints.

In this paper, we also assume that renewable resources are employed in the operations like RCPSP or MRCPSP. However, the main difference with RCPSP or MRCPSP is that processing times are influenced (or controllable) by the number of assigned resources in our model, whereas processing times are fixed in RCPSP or MRCPSP. Distinctive features of the problem we investigate in this paper are summarized in Table 1.

Evolutionary methods

We apply two evolutionary methods, the genetic algorithm (GA) and the tabu search (TS) methods, to improve a production schedule by focusing on certain measures of performance, such as the mean lateness, mean tardiness, and number of tardy jobs. The GA explains and simulates the mechanisms of natural systems. First introduced by Holland in 1975, this search algorithm mimics the principles of biological evolution; it essentially recombines existing solutions to obtain new ones and thus serves as a meta-heuristic for solving difficult optimization problems. The GA is an optimization technique for functions defined over finite domains, and it has been widely applied in a variety of scheduling problems over the past few decades. For an introduction to GA theory, we recommend works by Goldberg (1989), Melanie (2001), and Reeves and Rowe (2003). For an introduction to methods for implementing the GA in scheduling problems, the reader is referred to Chan et al. (2006); Damak et al. (2009); Fred and Nashat (1999), and Yang et al. (2007).

TS is a mathematical optimization method that belongs to the class of local search techniques; it is used for solving combinatorial optimization problems. TS was first proposed by Glover (1986), and it has been applied to several types of production scheduling problems (Barnes and Laguna 1993; Logendran et al. 1994; Persson et al. 2004), project scheduling problems (Calhoun et al. 2002; Garbowski and Pempera 2007), and optimization problems, including travelling salesman problems (TSPs) (Gendreau et al. 1998). It has also been applied to the global optimization of artificial neural networks (ANNs) (Sexton et al. 1998) and telecommunication networks (Costamagna et al. 1998). In this paper, we apply the GA and TS algorithms to the scheduling problem and compare the results of both approaches with an initial production schedule.

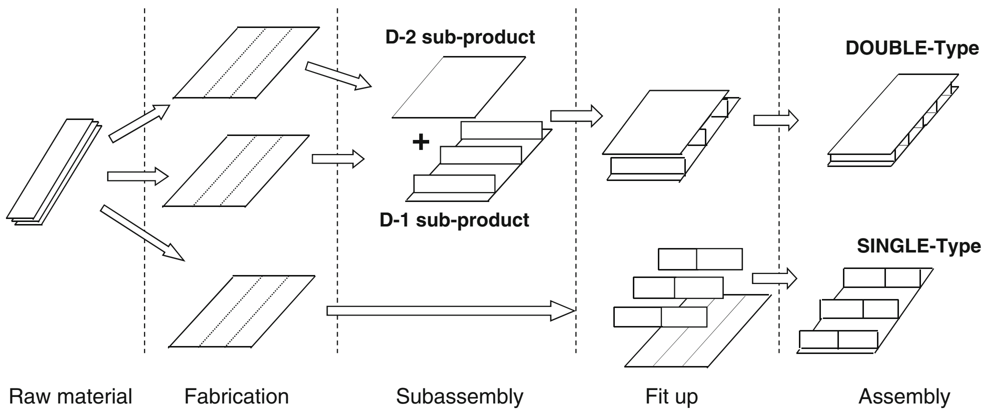

Fig. 1 Manufacturing processes for the products

Problem description

Shipbuilding environment

We consider the manufacturing jobs implemented at shipbuilding sites or factories. Diverse products that are completed via several operations, including fabrication, sub-assembly, and assembly, are undertaken at these sites. Fabrication is mainly done by machines, and the main resources for the sub-assembly and assembly operations are humans. Each operation starts after its antecedent operations are completed (satisfaction of the precedence constraints). Workloads are assigned to all of the operations that contribute to the finished product.

Two types of products are manufactured in the factory (see Fig. 1). The DOUBLE type undergoes a sub-assembly operation before the final assembly operation, and the SINGLE type undergoes the final assembly operation without a sub-assembly operation. That is, a DOUBLE-type product consists of several sub-products that are also composed of various parts. The sub-product with the longest production lead time is classified as a D-1 sub-product, and the other sub-products are classified as D-2 sub-products, as shown in Fig. 1. For each operation, resources are limited by the daily resource condition. The maximum number of resources assigned to an operation at a given time is predefined by the operation policies of the factory.

Production schedule

In this paper, production scheduling is used to determine the production sequence along with start dates of individual jobs, i.e. products; it is subject to the measures of total lateness, total tardiness, and the number of tardy jobs. Objective functions of each measure can be defined as follows.

- Total lateness:
  $Min \sum_j |F_j^{last} - D_j|$, for all job $j \in$ {D-1 sub-product, SINGLE}
- Total tardiness:
  $Min \sum_j \max(F_j^{last} - D_j, 0)$, for all job $j \in$ {D-1 sub-product, SINGLE}
- The number of tardy jobs:
  $Min \sum_j y_j$, if $F_j^{last} \le D_j$ then $y_j = 0$; otherwise $y_j = 1$, for all job $j \in$ {D-1 sub-product, SINGLE}

where, F_j^{last} is the finish time of the last operation of job j and D_j is the due date of job j. For the assessment of each schedule, one of aforementioned objective function is chosen and used according to the purpose of evaluation, and also utilized as a fitness function.

In the initial production schedule, the start date of each product is determined by offsetting due date of the product by the cumulative processing time, i.e. lead time (see Fig. 2). Here, historically estimated values are employed to initial processing times of individual operations. While the start date calculations for a SINGLE-type product are relatively straightforward, the determination is more complicated in the

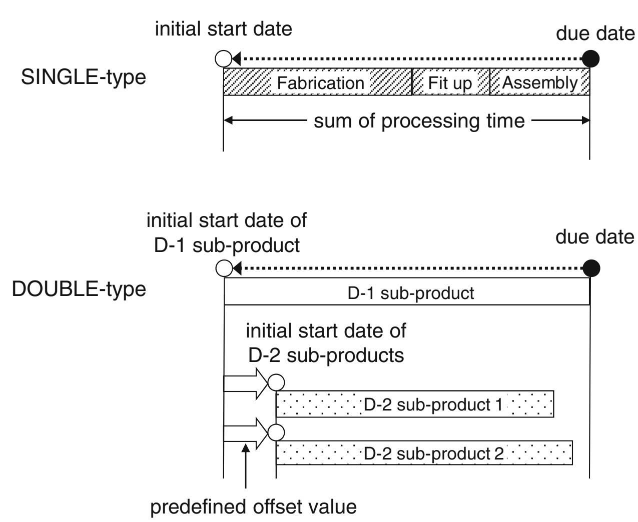

Fig. 2 Setting the operation start dates for the products

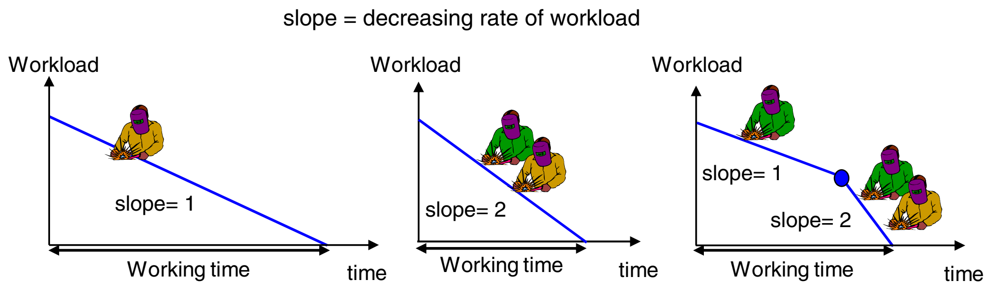

Fig. 3 Working times depending on the quantities of resources

case of a DOUBLE-type product, which consists of several sub-products. Start dates of each sub-product should be individually decided for production scheduling. Specifically, the start time of a D-2 sub-product is determined after deciding that of its paired D-1 sub-product, which shows longer production lead time than the other sub-products. In this paper, the start time of a D-2 sub-product is depicted as a predefined offset value that reflects delays based on its paired D-1 sub-product. In addition to the initial start dates, the offset values are improved by the evolutionary methods presented in "Evolutionary methods" section.

Actual processing time of each operation depends on the number of assigned resources as shown in Fig. 3, which shows that the processing time is inversely proportional to the number of resources. However, because the number of resources for a job may change during the processing of an operation, it is infeasible to apply a mathematical model or an analytical method to production scheduling. Instead, a method to determine the processing time of each operation, according to the dynamic condition of the resources, is needed. In this paper, a simulation method, further described in "Simulation model" section, is employed to estimate effectively the processing times of individual operations based on the emulation of resource assignments. Such a method may be appropriate in the context of variable processing times.

Simulation model

Assumptions

For simulation modeling, information on operations, operating resources, and processing times of jobs are required. In our problem, the operations of a job are specified in terms of workload rather than predefined processing time. Thus, the simulation method should include a calculation of the processing time of an operation according to the number of assigned resources and the manufacturing operation for each job. The proposed simulation model for the scheduling problem assumes the following conditions:

- The minimum number of resources is assigned.
- The daily business hours of the factory and the total available number of resources for each operation are given beforehand for each day.
- The maximum number of resources that can be assigned to an operation at a given time is limited and predefined.
- All the resources are identical.
- The guiding principle of resource assignment is "first available operation, first assigned".
- The preemption of assigned resources is not allowed.
- Idle resources can be assigned to an operation if the number of resources assigned is less than the maximum number of resources allowed at any time.

Simulation procedure

An overview of the simulation model is shown in Fig. 4. A simulation comprises of continually repeated resource assignments. At the beginning of each step, jobs are divided into two groups: those having feasible operations that meet the working conditions, when the job is either already started or ready to start, and those having no feasible operation. These two groups of jobs follow the procedures of static resource assignment and dynamic resource assignment, respectively. The static procedure is triggered at the beginning of each step, and the dynamic procedure is dynamically executed while each step is in progress.

In the static procedure, each feasible operation receives assigned resources at the current simulation time according to a predefined rule for resource assignment, where the current simulation time refers to the start time of a step. Then, the estimated finish time of each operation is calculated by dividing the workload by the number of assigned resources. The finish time of a step is set to the minimum estimated finish time of the operations, as well as start time of the next step. At the end of each step, jobs are classified into one of two groups: finishing an operation and part of an ongoing operation. The first group of jobs proceeds to its next operation, and the other group of jobs goes through the static resource assignment procedure once again in the next step.

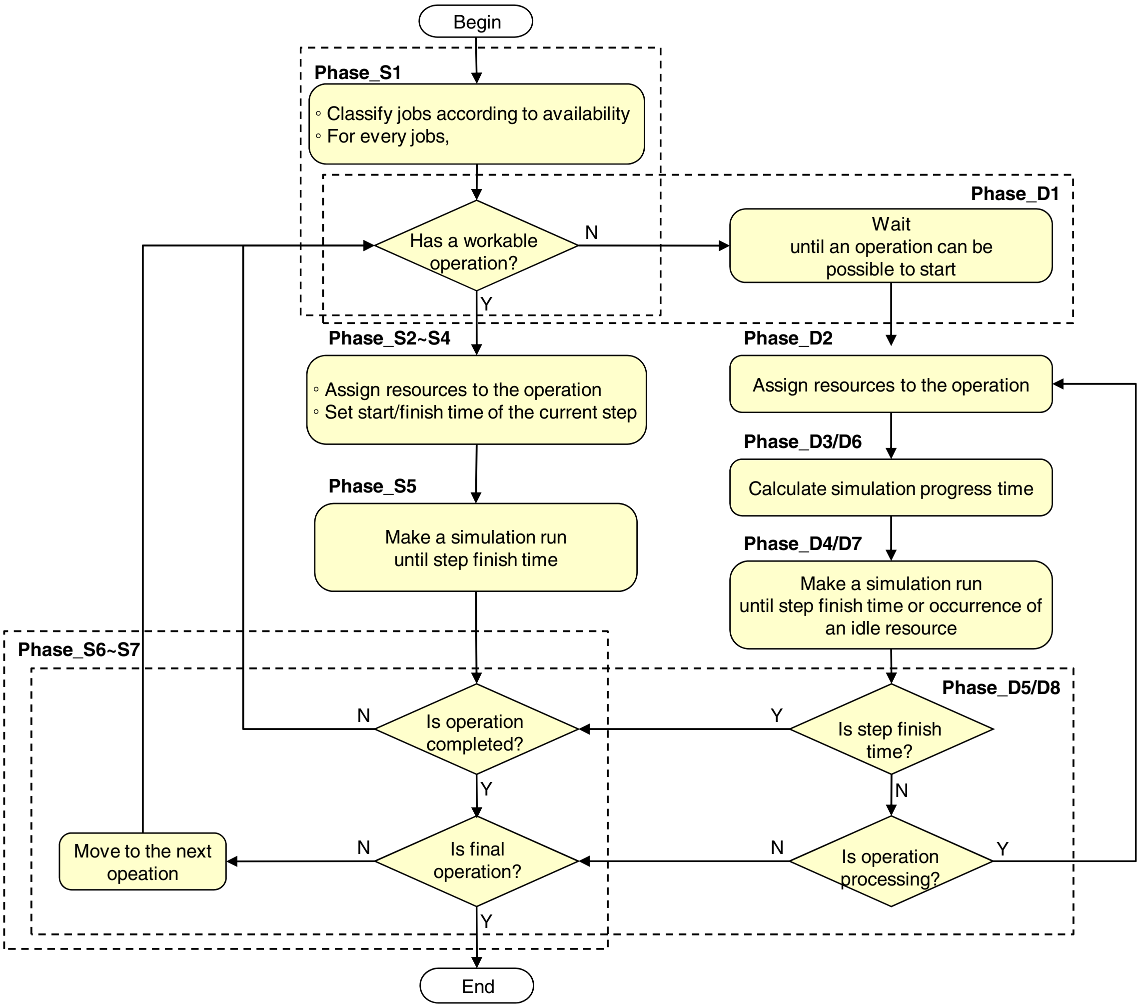

Fig. 4 Overview of the simulation model

In the dynamic procedure, jobs that have no feasible operation must be delayed until a satisfactory working condition emerges, which means the simulation time meets the start date of the jobs and some eligible resources become available. In this case, when the workable operation meets the working condition, the job is allocated resources. Then, the operation will proceed until the finish time of the step or until new idle resources become available for use. When the simulation time has proceeded to the time at which idle resources arrive, the dynamic resource assignment procedure is applied again to the job with an ongoing operation.

Resource assignment

The proposed simulation model defines the static resource assignment procedure and the dynamic resource assignment procedure, respectively, based on the following notation:

[Notation]

J_{ij} : Job j being operated in operation i, $i = 1 \sim M$, $j = 1 \sim M_i$

S_t : Start time of the tth step

C_t : Finish time of the tth step

P_i : Closing time of operation i for the day

N_i : Total quantity of resources available at operation i for the day

R_{ij} : Number of resources assigned to J_{ij}

F_{ij} : Estimated finish time of J_{ij}

$F_{\min}$: Minimum estimated finish time from all values of F_{ij}, i.e., $\min\{F_{ij}\}, \forall_i, \forall_j$

R_i^{Max} : Maximum quantity of resources simultaneously assigned to a job in operation i, i.e., $\max\{R_{ij}\}, \forall_j$

W_{ij} : Workload of J_{ij}

IR_i : Number of idle resources in operation i

IT_i : Time of occurrence of idle resources in operation i

T_{ij}^s : Time at which idle resources are added to J_{ij} between S_t and C_t

ST_{ij} : Simulation progress time of J_{ij}

Op_i^{Max} : Maximum number of assigned resources in operation i

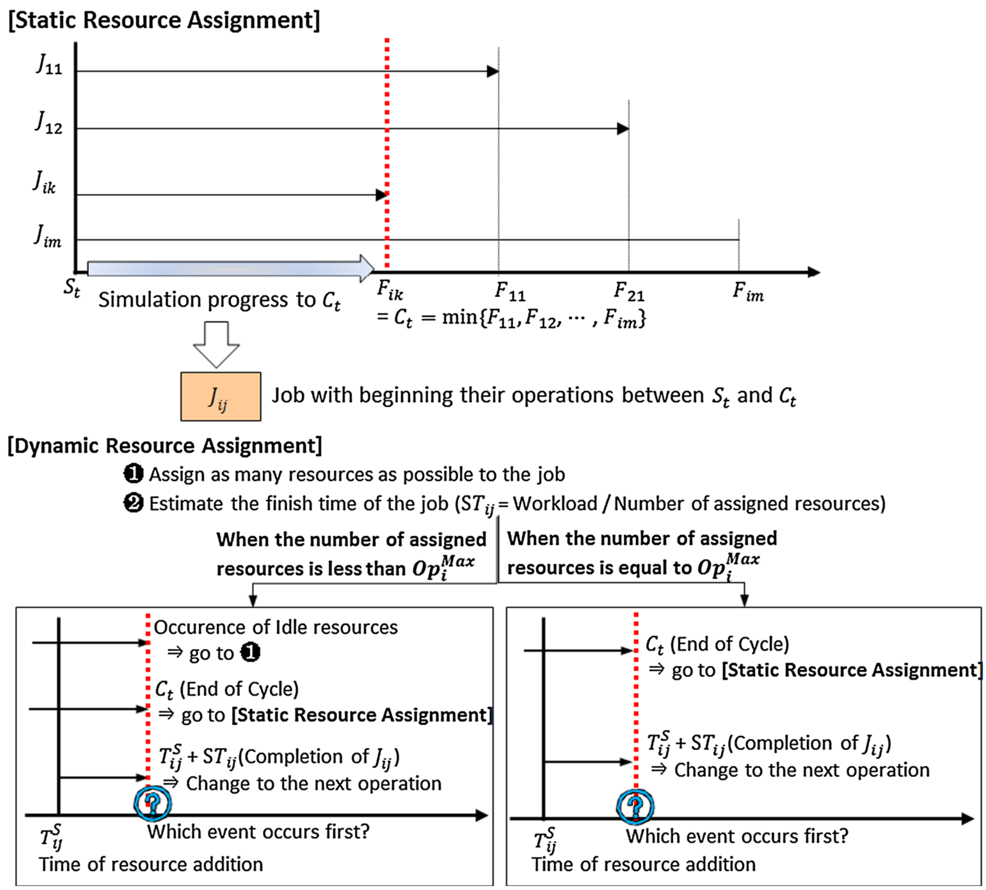

Fig. 5 Resource assignment in simulation

$A(t)$: Set of jobs for which processing operations are completed after the tth step

$B(t)$: Set of jobs for which operations are being processed during the tth step

The determinations of the start and finish times of each step during the simulation and the handling of operations begun during a step are shown in Fig. 5.

The resource assignment procedures for operations in the simulation model can be represented as follows:

Procedure of static resource assignment

Phase_S1. If the conditions to start the operation are met, then assign resources to the operation according to the predefined rule. Go to Phase_S2. If the conditions are not met, wait until the conditions to start the operation are met. If the conditions are met so that the worker can start the operation after waiting some time, go to the procedure of *dynamic resource assignment* (i.e., Phase_D1).

Phase_S2. Set the start and finish times of the current step and calculate the number of idle resources.

- $S_t = C_{t-1}, S_1 = 0$
- $IR_i = N_i - \sum_{j=1}^{M_i} R_{ij}$, for $i = 1 \sim M$

Phase_S3. Estimate the completion time of each operation.

- $F_{ij} = W_{ij}/R_{ij} + S_t$
- If $F_{ij} > P_i$, then $F_{ij} = P_i$, for $i = 1 \sim M$, $j = 1 \sim M_i$

Phase_S4. Set the finish time of the step.

- $F_{\min} = \min\{F_{ij}\}$, for $i = 1 \sim M$, $j = 1 \sim M_i$
- $C_t = F_{\min}$

Phase_S5. Run the simulation until C_t.

Phase_S6. Calculate the remaining workload of each operation at the end of the step.

- $W_{ij} = W_{ij} - (C_t - S_t) \times R_{ij}$, for $i = 1 \sim M$, $j = 1 \sim M_i$
- If $W_{ij} = 0$, then set $J_{ij} \in A(t)$
- If $J_{ij} \in A(t)$, then change the operation of the job to the next operation, and then wait until the conditions to start a new operation are met. If the conditions are met, go to the procedure of *dynamic resource assignment* (i.e., Phase_D1). If they are unmet, go to Phase_S7.

Phase_S7. For each operation, check the operations newly started during the cycle. That is to say, check whether there are operations included in $B(t)$.

- If $A(t)^c = \phi$ and $B(t) = \phi$, then stop.

  Else if $F_{\min} = P_i$, then stop.
  Else, go to Phase_S1.

Procedure of dynamic resource assignment

Phase_D1. If the last operation of a job is completed, dispose of the job. If the job is incomplete, go to Phase_D2.

Phase_D2. Assign the job as many resources as possible. Estimate the completion times of the operations under the quantities of resources assigned.

- $R_{ij} = \min\{IR_i, R_i^{\max}\}$
- $F_{ij} = W_{ij}/R_{ij} + T_{ij}^s$ (=the estimated processing time of J_{ij})
- If $R_{ij} = R_i^{\max}$, then go to Phase_D6.

  Else ($R_{ij} < R_i^{\max}$), go to Phase_D3.

Phase_D3. Set the simulation progress time (ST_{ij}) of the operation as follows:

- $ST_{ij} = \min\{IT_i, F_{ij}, F_{\min}\}$
- If $ST_{ij} = F_{ij}$, then $IT_i = F_{ij}$

Phase_D4. Run the simulation until ST_{ij}.

Phase_D5. Calculate the remaining workload of each operation after ST_{ij} as follows:

- $W_{ij} = W_{ij} - (ST_{ij} - T_{ij}^s) \times R_{ij}$
- If $ST_{ij} = F_{ij}$, then change the operation of the job to its next operation. Wait until the conditions to start a new operation are met. If the conditions are met, go to Phase_D1. If they are unmet and $ST_{ij} = IT_i$, then go to Phase_D2. If they are unmet and $ST_{ij} = F_{\min}$, then set $J_{ij} \in B(t)$ and go to the procedure of *static resource assignment* (i.e., Phase_S1).

Phase_D6. Set the progress time of the job processing operation (ST_{ij}).

- $ST_{ij} = \min\{F_{ij}, F_{\min}\}$

Phase_D7. Run the simulation until ST_{ij}.

Phase_D8. Calculate the remaining workload of the operation after ST_{ij}.

- $W_{ij} = W_{ij} - (ST_{ij} - T_{ij}^s) \times R_{ij}$
- If $W_{ij} = 0$, then change the operation of the job to its next operation. Wait until the conditions to start a new operation are met. If the conditions are met, go to the procedure of *dynamic resource assignment*. If they are unmet, then set $J_{ij} \in B(t)$ and go to the procedure of *static resource assignment* (i.e., Phase_S1).

Evolutionary methods

Representation

In this paper, the GA and TS are applied to improve a given production schedule. First, an individual representation, which is commonly used for both evolutionary methods, is introduced. A production schedule represents start dates of individual jobs, i.e. products, and each chromosome encodes deviations from an initial operation start dates of individual products as follows:

$$A_j = \begin{cases} SD_j - SD_j^{init}, & \text{if } J_j \in X \\ SD_j - SD_j^{paired}, & \text{otherwise} \end{cases}, \quad j = 1 \sim L$$

where A_j is an allele value denoting job j and L is the total number of jobs. SD_j is the start date of job j in the current solution. SD_j^{init} and SD_j^{paired} denote the initial start date of job j and the start date of its paired job, respectively. X denotes a set of jobs for SINGLE-type products and D-1 sub-products. The modified production schedule is developed by adding the value of each allele of the chromosome to the associated value of the initial production schedule. The structure of a chromosome is similar to that of the production schedule. A chromosome (that is, A_{L+1}), however, additionally includes an allele representing a resource assignment rule as shown in Fig. 6.

For example, product P8 is a SINGLE-type product and its initial operation start date is 2. Supposing that the value of associated allele of a chromosome is −1, the operation start date for product P8 is modified to 1 by increasing the initial start date (i.e. 2) by the value of its allele (i.e. −1). D-1 sub-products employ the same modification mechanism. However, D-2 sub-products should adopt a distinct deviation strategy from SINGLE-type products and D-1 sub-products. The start date of a D-2 sub-product is determined by increasing the start date of its paired D-1 sub-product by the delayed time (i.e. offset value) represented in the chromosome. For example, product P6 is a D-2 sub-product and product P1 is its paired D-1 sub-product. Supposing that the operation start time of product P1 is modified to 3 and the value of allele associated with product P6 is 1, the new operation start date of product P6, is modified to 4 due to the addition of the allele value (i.e. 1) to operation start date for product P1 (i.e. 3).

The alleles of the SINGLE-type products and the D-1 sub-products are integers. In the case of the D-2 sub-product, the alleles should be non-negative integers, because it is

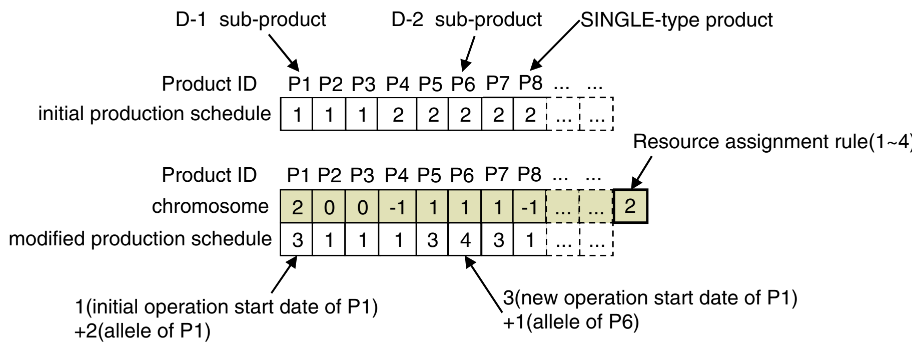

Fig. 6 Relationship between a production schedule and a chromosome

inefficient for a D-2 sub-product to start before its paired D-1 sub-product. The last allele of a chromosome represents one of the predefined resource assignment rules. The resource assignment rule is commonly applied during a simulation run for production scheduling.

A production schedule represented by an individual chromosome is evaluated by the simulation method discussed in “Simulation model” section. The result of the simulation includes the completion date of each product. The fitness of each individual is evaluated by comparing the due date for each job with the simulation results. The measures of tardiness and lateness of each job, as obtained from the simulation results, are used to determine the fitness of the schedule.

Genetic algorithm

The flowchart of the GA associated with the proposed simulation method is shown in Fig. 7. The overall structure of the proposed GA can be described as follows:

1. Representation. An individual is expressed as the difference between the operation start date and that of the initial schedule for each job.
2. Encoding and decoding. An individual is coded by an integer value within the predefined ranges given by the user. In the case of SINGLE products and D-1 sub-products, the decoding of the alleles is implemented by adding each allele to the initial production schedule. In the case of the D-2 sub-products, the modified operation start dates are determined by adding each allele to the operation start date for each pair of sub-products.
3. Crossover. The crossover operates on two chromosomes at a time and generates offspring by combining the features of both chromosomes. In this study, one-point, two-point, and uniform crossover operators presented by Gwiazda (2006) were employed to make a new production schedule.
4. Mutation. The value of an allele of the randomly selected gene is increased or decreased in the predefined range.
5. Fitness evaluation. The fitness is evaluated by the objective functions, which are the mean tardiness, number of tardy jobs, and mean lateness.
6. Stopping condition. The number of generations defined by the user is reached.
7. Selection. Both fitness proportional and tournament selections are employed to create a new population.

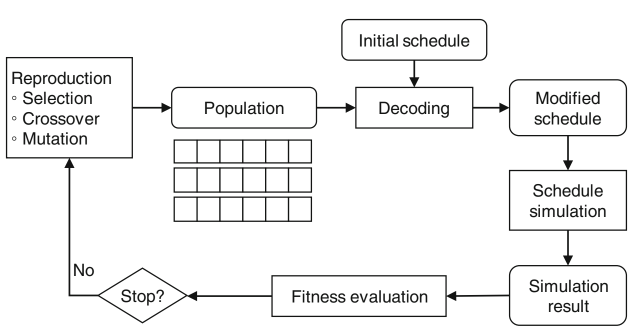

Fig. 7 Flowchart of genetic algorithms involving simulation

Tabu search approach

The TS has flexible memory structures in conjunction with restrictions on strategies and aspiration levels as means to exploit search spaces. The TS is a meta-heuristic approach that is mainly used to find a near-optimal solution for combinatorial optimization problems. It starts from the current solution and creates neighborhoods. At each step, the neighbors are searched to find the best neighbor, and a new solution is selected for the next step. To prevent cycling and to lead to good regions of the search space, a history of searches (tabu list) is recorded and employed in the search. The tabu list contains the attributes of forbidden solutions that have already acted as the solution in recent steps. The overall structure of the proposed TS algorithm can be described as follows:

1. Representation. The representation is the same as that of the GA.

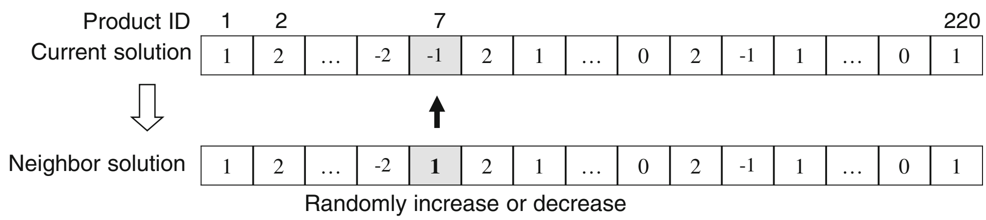

Fig. 8 Neighborhood according to an increase or decrease in the value of the allele

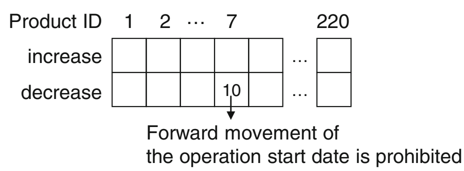

Fig. 9 Structure of the tabu list

2. Initial current solution. All the alleles are zero at the initial current solution. This means that the initial production schedule is the initial solution of tabu search.

3. Neighborhood. Neighbors are created by increasing or decreasing the value of the allele of the selected gene within the predefined range as shown in Fig. 8. If the jth gene is selected, the associated value is changed as follows:

$$A_j^{new} = \max\left(\min\left(A_j + I, A_{max}\right), A_{min}\right),$$
$$I = [0, A_{max} - A_{min}]$$

where A_j^{new} denotes the changed value and I is a randomly generated integer value. A_{min} and A_{max} are predefined minimum and maximum values of the allele, respectively.

4. Tabu attributes and tabu list. The tabu list is used to prevent the search from cycling between solutions. The tabu attributes refer to the features of solutions, the change history of which are documented on the tabu list; that is, when a neighborhood is obtained by increasing or decreasing the value of the allele representing the selected operation in the predefined range, the tabu list records the change (see Fig. 9). For example, if the tabu list has the value of 10 as the decrease record of product 7 as shown in Fig. 9, this means that the operation start date of product 7 is prohibited from moving forward.

5. Aspiration condition. If the fitness of a neighborhood solution is better than any prior solutions, then that neighborhood solution is chosen as the subsequent solution whether or not it is on the tabu list.

The TS employs the proposed simulation method to evaluate the fitness of a neighboring solution, as in the case of the GA. The flowchart of the TS associated with the simulation is shown in Fig. 10.

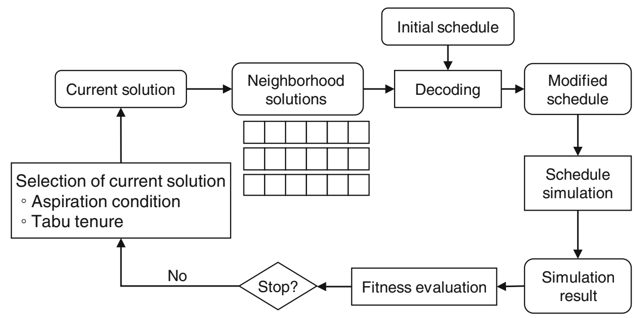

Fig. 10 Flowchart of tabu search involving simulation

As shown in Figs. 7 and 10, the proposed evolutionary methods use the simulation results to evaluate the fitness of each chromosome. The simulation results for the schedule are more realistic and accurate than any mathematical calculation. The various manufacturing constraints involved in the operations and the factory are included in the simulation model.

Computational experiments

Experimental configuration

On the basis of thorough operation analysis for basic working groups of a real shipbuilding factory, resource configurations of the working groups are classified into eight types as shown in Table 2. Resource configurations describe the respective limits on the number of available resources, which ranged from 3 to 5 for each operation in the factory. In this paper, computational experiments were conducted under these 8 resource configurations, supposing that the number of daily available resources for each operation is evenly randomized under the range shown in Table 2.

In addition, it is assumed that a total of 40 SINGLE-type products and 90 DOUBLE-type products should be produced in the factory. Because a DOUBLE-type product is composed of the D-1 and D-2 sub-products, the total number of products to be scheduled becomes 220. Whole working days in the simulation are 35, and daily working hours are ranged from 9 to 10. The initial production schedule is developed by considering the due date and total lead time of each

Table 2 Resource configuration for simulation (number of workers for each operation)

Operation	Maximum quantity of assignable resources								Number of daily available resources
	C01	C02	C03	C04	C05	C06	C07	C08	
Subassembly	3	3	3	3	4	4	4	4	7–9
Fit up	4	4	5	5	4	4	5	5	10–12
Assembly	4	5	4	5	4	5	4	5	9–12

product as well as the limits imposed by daily work capacity. If a resource assignment rule is defined before the simulation, the rule is applied to all operations through the simulation.

Resource assignment rules

The resource assignment rules, as follows, were examined to find out a dominant resource assignment rule in terms of their suitability for the objective (performance measure) of a given schedule:

1. Minimum Workload First Assignment (MWFA)
2. Earliest Due Date First Assignment (EDDFA)
3. Maximum Critical Ratio First Assignment (MaxCRFA)
4. Minimum Critical Ratio First Assignment (MinCRFA)

$$\text{Critical Ratio} = \frac{\textit{Due Date–Now}}{\textit{Current Workload}}$$

The criterion measures used in these experiments were the mean tardiness, number of tardy jobs, and mean flow time. The average values of the criterion measures for the predefined resource assignment rules are shown in Table 3. Number of simulation runs for each resource configuration is 10. That is, each value for an assignment rule averages the result from 80 simulation runs. It is reasonable to average the values because these 8 resource configurations represent general environment without bias.

According to the experimental results, the MinCRFA rule proved inferior to the other rules for all of the criterion measures, and three other rules yielded similar performances for the given criterion measures on the schedule. That is, we could not find a dominant rule for resource assignment, and it is worth finding a suitable resource assignment rule by means of an evolutionary method.

Table 3 Result of simulation experiments for resource assignment rules

	MWFA	EDDFA	MaxCRFA	MinCRFA
Number of tardy jobs	25.3	24.0	23.9	49.1
Mean tardiness	36.9	31.5	54.4	91.9
Mean flow time	123.2	130.1	125.4	154.0

Improvement

Evolutionary methods (that are GA and TS) are applied to improve on an initial production schedule, which is generated according to a randomly selected resource assignment rule. Three measures are used to determine the fitness of a schedule: the mean lateness, mean tardiness, and number of tardy jobs. Simulation modeling is performed by using ProModel 6.0. The system for the GA and TS is implemented using Microsoft Visual Basic 6.0, which can be interfaced with the simulation model. The following are the parameters of the GA for the experiment:

- Number of iterations: 100
- Population size: 20
- Crossover rate: 80 %
- Crossover method: One-point crossover
- Mutation rate: 3 %

The following are the parameters of the TS for the experiment.

- Number of iterations: 100
- Neighborhood size: 20
- Aspiration criterion: Improving the best-so-far solution
- Tabu tenure: 10
- Tabu attributes: An increase or decrease in the initial operation start date for each product

Figure 11 shows an instance of the total lateness change caused by increasing simulation iterations based on the GA and TS, respectively. The computational results are derived from eight simulation experiments each of which adopts an individual resource configuration and two simulation runs, as summarized in Table 4. As a consequence, the experimental results of both the GA and the TS are better than those of an initial schedule for all the measures. The initial schedule has been generated by the proposed simulation model based on a randomly selected rule. Even though the initial schedule is

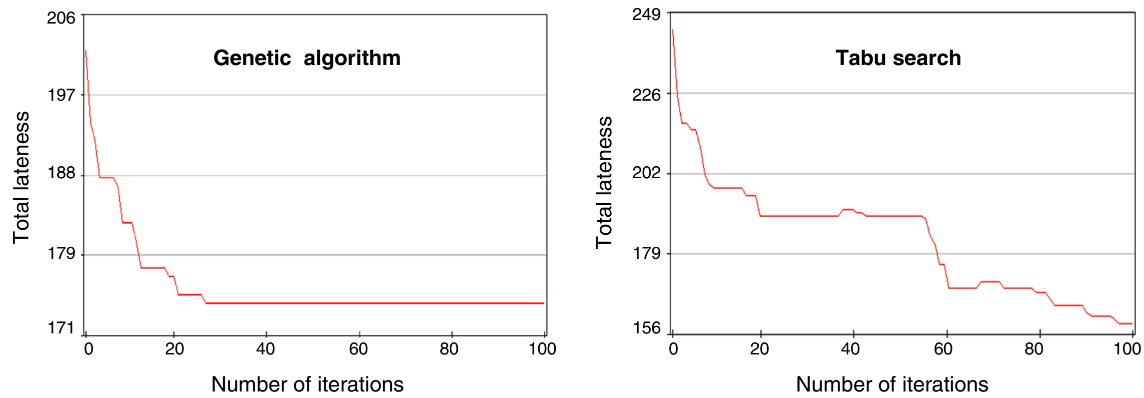

Fig. 11 Improvement in total lateness achieved by using the genetic algorithm and tabu search

Table 4 Summary of experiments

Fitness	Initial schedule	Genetic algorithm	Tabu search
Total lateness	248	174 (30 %)	159 (36 %)
Total tardiness	68	56 (18 %)	34 (50 %)
Number of tardy jobs	38	32 (16 %)	24 (37 %)

Improvement rates, compared to the fitness value of the initial schedule, for the genetic algorithm and tabu search methods are indicated in parentheses

a reasonable solution, the GA and the TS improves on the solution. In particular, the TS results in better performance than GA. This seems to be due to the fact that the initial schedule itself is reasonable in its degree and contribution of a current solution to its neighborhood in TS is relatively larger than that of a parent chromosome to its offspring in GA. However, it is hard to generalize from the comparison results. It is required to conduct extensive research on more various cases in order to draw a generic and definite conclusion.

Conclusions

In this paper, an evolutionary method with a simulation model was proposed for scheduling operations with specified workload and with processing times dependent on resource assignment. The workload-based scheduling problem is an unexplored topic; thus, we proposed a foundational solution approach. The simulation model results in a production schedule along with a scenario of resource allocation based on a predefined resource assignment rule. In addition, an evolutionary method improves the production schedule by selecting alternate resource assignment rules and changing timing constraints (e.g., start dates). The fitness of an individual production schedule is evaluated from the simulation result. In this paper, four resource assignment rules are employed, and two evolutionary methods (GA and TS) involving simulation are developed. The performance of both GA and TS is better than that of an initial schedule in terms of meeting the due dates of ordered products. The objective function of the final schedule improves by as much as 30 % of the initial production schedule.

Several interesting aspects inform future research. In addition to the GA and TS, other meta-heuristics can be developed. In this paper, the relationship between workload and processing time for an operation is assumed to be linear. However, there may be situations in which the relationship is non-linear. Developing algorithms for these cases may be worthwhile despite the complexity of the solution methods. We estimated the number of tardy jobs for a given number of workers. It would be interesting to estimate the number of workers required to avoid tardy jobs.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education(NRF-2011-0022767).

References

Alfieri, A. (2009). Workload simulation and optimization in multi-criteria hybrid flowshop scheduling: A case study. *International Journal of Production Research*, *47*(18), 5129–5145.

Bailey, R. (1996). *Human performance engineering: Designing high quality professional user interfaces for computer products, applications and systems*. Englewood Cliffs, NJ: Prentice-Hall.

Barnes, J., & Laguna, M. (1993). A tabu search experience in production scheduling. *Annals of Operations Research*, *41*, 141–156.

Blazewicz, J., & Lenstra, J. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, *5*, 11–24.

Boctor, F. (1996). A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. *European Journal of Operational Research*, *90*, 349–361.

Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, *149*, 268–281.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, *112*, 3–41.

Calhoun, K., Deckro, R., Moore, J., Chrissis, J., & Van Hove, J. (2002). Planning and re-planning in project and production scheduling. *Omega*, *30*(3), 155–170.

Chan, F. T. S., Wong, T. C., & Chan, L. Y. (2006). Flexible job-shop scheduling problem under resource constraints. *International Journal of Production Research*, *44*(11), 2071–2089.

Chaudhry, I. A., & Drake, P. R. (2009). Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. *International Journal of Advanced Manufacturing Technologies*, *42*, 581–594.

Cho, K., Chung, K., Park, C., Park, J., & Kim, H. (2001). A spatial scheduling system for block painting process in shipbuilding. *CIRP Annals-Manufacturing Technology*, *50*(1), 339–342.

Cho, K., Oh, J., Ryu, K., & Choi, H. (1998). An integrated process planning and scheduling system for block assembly in shipbuilding. *CIRP Annals-Manufacturing Technology*, *47*(1), 419–422.

Coelho, J., & Vanhoucke, M. (2011). Multi-mode project scheduling using RCPSP and SAT solvers. *European Journal of Operational Research*, *213*, 73–82.

Costamagna, E., Fanni, A., & Giacinto, G. (1998). A tabu search algorithm for the optimization of telecommunication networks. *European Journal of Operational Research*, *106*, 357–372.

Damak, J., Jarboui, B., Siarry, P., & Loukil, T. (2009). Differential evolution for solving multi-mode resource-constrained project scheduling problems. *Computers & Operations Research*, *36*(9), 2653–2659.

Debels, D., Reyck, B. D., Leus, R., & Vanhoucke, M. (2006). A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*, *169*, 638–653.

Demeulemeester, E., & Herroelen, W. (2002). *Project scheduling: A research handbook*. Dordrecht: Kluwer Academic Publishers.

Dudek, R., Panwalkar, S., & Smith, M. (1992). The lessons of flowshop scheduling research. *Operations Research*, *40*(1), 7–13.

ElMaraghy, H., & Patel, V. (2000). Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *Journal of Manufacturing Systems*, *19*(3), 186–198.

Ford, F., Bradbard, D., Ledbetter, W., & Cox, J. (1987). Use of operations research in production management. *Production and Inventory Management*, *28*(3), 59–62.

Fred, F. E., & Nashat, M. (1999). A distributed genetic algorithm for deterministic and stochastic labor scheduling problems. *European Journal of Operational Research*, *118*, 505–523.

Garbowski, J., & Pempera, J. (2007). The permutation flow shop problem with blocking: A tabu search approach. *Omega*, *35*(3), 302–311.

Gendreau, M., Laporte, G., & Semet, F. (1998). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, *106*, 539–545.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, *13*(5), 533–549.

Graves, S. (1981). A review of production scheduling. *Operations Research*, *29*(4), 646–675.

Gwiazda, T. D. (2006). *Genetic Algorithms Reference*. Tomasz Gwiazda E-book.

Hartmann, S. (2001). Project scheduling with multiple modes: A genetic algorithm. *Annals of Operations Research*, *102*, 111–135.

Hasgül, S., Saricicek, I., Ozkan, M., & Parlaktuna, O. (2009). Project-oriented task scheduling for mobile robot team. *Journal of Intelligent Manufacturing*, *20*(2), 151–158.

Hoogeveen, H., & Woeginger, G. J. (2002). Some comments on sequencing with controllable processing times. *Computing*, *68*, 181–192.

Jarboui, B., Damak, N., Siarry, P., & Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, *195*, 299–308.

Kim, K., & Jung, S. (2006). Load scheduling for multiple quay cranes in port container terminals. *Journal of Intelligent Manufacturing*, *17*(5), 453–463.

Kim, K., & Lee, M. (2007). Scheduling trucks in local depots for door-to-door delivery services. *Journal of the Operational Research Society*, *58*(9), 1195–1202.

Kim, S., & Lee, K. (1997). An optimization-based decision support system for ship scheduling. *Computers & Industrial Engineering*, *33*(3–4), 689–692.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, *174*, 23–37.

Lee, J., Lee, K., Park, H., Hong, J., & Lee, J. (1997). Developing scheduling systems for Daewoo shipbuilding: DAS project. *European Journal of Operational Research*, *97*(2), 380–395.

Lee, K., Lee, J., & Choi, S. (1996). A spatial scheduling system and its application to shipbuilding: DAS-CURVE. *Expert Systems with Applications*, *10*(3–4), 311–324.

Lee, Y.-H., Kumara, S. R. T., & Chatterjee, K. (2003). Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism. *Journal of Intelligent Manufacturing*, *14*(5), 471–484.

Liang, C., Huang, Y., & Yang, Y. (2009). A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, *56*(3), 1021–1028.

Lodree, E. J., Geiger, C. D., & Jiang, X. (2009). Taxonomy for integrating scheduling theory and human factors: Review and research opportunities. *International Journal of Industrial Ergonomics*, *39*(1), 39–51.

Logendran, R., Ramakrjshna, P., & Sriskandarajah, C. (1994). Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research*, *32*(2), 273–297.

Lova, A., Tormos, P., & Barber, F. (2006). Multi-mode resource constrained project scheduling: Scheduling schemes, priority rules and mode selection rules. *Inteligencia Artificial*, *30*, 69–86.

Lova, A., Tormos, P., Cervantes, M., & Barber, F. (2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics*, *117*, 302–316.

MacCarthy, B., & Liu, J. (1993). Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, *31*(1), 59–79.

Matthews, G., Davies, D., Westerman, S., & Stammers, R. (2000). *Human performance: Cognition, stress and individual difference*. Philadelphia: Psychology Press.

McKay, K., Pinedo, M., & Webster, S. (2002). Practice-focused research issues for scheduling systems. *Production and Operations Management*, *11*(2), 249–258.

McKay, K., Safayeni, F., & Buzacott, J. (1988). Job-shop scheduling theory: What is relevant? *Interfaces*, *4*(18), 84–90.

Nonobe, K., & Ibaraki, T. (2002). Formulation and tabu search algorithm for the resource constrained project scheduling problem. In C. Ribeiro & P. Hansen (Eds.), *Essays and surveys in metaheuristics* (pp. 557–588). Dodrecht: Kluwer Academic Publishers.

Nowicki, E., & Zdrzalka, S. (1995). A bicriterion approach to preemptive scheduling of parallel machines with controllable job processing times. *Discrete Applied Mathematics*, *63*, 237–256.

Oddi, A., Cesta, A., Policella, N., & Smith, S. F. (2010). Iterative flattening search for resource constrained scheduling. *Journal of Intelligent Manufacturing*, *21*(1), 17–30.

Olhager, J., & Rapp, B. (1995). Operations research techniques in manufacturing planning and control systems. *International Transactions in Operational Research*, *2*(1), 29–43.

Pantouvakis, J., & Manoliadis, O. G. (2006). A practical approach to resource-constrained project scheduling. *Operational Research*, *6*(3), 299–309.

Persson, J. A., Göthe-Lundgren, M., Lundgren, J. T., & Gendron, B. (2004). A tabu search heuristic for scheduling the production processes at an oil refinery. *International Journal of Production Research*, *42*(3), 445–471.

Reisman, A., Kumar, A., & Motwani, J. (1997). Flowshop scheduling/sequencing research: A statistical review of the literature 1952–1994. *IEEE Transactions on Engineering Management*, *44*(3), 316–329.

Sexton, R., Alidaee, B., Dorsey, R., & Johnson, J. (1998). Global optimization for artificial neural networks: A tabu search application. *European Journal of Operational Research*, *106*, 570–584.

Shabtay, D., & Steiner, G. (2007). A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, *155*, 1643–1666.

Shue, L.-Y., & Zamani, R. (1999). Intelligent search method for project scheduling problems. *Journal of Intelligent Manufacturing*, *10*(3), 279–288.

Tavakkoli-Moghaddam, R., Markui, A., Salahi, S., Bazzazi, M., & Taheri, F. (2009). An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*, *56*(1), 241–248.

Thammano, A., & Phu-ang, A. (2012). A hybrid evolutionary algorithm for the resource-constrained project scheduling problem. *Artificial Life and Robotics*, *17*(2), 312–316.

Tseng, L. Y., & Chen, S. C. (2006). A hybrid metaheuristic for the resource constrained project scheduling problem. *European Journal of Operational Research*, *175*, 707–721.

Vickson, R. G. (1980). Two single machine sequencing problem involving controllable job processing times. *AIIE Transactions*, *12*(3), 258–262.

Xu, J., Xu, X., & Xie, S. Q. (2011). Recent developments in dual resource constrained (DRC) system research. *European Journal of Operation Research*, *215*, 309–318.

Yang, T., Kuo, Y., & Cho, C. (2007). A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem. *European Journal of Operational Research*, *176*, 1859–1873.

Zhang, H., Tam, C. M., & Li, H. (2006). Multi-mode project scheduling based on particle swarm optimization. *Computer-Aided Civil and Infrastructure Engineering*, *21*(2), 93–103.

Zhu, G., Bard, J., & Tu, G. (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *Journal on Computing*, *18*(3), 377–390.