# ORIGINAL ARTICLE

# Simulation optimization for a flexible jobshop scheduling problem using an estimation of distribution algorithm

Ricardo Pérez-Rodríguez • S. Jöns • Arturo Hernández-Aguirre • Carlos Alberto-Ochoa

Received: 20 November 2013 / Accepted: 6 March 2014 / Published online: 4 April 2014 © Springer-Verlag London 2014

Abstract The flexible jobshop scheduling problem permits the operation of each job to be processed by more than one machine. The idea is to assign the processing sequence of operations on the machines and the assignment of operations on machines such that the system objectives can be optimized. The assignment mentioned is a difficult task to implement on real manufacturing environments because there are many assumptions to satisfy, especially when the amount of work is not constant or sufficient to keep the manufacturing process busy for a long time, causing intermittent idle times. An estimation of distribution algorithm-based approach coupled with a simulation model is developed to solve the problem and implement the solution. Using the proposed approach, the shop performance can be noticeably improved when different machines are assigned to different schedules.

Keywords Scheduling · Event-discrete simulation · Estimation of distribution algorithm · Simulation optimization · Work in process · Flexible jobshop scheduling problem

R. Pérez-Rodríguez (⊠) CIATEC, A. C. Omega 201, Fracc. Delta, C.P. 37545 León, Guanajuato, México e-mail: rperez.picyt@ciatec.mx

### S. Jöns

UPGTO, CONACYT, Av. Universidad Norte s/n, Comunidad "Juan Alonso", C.P. 38483 Cortázar, Guanajuato, México

A. Hernández-Aguirre

CIMAT, A. C. Callejón de Jalisco s/n, Mineral de Valenciana, C.P. 36240 Guanajuato, Guanajuato, México

## C. Alberto-Ochoa

UACJ, Henry Dunant 4016, Zona Pronaf, C.P. 32310 Ciudad Juárez, Chihuahua, México

# **1** Introduction

Estimation of distribution algorithms (EDA), introduced by Mühlenbein and Paa $\beta$  [1], have been successfully used to solve complex combinatorial optimization problems such as scheduling. Chen et al. [2], Liu et al. [3], and Pan and Ruiz [4] can be consulted on this issue.

Disadvantages of EDAs such as loss of diversity and insufficient use of location information of solutions have been tackled successfully by incorporating other methods such as genetic algorithms (GAs) during the evolutionary process. Chen et al. [5] use this approach.

Several works have been done in order to capture the problem structure with more precision. Advanced probabilistic models, which solve scheduling problems through EDAs, have been proposed to attempt to integrate higher-order interactions to enhance the solution quality. Wang et al. [6] and Chen et al. [7] have contributed to research on it.

Flowshop scheduling problem and jobshop scheduling problem on flexible shops have been studied and solved through EDAs from an academic perspective. Jarboui et al. [8] and Zhang and Li [9] are in this category.

Working out real-world scheduling problems has been considered a priority in the last three decades because the conditions of any service or manufacturing process requires an appropriate modeling on the most important variables that affect the performance of the process. The process might have different scenarios to optimize or different characteristics to consider in order to solve the scheduling problem, not only the academic perspective on such things as market conditions, competitors, information systems, work conditions, setup time, maintenance, transfering of parts, labor requirements, storage, shifts, breaks, the process itself. Chan et al. [10], Wadhwa et al. [11], Chan and Chung [12], and Chan [13] have considered the real process environment and its flexibility in their approaches to work out real-world scheduling problems.

All current research work focuses on the fact that only one schedule is assigned to all the machines on the shop floor, but the performance of scheduling is different under different circumstances; the shop performance can be improved if different machines are assigned to different schedules. For example, the steel doors manufacturing process studied in this research does not require all machines to be ready at the beginning of the schedule, because it is extensive and requires time to produce a complete job. The work content varies greatly at each step. If all machines are ready at the beginning of the schedule, it should cause idle times. For this reason and contrary to current research, this paper presents an alternative solution to the scheduling problem from an industrial perspective where the start time for each work shift is an input variable to consider. We provide a simple example of the variable mentioned by considering a problem with four jobs: the four machines shown in Fig. 1. We try to achieve the minimum idle time possible for each machine in a work shift of 8 h. An alternative schedule, including off-duty hours for each machine, is proposed in Fig. 2. To the best of our knowledge, the parameter described has not been developed through these kinds of algorithms.

In addition, all EDAs that have been used for scheduling up to this point can be used for service and/or manufacturing applications. However, in these environments, some assumptions studied from an academic approach may not be met. For example, the assumption that operations cannot be interrupted is difficult to meet, because there are many reasons to interrupt operations at any workstation or machine due to failures, adjustments, incorrect set up, scrap, shutdown, or the sudden arrival of priority jobs.

In order to avoid the use of theoretical assumptions that can hardly be met or to try to capture the real operating conditions on service or manufacturing systems, simulation optimization is an alternative.

Although there are several commercial simulation languages that provide optimization tools, to the best of our knowledge, no current commercial simulation language uses EDAs as an optimization method.

The approach taken in this study combines the key advantages of both EDAs and simulation. The focus of this study is

### Fig. 1 Scheduling example

			Opera	tion	Fix	ed Process	sing Time	s (hrs)		
			job, prece	edence						
					M <sub>1</sub>	M <sub>2</sub>	$M_3$	Μ	4	
			O <sub>1,</sub>	1	2	-	-	-		
			O <sub>1,</sub>	2	-	4	-	-		
			O <sub>2</sub>	1	3	-	-			
			O <sub>2</sub>	2	-	-	3	-		
			O <sub>2</sub>	3	-	-	6	-		
			O <sub>3</sub> ,	1	2	-	-			
			O <sub>3</sub>	2	-	-	3	3	;	
			0 <sub>4</sub>	1	1	-	-			
			0 <sub>4</sub>	2	-	3	-	3	5	
			O <sub>4</sub> ,	3	-	-	-	5	;	
			Real	work shift						
M1	O <sub>11</sub>	O <sub>31</sub>	0 <sub>2</sub>	1	O <sub>41</sub>					
M2		C	<b>)</b> <sub>12</sub>							
М3					O <sub>22</sub>			O <sub>23</sub>		
M4			O <sub>3</sub>	2		O <sub>42</sub>		0	43	
	1 2	3 4	5	6 7	8	9 10	11 12	13	14 15	16
	Idle time		Work	in proces	s					

### Fig. 2 Scheduling alternative



5

to employ simulation with estimation of distribution algorithm (SEDA) where three probabilistic models are utilized. The first one generates the processing sequence of operations on the machines: the second produces the assignment of operations on machines, and the third obtains the start time for each work shift.

Although different performance measures have been considered for optimizing scheduling problems, the work in process (WIP), as an important output variable on scheduling problems, has been handled with little depth through EDAs. This research proposes to build an eventdiscrete simulation model and use SEDA as an optimization method in a steel doors manufacturing shop where operations are extensive and diverse and belong to flexible jobshop configuration. The objective is to generate schedules that can obtain a small amount of WIP as a performance measure.

### 2 Literature review

A discussion about the most current research on scheduling problems with EDAs is outlined below.

Chen et al. [2] propose guidelines for developing effective EDAs to solve single machine scheduling problems, particularly the minimization of the total weighted earliness and tardiness costs. In general, they used an EDA with an operator that they call "guided mutation" to generate effective offspring. The beginning of their algorithm produces new solutions mainly by genetic operators. After this, they use the probabilistic model to generate better individuals when the searching process reaches a more stable state. Therefore, sampling new individuals periodically differentiates EDAs

from each other because most EDAs generate entirely new solutions.

Recently, some attempts have been made to combine EDAs with the traditional crossover and mutation operators of GAs [14]. Chen et al. [5] use this approach. They employ an approximate probabilistic model to estimate the quality of candidate solutions to enable the crossover and mutation operators to generate more promising solutions. They work on the permutation flowshop scheduling problem (PFSP). It is one of the best-known NP-hard (non-deterministic polynomial-time) problems. The probabilistic model used is not a source for generating new solutions but acts as a fitness predictor for guiding the crossover and mutation operators to generate better solutions.

Chen et al. [7] also work on PFSPs. They employ two probabilistic models, while most EDAs do not apply more than one model. The first model represents the number of times that any job appears before or at a specific position in the sequences. This model shows the importance of the jobs in the sequence and was used in research conducted by Jarboui et al. [8]. The second model indicates whether any job immediately follows another in the sequences, i.e., this model indicates the number of times that any job immediately follows another. In addition, it is important to note that by combining a genetic-operator approach with a probabilistic model, the authors were able to avoid the loss of data diversity which EDAs often demonstrate.

Pan and Ruiz [4] offer an EDA for lot-streaming flowshop problems with setup times. According to the researchers, in a traditional flowshop, each job is assumed to be indivisible and cannot be transferred to a downstream machine until the whole operation on the preceding machine is finished. Nevertheless, this is not the case in many practical

environments where a job or lot consists of many identical items. The real contribution is how Pan and Ruiz [4] handle the setup time concept in their algorithm.

Wang et al. [6] work on the flexible jobshop scheduling problem (FJSP). The authors proposed a bi-population based on EDA, which they called BEDA to solve FJSP with the criterion of minimizing the maximum completion time. In the BEDA, the population may be divided into two subpopulations with a splitting criterion, and the two subpopulations may be recombined as the entire population with a combination criterion to achieve a satisfactory searching quality.

All of this current research work uses discrete EDAs. In these kinds of EDAs, each individual explicitly shows its information in the sequence of jobs to be processed. The hybridization between any discrete EDA and any heuristic method permits obtaining promising solutions. The probabilistic models used in all of this current research work are updated each time a job is assigned in the sequence. This updating eliminates the possibility of choosing a previous job, although the authors of this research almost never explicitly mention that a modification in the sampling process has to be carried out. For example, Shim et al. [15] use EDAs for solving the multi-objective traveling salesman problem. The authors opined that the sampling mechanism does not consider which city has or has not been included in the route. In order to get feasible solutions, a refinement operator is proposed for tackling the inconvenience of the permutationbased representation.

Although some promising results have been reported by using high-order interactions in probabilistic models, those results do not necessarily outperform simple models in dealing with some real-world hard problems because these complicated models can only consider a very tiny percentage of variable interactions in a hard problem [7]. As a result, simulation optimization is a good alternative to tackle this situation.

The main differences and similarities between current research and this paper are shown in Table 1. In addition, in Table 2, it details the most important commercial simulation languages that provide optimization tools.

As we can see from the previous review, to the best of our knowledge, the industrial perspective has not been thoroughly considered In addition, EDAs have not been developed for use as an optimization method for simulation optimization. Finally, some assumptions that are not necessarily met in service and/or manufacturing systems are used in all of the current research.

### **3 Problem statement**

The steel doors manufacturing shop, where operations are extensive and diverse, belongs to flexible jobshop

configuration. Wang et al. [6] and Yan and Wang [16] explain the problem formulation for this configuration.

The flexible jobshop scheduling problem FJSP is commonly defined as follows: there are *n* jobs  $J = \{J_1, J_2, ..., J_n\}$  to be processed on *m* machines  $M = \{M_1, M_2, ..., M_n\}$ . A job  $J_i$  is formed by a sequence of  $n_i$  operations  $\{O_{i,1}, O_{i,2}, ..., O_{i,n_i}\}$ to be performed one after another according to a given sequence. The execution of  $O_{i,j}$  requires one machine out of a set of  $m_{i,j}$  given machines  $M_{i,j} \subseteq M$ . Preemption is not allowed, i.e., each operation must be completed without interruption once it starts.

In order to simplify the notations for the flexible jobshop scheduling problem in this research, it is convenient to identify the  $O_{i,j}$  by numbers 1,..., N where  $N := \sum_{i=1}^{n} n_i$ . Consequently, the processing time of the operation *i* on machine  $k \in M_i$  is denoted by  $t_{i,k}$ . In addition, the operations set is established as O.

Let J(i) denote the job to which operation *i* belongs and let P(i) be the position of operation *i* in the sequence of operations belonging to job J(i) starting with one, i.e., P(i)=1 if the operation *i* is the first operation of a job. Furthermore, the index set  $I_k$  defined by  $I_k := \{i \in O | k \in M_i\}$  denotes the indices of operations  $i \in O$ that can be processed on machine *k*. Consequently, there are  $|I_k|$  positions on machine *k*.

In order to model the assignment of operations to machines, assignment binary variables  $x_{i,k,p}$  for all  $p_k = 1, ..., |I_k|$ , k=1, ..., m,  $i \in O$  are introduced if  $x_{i,k,p} = 1$  means that the operation *i* is scheduled for position *p* on machine *k*.

Furthermore,  $S_i$  is defined as the starting time for operation *i*.

For each job, the corresponding operations have to be processed in the given order, that is, the starting time for an operation must not be earlier than the point at which the preceding operation in the sequence of operations of the respective job is completed. This constraint is imposed simultaneously on all appropriate pairs of operations, aggregated in the set of conjunctions *C* given by  $C := \{(i,j) | i,j \in O: J(i) = J(j) \land P(j) = P(i) + 1\}$ . Consequently, the *precedence constraints* are given by

$$S_i + \sum_{k \in M_i} \sum_{p=1}^{|I_k|} x_{i,k,p} t_{i,k} \le S_j \text{ for all}(i,j) \in C.$$

$$\tag{1}$$

Moreover, each operation has to be assigned to exactly one position, which is ensured by

$$\sum_{k=1}^{m} \sum_{p=1}^{|l_k|} x_{i,k,p} = 1 \text{ for all } i \in O.$$
(2)

### Table 1 EDAs for scheduling

	Algorithms						
Elements	Chen et al. [2] EA/G	Chen et al. [5] Self-guided GA	Chen et al. [7] eACGA	Jarboui et al. [8] JEDA	Pan and Ruiz [4]	Wang et al. [6] BEDA	SEDA
Configuration style	Single-machine	PFSP	PFSP	PFSP	Lot-streaming PFSP	FJSP	FJSP
Processing times	Fixed	Fixed	Fixed	Fixed	Fixed	Fixed	Variable
Assumptions	Required	Required	Required	Required	Partially required	Required	Not required
Probabilistic model	Univariate	Univariate	Uni/bivariate	Univariate	Uni/bivariate	bivariate	Bivariate
Type of EDA	Discrete	Discrete	Discrete	Discrete	Discrete	Discrete	Discrete and continuous
Hybridization	EDA-GA	EDA-GA	EDA-GA	EDA-VNS	EDA-VNS	EDA-local search	EDA-simulation
Populations	One	One	One	One	One	Two	One
Objective	Weighted earliness, tardiness costs	Makespan	Makespan	Total flow time	Makespan	Makespan	Work in process
Simulation	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Included
Approach	Academic	Academic	Academic	Academic	Industrial, sequence- dependent setup time considered	Academic	Industrial, all the environment as really it is
PFSP permutation f problem	lowshop scheduling						
FJSP flexible jobsho problem	op scheduling						
VNS variable neigh	borhood search						

Additionally, only one operation can be assigned to each position, due to constraints

$$\sum_{i \in O} x_{i,k,p} \le 1 \text{ for all } p = 1, ..., |I_k|, k = 1, ..., m.$$
(3)

The positions on each machine have to be subsequently filled, that is, an operation is only allowed to be assigned to a

	Optimization	i tool	
Languages	Tabu search	Genetic algorithms	Evolutionary strategies
Simul8®	+		
SimFlex®	+		
Simprocess®	+		
Promodel®	+		
Arena®	+		
Quest®	+		
Extend®		+	
@Risk <sup>®</sup>		+	
AutoMod®			+
+ Included			

position on a machine if the preceding position is already filled. This condition is ensured by

$$\sum_{i\in O} x_{i,k,p} \le \sum_{i\in O} x_{i,k,p-1} \tag{4}$$

for all  $p = 2, ..., |I_k|, k = 1, ..., m$ .

In order to interconnect the machine position variables with the starting time variables and to enforce a feasible schedule, non-overlapping constraints are defined by

$$S_{i} + t_{i,k} - M \left( 2 - x_{i,k,p-1} - x_{j,k,p} \right) \le S_{j}$$
(5)

for all  $p=2,...,|I_k|$ ,  $i \neq j \in I_k$ , k=1,...,m. If the operations *i* and *j* are assigned to the same machine *k* for consecutive positions p-1 and *p*, then the starting time  $S_j$  of operation *j* must not be earlier than the completion time  $S_i+t_{i,k}$  of operation *i*. *M* is a big constant taken sufficiently large in order to guarantee constraints (Eq. 5) to be valid if at least one of the machine position variables  $x_{i,k,p}$  and  $x_{i,k,p-1}$  is zero; in other words, operations *i* and *j* are not assigned to consecutive positions on the same machine, and consequently,

a non-overlapping constraint does not have to be taken into account.

Different starting times for each machine were considered in this research. Let  $T_k$  be the off-duty time for each machine k. The starting time for the first operation on each machine must not be earlier than  $T_k$ , which is ensured by

$$T_k + S_i + \sum_{p=1} x_{i,k,p} t_{i,k} \le S_j \text{ for all } i \in I_k, k = 1, ..., m.$$
 (6)

Let H be the total time in the work shift for the entire manufacturing process. Consequently, the real work shift R for each machine k is defined by

$$T_k + H = R_k \text{ for all } k = 1, \dots, m.$$
(7)

The completion time  $C_k$  on each machine k is defined as the total time required to conclude all the operations scheduled, which is ensured by

$$\max_{i \in I_k} \{S_i + t_{i,k}\} \le C_k \quad \text{for all } k = 1, ..., m.$$
(8)

We consider to minimizing the work in the process at the end of each real work shift on each machine, that is, the difference between the completion time and the real work shift for each machine, given by

$$Min WIP = \sum_{k=1}^{m} \max\{C_k - R_k, 0\}$$
(9)

Thus, the problem formulation for the steel doors manufacturing process is given by

$$Min WIP = \sum_{k=1}^{m} \max\{C_k - R_k, 0\}$$

, m.

s.t  

$$\max_{i \in I_k} \{S_i + t_{i,k}\} \le C_k \quad \text{for all } k = 1, ..., m. \\ S_i + \sum_{k \in M_i} \sum_{p=1}^{|I_k|} x_{i,k,p} t_{i,k} \le S_j \quad \text{for all } (i,j) \in C . \\ \sum_{k=1}^m \sum_{p=1}^{|I_k|} x_{i,k,p} = 1 \quad \text{for all } i \in O. \\ \sum_{i \in O} x_{i,k,p} \le 1 \quad \text{for all } p = 1, ..., |I_k|, k = 1, ..., m. \\ \sum_{i \in O} x_{i,k,p} \le \sum_{i \in O} x_{i,k,p-1} \quad \text{for all } p = 2, ..., |I_k|, k = 1, ..., m. \\ S_i + t_{i,k} - M (2 - x_{i,k,p-1} - x_{j,k,p}) \le S_j \quad \text{for all } p = 2, ..., |I_k|, i \neq j \in I_k, k = 1, ..., m. \\ T_k + S_i + \sum_{p=1} x_{i,k,p} t_{i,k} \le S_j \quad \text{for all } i \in O, k = 1, ..., m. \\ T_k + H = R_k \quad \text{for all } k = 1, ..., m. \\ S_i \ge 0 \quad \text{for all } k = 1, ..., m. \\ x_{i,k,p} \in \{0, 1\} \quad \text{for all } p = 1, ..., |I_k|, k \in M_i, i \in O. \\ H \text{ constant} \end{cases}$$

Although the problem formulation for the steel doors manufacturing shop has been explained above, it includes assumptions that are not relevant or consistent with the manufacturing process mentioned. Unfortunately, some of these assumptions simply cannot be applied in the steel doors manufacturing process. This situation impedes using EDAs directly. Some assumptions that are not met are:

1. The machines are assumed to be set up in series. In the steel door manufacturing process, this is not possible,

Fig. 3 Simulation model's assembly department







especially for a layout that does not accomplish such a configuration.

- 2. The storage or buffer capacities in between successive machines may sometimes be virtually unlimited. Unfortunately, this does not happen with the actual manufacturing process. When the products are physically as large as steel doors, the buffer space in between two successive machines has limited capacity, causing blockage. When this occurs, the job has to remain at the machine, preventing a job in the queue at that machine from starting its processing.
- 3. Any job can be processed at each stage by any machine. Although some workstations in the steel door manufacturing process have parallel machines to process any job on, the impact on performance measures can be totally different using one machine to another. The main reason is because there are a limited number of skilled workers for certain types of doors that use those parallel machines.
- 4. Operations cannot be interrupted. There are many reasons to interrupt the operations at any workstation or machine such as failures, adjustments, wrong setup, scrap, and the sudden arrival of priority jobs.

- 5. Each machine can process only one operation at a time. In the actual manufacturing process, the cure oven system acts as a machine, processing more than 100 doors (from different jobs) at the same time.
- 6. One kind of machine is available. Normally, when the jobs have entered the shop floor, they are produced according to a specific route as in any jobshop configuration; however, there are working groups

Verification of simulation model



Fig. 4 Fixed values test



**Operational Validity** 

Fig. 5 Operational validity

that may not be available due to personnel and shift scheduling.

7. The time to transfer jobs between machines is not relevant. In the steel doors manufacturing shop, most of the

Table 3 Validation method

jobs need to be transferred between machines or workstations by dollies, platforms, or forklifts in order to continue the process, and these transfers take time.

- Processing time is fixed or is known in advance. Most parts require similar processing, although specific, required features for each model cause some variation in actual work content. Due to this uniqueness, parts require varying amounts of resources and processing times.
- 9. All jobs are available at time 0. Normally, not all jobs are available at the beginning of the study horizon. They arrive throughout the study horizon.

In addition, the operating conditions for the steel doors manufacturing process are different and more sensible than classic configurations where operating conditions can be irrelevant. For example:

Output variables	Input variables		
	Real system	Conveyor speed	Conveyor speed
	0.036 m/s	0.03 m/s	0.05 m/s
Avg utilization (%)	17.39	17.97	18.25
Avg busy processing time (s)	40,922	39,975	40,964
Work in process (doors)	84	91	68
Throughput (doors/shift)	916	909	932
	Real system	Sealed silicon	Sealed silicon
	30 min	25 min	35 min
Avg utilization (%)	17.39	18.07	17.91
Avg busy processing time (s)	40,922	35,749	38,815
Work in process (doors)	84	113	118
Throughput (doors/shift)	916	886	881
	Real system	Drying	Drying
	30 min	25 min	35 min
Avg utilization (%)	17.39	18.22	18.15
Avg busy processing time (s)	40,922	40,938	40,871
Work in process (doors)	84	68	62
Throughput (doors/shift)	916	931	930
	Real system (FIFO rule)	Sequences (by complexity)	Sequences (by simplicity)
Avg utilization (%)	17.39	15.23	19.63
Avg busy processing time (s)	40,922	35,516	40,724
Work in process (doors)	84	268	82
Throughput (doors/shift)	916	731	917
	Real system (without off-duty hours)	Shifts (8 off-duty hours)	Shifts (4 off-duty hours)
Avg utilization (%)	17.39	16.97	20.30
Avg busy processing time (s)	40,922	33,515	41,127
Work in process (doors)	84	270	68
Throughput (doors/shift)	916	730	931

Operation		Feas	siblity	
(job, precedence)	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>
O <sub>1,1</sub>	0	0	0	0
O <sub>1,2</sub>	0	0	x	0
O <sub>2,1</sub>	0	х	x	х
O <sub>2,2</sub>	0	х	0	0
O <sub>2,3</sub>	х	0	0	0
O <sub>3,1</sub>	0	х	x	х
O <sub>3,2</sub>	х	0	0	0
O <sub>4,1</sub>	0	х	х	x
O <sub>4,2</sub>	х	о	о	x
O <sub>4,3</sub>	0	0	0	0
Off-duty hours	0	2	4	2

Fig. 6 Feasibility of operation

- a. Storage rules for the work in process on the workstations. Normally, when a job finishes its process through a machine, it continues to the next process in other machines, but some processes need to be done in batches. In such a case, the job needs to wait until a certain number of jobs have already come to the start of the next process. While waiting, the operators stack the jobs and when the process can finally continue, the operators take the jobs one by one from the last to first modifying the original sequence.
- b. Transfer rules for raw material and work in process. Usually, when a job is ready to go to another workstation or machine to start the next process, it simply goes, but sometimes it has to wait until some load rule is satisfied. This is common in the actual manufacturing process, where the door's transfer is done by dollies, which cannot move until there is a minimum of ten doors.
- c. Load rules on the conveyor. Commonly, when a machine is available, it can process another job. In the case of the painting process, an elevated conveyor transfers doors and frames to the cure oven. However, this does not always happen. Although it contains more than one hundred spaces (carriers) to load, they cannot be fully used due to the size of doors or frames. The operators need to allow some spaces in order to prevent contact between doors or frames on conveyor curves. Those carriers are then idle machines.
- d. Startup policies. When some machines are turned on, any job can be processed. There are also certain tools or machines that need to reach some critical parameter,

11

such as the minimum temperature for operations to begin.

e. Capacities. In many cases, operations personnel are available when they need to produce a given job, but the manufacturing process can require different capacities for given machines and workstations. Because of this, personnel scheduling affects the actual capabilities of the manufacturing process.

The steel door manufacturing process contains several workstations associated with different products. Each type of door goes through a different sequence of processing steps; furthermore, the work content varies greatly at each step. Unlike other manufacturing processes, once production starts on the doors, the sequence may be changed, storing it as work in process. When the manufacturing process becomes overcommitted, parts must be constructed with overtime or subcontracted in order to meet upstream demand, resulting in higher costs and longer lead times.

The processing requirements of the jobs are:

- Specific workstations should be used for each job. Nevertheless, this manufacturing process is flexible, and some jobs can be processed in other workstations.
- Each job involves a set of operations.
- Sequences of operations vary noticeably from job to job.

Finally, the objective is to generate schedules of N different jobs that require processing on M workstations based on different work shifts in a flexible manufacturing process of steel doors to obtain the minimum level of work in process through the interaction between simulation and SEDA. Simulation is used to model the facility being studied, while SEDA is used to guide the overall schedule search process to identify the best performing ones.

# 4 Simulation model—for a steel doors manufacturing shop

In order to avoid the use of theoretical assumptions that can hardly be met and to try to capture the real operating conditions on the manufacturing process mentioned, we do not omit the global behavior of the process when solutions are

Fig. 7 Representation of an individual	Operation Sequence - Continous values U[0,1]								Machine assignment					Off-duty hours		uty ′s				
Individual	0.46	0.62	0.25	0.52	0.79	0.86	0.09	0.26	0.79	0.19	2 '	1 2	3	4	4 3	3 2	3 4	1	2 0	12
											Job	1	Job	2	Job :	3 .	Job 4	Ma	chin	e
												·				·		1	2 3	34

Fig. 8 Processing time data and a feasible sequence

	Int J	Adv	Manuf	Technol (	(2014)	) 73:3-	-21
--	-------	-----	-------	-----------	--------	---------	-----

		C	peratio	on		Fixe	ed Proc	cess	sin	gТ	īme	es (	hrs	)		]	
		(job,	preced	ence)		M <sub>1</sub>		M <sub>2</sub>			M <sub>3</sub>			M <sub>4</sub>			
			O <sub>1,1</sub>			2		4			3			5		]	
			O <sub>1,2</sub>			4		4			-			3			
			O <sub>2,1</sub>			3		-			-			-		1	
			O <sub>2,2</sub>			3		-			3			3			
			O <sub>2,3</sub>			-		5			6			2			
			O <sub>3,1</sub>			2		-			-			-		]	
			O <sub>3,2</sub>			-		2			3			3			
		[	O <sub>4,1</sub>			1		-			-			-		]	
			O <sub>4,2</sub>			-		3			3			-			
			O <sub>4,3</sub>			6		4			5			5			
	Ope	eration	Seque	ence					N	/la	chin	ie a	ssi	gnr	ner	- nt	
 2	4	2	1	2	4	4	3	1	1	1	3	3	1	4	1	4	4
									<u>, 1</u>		lob	2		h 3		loh	1

proposed. If this happens, it is not possible to ensure real solutions that can be implemented. Therefore, we built a simulation model that emulates the manufacture of the most important kinds of doors sold by the company. The model contains all the tasks performed by different machines in the manufacturing process for each sort of product that is scheduled. Our simulation model included many types of details that a manufacturing process presents: setup time, maintenance programming, load and unload processing, packing and unpacking materials, transferring of parts between departments, labor requirements for each machine or process, storage rules in buffers, delay time in some areas, shifts, breaks, and meals. All of these situations are present in the given manufacturing process. Figure 3 shows a side of the final assembly department.

Nevertheless, even if we were able to integrate operation times and workflows in the model, it does not mean the model is faithful to and representative of the real manufacturing process. For this reason, we verified and validated the simulation model.

We ran the simulation model under different conditions to determine if its computer programming and implementation are correct as an applying verification technique which is known as the fixed values test [17, 18]. We decided to verify the throughput as model results. It was verified against calculated a priori values. Figure 4 depicts previous descriptions in three production departments.

The validation method developed for this case study was based on Pita and Wang's [19] work which classifies the input variables as a function of how they affect output variables. The selected input variables are conveyor speed in painting, process time of the sealed system in assembly, process time of the drying system in washing, sequences, and work shifts. The output variables are utilization average of all employees, effective process time average of all employees, work in process, and throughput. Table 3 shows the main results.



3



From the results shown above, the work sequences and off-duty hours before starting the work shift are a key opportunity to improve the manufacturing process. This is mainly because the amount of releases is not constant or sufficient to keep the manufacturing process busy for a long time, causing intermittent idle times. This characteristic is sensitive to the actual manufacturing process. A good balance of off-duty hours helps to improve the output variables values shown in the validation method described above, and it includes the WIP. A combined approach using work sequences, and off-duty hours was the purpose of this research.

Furthermore, the validation of the simulation model was realized statistically. We compared the results derived by the model with real production in the same initial conditions, satisfying statistical assumptions in the validation. Figure 5 depicts the information below.

# 5 SEDA—for a steel doors manufacturing shop

# 5.1 Solution representation

Any solution to the manufacturing process mentioned should be a combination of operation scheduling decision, machine assignment, and off-duty hours before starting the work shift. Thus, a solution can be expressed by the processing sequence of operations on the machines, and the assignment of operations on machines and the off-duty hours, which would be the operators before starting their activities. In this paper, a solution is represented by three vectors (operation sequence vector, machine assignment vector, and off-duty hours vector).

For the operation sequence vector, the number of elements equals the total number of operations, where each element contains a random value U[0,1], an important difference



between our approach and Wang et al.'s [6] work. For the machine assignment vector, each element represents the corresponding selected machine for each operation. For the offduty hours vector, each element shows the off-duty hours, which would be the operators before starting their activities on the most important manufacturing departments on the shop. To explain the representation, we provide an example by considering a problem with four jobs, four machines, and

Fig. 13 Communication SEDAsimulation model

different off-duty hours possible as shown in Fig. 6. In Fig. 7, the representation of an individual is illustrated.

To show a comparison between this approach and Wang et al.'s [6] work, an example is provided. Figure 8 details the fixed processing time for each job on each machine and a feasible sequence. Figure 9 illustrates the Gantt chart of this solution working at the same time on all machines. Figure 10 shows another feasible sequence, but it includes off-duty





Fig. 14 Integration SEDA and simulation model

hours and its impact over work in process. Idle time can be noted in Fig. 11. However, in the industrial environment mentioned, the processing time is not fixed. Therefore, Wang et al.'s [6] representation should not be directly applied. The results may be totally different. Because of this, a simulation model is used to tackle this drawback.

However, it is possible that, when a given job reaches a workstation that remains off-duty, the job will have to wait until the workstation starts production again, converting the job to WIP. We sought as little WIP as possible, given the operational, physical, and programmatic restrictions.

# 5.2 Generation of the population

Initial population members are generated randomly in order to enable a wide range of solutions [20].

Table 4 Comparison of results for each average

Trial	GA Best	SEDA Best	$GA$ $\mu_1$	$\begin{array}{c} \text{SEDA} \\ \mu_2 \end{array}$	$\mu_1 - \mu_2$	$H_0: \mu_1 = \mu_2$ $\alpha = 0.10$ $ Z_c  \le 1.645$
1	2.66	3.50	4.97	6.37	-1.40	16.64*
2	3.21	2.56	5.30	5.35	-0.05	0.58
3	2.93	0.91	5.29	5.63	-0.34	3.37*
4	3.31	0.00	5.23	4.14	1.09	11.47*
5	2.66	0.03	5.01	4.09	0.92	9.175*
6	2.66	0.00	4.95	4.15	0.80	8.160*
7	2.66	0.00	4.90	4.16	0.74	7.580*
8	2.66	0.00	5.03	4.19	0.85	8.450*
9	2.66	0.39	5.01	4.15	0.87	8.785*
10	2.66	0.03	4.91	4.09	0.81	8.381*
11	2.66	0.00	4.98	4.20	0.78	7.931*
12	2.66	0.03	4.98	4.05	0.93	9.447*
13	2.66	0.03	5.08	4.14	0.94	9.402*
14	2.66	0.00	5.02	4.14	0.87	8.906*
15	2.66	0.03	4.96	4.11	0.85	8.994*
16	2.66	0.00	4.98	4.18	0.80	8.430*
17	2.66	0.00	5.05	4.06	0.99	9.730*
18	2.66	0.21	5.09	4.10	0.99	9.946*
19	2.66	0.00	5.02	4.14	0.88	8.927*
20	2.66	0.00	4.98	4.24	0.74	7.620*
21	2.66	0.03	5.07	4.32	0.76	7.390*
22	2.66	0.00	4.89	4.17	0.72	7.603*
23	2.66	0.00	4.97	4.17	0.80	8.248*
24	2.90	0.00	4.89	4.31	0.58	5.943*
25	2.66	0.03	5.02	4.21	0.81	8.089*
						24 / 25

### 5.3 Probability model

SEDA contains three different graph models. The first graph model aims to determine an estimation of distribution model to generate new offspring (operation sequence) using a subset of 'm' selected sequences (individuals). In order to do this, we adopted a continuous optimization procedure instead of a discrete one to solve the scheduling problem. This is an important difference between this approach and the current research. The advantage of this representation for each individual, through continuous values, is that they do not have direct meaning to the solution they represent. There is no problem if each individual does not explicitly shows its information on the sequence of jobs to be processed. It is not necessary that the probabilistic model be updated each time a job is assigned in the sequence, and it is not necessary to make any modification in the sampling process. Rudolph [21]

Table 5 Comparison of results for each variance

Trial	GA Best	SEDA Best	$\operatorname{GA}_{\sigma_1^2}$	$\begin{array}{c} \text{SEDA} \\ \sigma_2^2 \end{array}$	$\sigma_1^2/\sigma_2^2$	$H_0: \sigma_1^2 = \sigma_2^2 \\ \alpha = 0.01 \\ F_c < 0.545 \text{ or} \\ F_c > 1.832$
1	2.66	3.50	2.926	2.903	1.0079	1.0079*
2	3.21	2.56	3.124	2.661	1.1738	1.1738*
3	2.93	0.91	2.763	5.553	0.4976	0.4976
4	3.31	0.00	2.377	5.028	0.4727	0.4727
5	2.66	0.03	3.307	4.933	0.6704	0.6704*
6	2.66	0.00	2.848	5.050	0.5640	0.5639*
7	2.66	0.00	2.482	5.284	0.4698	0.4698
8	2.66	0.00	3.116	5.150	0.6050	0.6050*
9	2.66	0.39	2.977	5.053	0.5891	0.5891*
10	2.66	0.03	2.932	4.828	0.6072	0.6072*
11	2.66	0.00	2.803	5.149	0.5443	0.5443
12	2.66	0.03	3.055	4.881	0.6258	0.6258*
13	2.66	0.03	3.267	5.004	0.6529	0.6529*
14	2.66	0.00	2.969	4.990	0.5950	0.5950*
15	2.66	0.03	2.743	4.601	0.5962	0.5962*
16	2.66	0.00	2.897	4.541	0.6380	0.6380*
17	2.66	0.00	3.223	5.323	0.6055	0.6055*
18	2.66	0.21	3.214	4.936	0.6511	0.6511*
19	2.66	0.00	3.049	4.935	0.6178	0.6178*
20	2.66	0.00	2.759	5.036	0.5478	0.5478*
21	2.66	0.03	3.003	5.662	0.5303	0.5303
22	2.66	0.00	2.581	4.919	0.5246	0.5246
23	2.66	0.00	2.913	4.891	0.5955	0.5955*
24	2.90	0.00	2.704	5.152	0.5248	0.5248
25	2.66	0.03	3.006	5.176	0.5807	0.5807*
						18 / 25

 $\mu$  the average

 $\sigma^2$  the average

Table 6	Comparison	of results	for each	algorithm

Trial	GA Best	SEDA Best	GA Worst	SEDA Worst	GA Reliability	SEDA Reliability
1	2.66	3.50	11.05	11.28	1	3
2	3.21	2.56	11.19	11.18	1	1
3	2.93	0.91	11.28	11.04	1	1
4	3.31	0.00	10.85	11.19	1	2
5	2.66	0.03	11.05	10.91	1	4
6	2.66	0.00	11.05	11.19	1	4
7	2.66	0.00	11.05	11.19	1	3
8	2.66	0.00	11.05	11.19	1	2
9	2.66	0.39	10.96	11.19	3	5
10	2.66	0.03	11.05	11.19	3	2
11	2.66	0.00	11.05	11.19	1	2
12	2.66	0.03	11.05	11.05	1	7
13	2.66	0.03	11.05	11.19	1	6
14	2.66	0.00	11.05	11.19	1	3
15	2.66	0.03	10.89	10.91	1	2
16	2.66	0.00	10.96	11.19	2	2
17	2.66	0.00	11.05	11.05	2	2
18	2.66	0.21	11.05	11.19	2	3
19	2.66	0.00	11.05	10.43	1	2
20	2.66	0.00	10.96	11.19	1	2
21	2.66	0.03	11.05	11.19	2	1
22	2.66	0.00	10.89	11.19	1	2
23	2.66	0.00	11.05	11.19	1	3
24	2.90	0.00	10.96	11.05	1	4
25	2.66	0.03	11.05	11.05	2	4
					34 / 25	72 / 25

Reliability: number of times that the best value is obtained

and Bean and Norman [22] can be consulted about continuous optimization procedures. We used the  $MIMIC_C^G$  algorithm to



build the first probabilistic graph model introduced by Larrañaga et al. [23]. It is an adaptation of the MIMIC algorithm presented by De Bonet et al. [24] to continuous domains. Once the individuals have been generated from the algorithm  $\text{MIMIC}_{C}^{G}$ , they must be decoded to be represented as a valid operation sequence. Thus, we need a method to decode these real vectors into discrete vectors. A fixed integer number is assigned for each operation. Each fixed integer number is associated with a job. A sort on the continuous values of each individual is done. Assigning each continuous value to the corresponding fixed integer number that belongs to each operation and setting each fixed integer number to a job to finish. Figure 12 details an example of a real vector and its decoding.

The second probabilistic graph model aims to determine an estimation of distribution model to generate new offspring (machine assignment) using a subset of 'm' selected assignments (individuals). To obtain the estimation, we used the COMIT algorithm introduced by Baluja and Davies [25].

The third probabilistic graph model aims to determine an estimation of distribution model to generate new offspring (off-duty hours work shift) using a subset of 'm' selected offduty hours work shift (individuals). Although the number of hours is defined for any shift based on current legal guidelines, the start time of the shift is not, so we aim to generate individuals representing the off-duty hours that would be the operators before starting their activities. Again, we used the COMIT algorithm to obtain the estimation.

### 5.4 Diversity

SEDA uses a Tabu search method in order to avoid losing diversity on the evolutionary progress. The Tabu search method is based on Pinedo's [26] research.







5.5 SEDA–simulation model

The output of the simulation model, the work in process level of the steel doors manufacturing process, is used by SEDA to provide feedback on the progress of the search for the best solution. This in turn guides further input to the simulation model.

In order to get communication between SEDA—written in DevC++®—and the simulation model—built on Delmia Quest® R20—and to get feedback on progress, some commands from the Batch Control Language (BCL) provided from Delmia Quest® were used for the objective. Figure 13 details the overall process.

Figure 14 shows the simulation optimization approach proposed.

# 6 Results and comparison

SEDA was built to be functional on the specific manufacturing process mentioned; therefore, we consider evaluating the SEDA in the same industrial environment.

A GA is proposed as a benchmark for comparison with the SEDA scheme. GA works with tournament selection. The

"edge recombination operator" is used as a cross operator based on Whitley et al. [27], and a mutation operator changes jobs among different positions.

We used a Dell<sup>®</sup> Vostro<sup>®</sup> 3500 computer, Intel<sup>®</sup> Core<sup>TM</sup> i3 processor, 2.6 GHZ, 4 GB of RAM, Windows<sup>®</sup> 7 for 64 bits to run each algorithm.

To account for the stochastic nature of the shop, we ran 25 trials for both algorithms. Each trial contains 11 generations; 75 individuals belong to each generation.

We established a workload to evaluate and find the best schedule. Our experiments were based on production of 1,000 doors. The workload mentioned contains different orders, due dates, and kinds of doors produced in a workweek, replicating the actual manufacturing process. Each production order corresponding includes different numbers of jobs. The arrival times of the production orders are indeterminable.

As a response variable for the experiment, we measure the relative percentage increase (RPI)

RPI 
$$(c_i) = (c_i - c^*)/c^* \times 100$$
 (10)

where  $c_i$  is the work in process value obtained in the *i*th replication by a given algorithm configuration, and  $c^*$  is the



Fig. 17 Performance GA





best objective value found by any of the algorithm configurations. Note that for this problem, there are no known effective exact techniques and comparing against an optimum solution is not possible.

Table 4 details the average obtained for each trial. We analyze whether there is a statistically significant difference between averages of both algorithms.

As we can see in Table 4, there is a statistically significant difference between the averages of both algorithms. The performance of SEDA was superior in 24 of the 25 trials with  $\alpha$ = 0.10 of significance level.

Table 5 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

As we can see in Table 5, there is no statistically significant difference between variances of both algorithms. The performance was the same in 18 of the 25 trials with  $\alpha$ =0.01 of significance level. We consider that the stability of both algorithms is practically the same (72 % of the time).

Table 6 details the reliability obtained for each trial. We analyze the amount of times that both algorithms got the best value in each trial.

As we can see in Table 6, there is difference between the reliability of both algorithms. The performance of SEDA was better in 111 %. It found 38 more times better value than GA.

Figure 15 shows the individuals generated by GA in the trails 5, 10, 15, and 20 through a percentiles chart.

For all trials, GA would converge and got its best value for the response variable, RPI at around 3.02

Figure 16 shows the individuals generated by SEDA in the same trials through a percentiles chart.

SEDA's trials were different; these converged and got its best value below 3.00 for the response variable, RPI.

Figure 17 shows the performance by GA on each trial.

In all, GA's trials had close median values, around 5 RPI, and the search was concentrated between 2.5 and 6 RPI. The plus symbol (+) indicates where the median is located.

Figure 18 shows the performance by SEDA in each trial.

SEDA's performance was different. It got median values around 4 RPI, but it could achieve better values against GA in almost all trials. SEDA could search in a more promising area. The plus symbol (+) indicates where the median is located.

The experimental results were analyzed using the analysis of variance (ANOVA) method. In the experiment, the main assumptions were checked and accepted. Table 7 details that there is a statistically significant difference between the algorithms.

Figure 19 shows the overall behavior between both algorithms, and Fig. 20 plots the performance metric on each trial.

### Table 7 Analysis of variance

ANOVA									
Source of variation	SS	df	MS	F	<i>p</i> value	F crit			
Sample	50.3280	1	50.328	19.0555	1.28934E-05	3.84288			
Generations	4,846.39	43	112.706	42.6738	0	1.38108			
Interaction	2,660.61	43	61.874	23.4273	5.274E-169	1.38108			
Within group	17,199.0074	6,512	2.641						
Total	24,756.344	6,599							



# 7 Conclusions

Although the traditional scheduling problem has been solved from an academic perspective, its implementation in industrial environments has been a difficult task because there are many assumptions to satisfy. Based on the experimental results shown, we confirmed that an appropriate modeling of the most important variables that affect the performance of the process should be considered in the proposed solution. We reach the conclusion that the shop performance can be improved if different machines are assigned to different schedules. Although some assumptions of the FJSP could not be met in the manufacturing process studied, because a number of the assumptions are not consistent, the simulation model was able to tackle this situation by incorporating the real operation conditions of the manufacturing process. The reason is that these are more sensitive than classical configurations. The validation method allowed for identification of the key opportunity to improve the manufacturing process, that is, work sequences and off-duty time for each machine.



Fig. 20 Performance metric for each trial

Using a continuous EDA was not necessary to make any modifications in the sampling process in the processing sequence of operations on the machines, as is generally required by other algorithms. It allowed for better trust in the data against the GA. We consider using three graphical models, while most EDAs do not apply more than one model. It permitted handling the most important variables of the manufacturing process studied in the sampling mechanism, which was refined by the Tabu search method. This strategy was created to avoid losing diversity in the evolutionary progress of the algorithm. We conclude that the simulation optimization can be an efficient mechanism to handle different manufacturing conditions where there are diverse variable interactions such as the FJSP.

Finally, this research contributes using an EDA as an optimization method for any simulation language.

Acknowledgments We would like to express our gratitude to Jon Fournier and Martin Barnes for their technical assistance on the Quest<sup>®</sup> platform.

### References

- Mühlenbein H, Paaß G (1996) From recombination of genes to the estimation of distributions: I. Binary parameters. Parallel problem solving from nature PPSN IV, 178–187
- Chen SH, Chen MC, Chang PC, Zhang Q, Chen YM (2010) Guidelines for developing effective estimation of distribution algorithms in solving single machine scheduling problems. Expert Syst Appl 37:6441–6451
- Liu H, Gao L, Pan Q (2011) A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem. Expert Syst Appl 38:4348–4360
- Pan QK, Ruiz R (2012) An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega 40:166– 180
- Chen SH, Chang PC, Cheng T, Zhang Q (2012) A self-guided genetic algorithm for permutation flowshop scheduling problems. Comput Oper Res 39:1450–1457
- Wang L, Wang S, Xu Y, Zhou G, Liu M (2012) A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. Comput Ind Eng 62:917–926
- Chen YM, Chen MC, Chang PC, Chen SH (2012) Extended artificial chromosomes genetic algorithm for permutation flowshop scheduling problems. Comput Ind Eng 62:536–545
- Jarboui V, Eddaly M, Siarry P (2009) An estimation of distribution algorithm for minimizing the total flow time in permutation flowshop scheduling problems. Comput Oper Res 36:2638–2646

- Zhang Y, Li X (2011) Estimation of distribution algorithm for permutation flow shops with total flowtime minimization. Comput Ind Eng 60:706–718
- Chan FTS, Wong TC, Chan LY (2006) Flexible job-shop scheduling problem under resource constraints. Int J Prod Res 44(11):2071– 2089
- Wadhwa S, Rao KS, Chan FTS (2005) Flexibility-enabled lead-time reduction in flexible systems. Int J Prod Res 43(15): 3131–3162
- Chan FTS, Chung SH (2004) A multi-criterion genetic algorithm for order distribution in a demand driven supply chain. Int J Comput Integr Manuf 17(4):339–351
- Chan FTS (2001) The effects of routing flexibility on a flexible manufacturing system. Int J Comput Integr Manuf 14(5):431-445
- Peña J, Robles V, Larrañaga P, Herves V, Rosales F, Pérez M (2004) Ga-eda: hybrid evolutionary algorithm using genetic and estimation of distribution algorithms. Innovations in Applied Artificial Intelligence. Berlin/Heidelberg
- Shim VA, Chen Tan K, Yong Chia J, Kiat Chong J (2011) Evolutionary algorithms for solving multi-objective travelling salesman problem. Flex Serv Manuf J 23:207–241
- 16. Yan Y, Wang G (2007) A job shop scheduling approach based on simulation optimization. Proceedings of the 2007 I.E. IEEM
- Sargent R (2007) Verification & validation of simulation models. Proceedings of the 2007 Winter Simulation Conference
- Whitner R, Balci O (1989) Guidelines for selecting and using simulation model verification techniques. Proceedings of the 1989 Winter Simulation Conference. Washington
- Pita J, Wang Q (2010) A simulation approach to facilitate manufacturing system design. Int J Simul Model 9–3:152–164
- 20. Greenwood A, Vanguri S, Eksioglu B, Jain P, Hill T, Miller J, Walden C (2005) Simulation optimization decision support system for ship panel shop operations. Proceedings of the 2005 Winter Simulation Conference
- Rudolph G (1991) Global optimization by means of distributed evolution strategies. Parallel problem solving from nature PPSN I. Lectures notes in computer science
- 22. Bean J, Norman B (1993) Random keys for job shop scheduling problem. The University of Michigan
- 23. Larrañaga P, Exteberria R, Lozano J, Peña J (2000) Optimization in continuous domains by learning and simulation of Gaussian networks. Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program
- 24. De Bonet J, Isbell C, Viola P (1997) MIMIC: finding optima by estimation probability densities. Adv Neural Inf Process Syst 9
- 25. Baluja S, Davies S (1997) Combining multiple optimization runs with optimal dependency trees. Carnegie Mellon University
- Pinedo M (2001) Scheduling. Springer Science + Business Media, New York
- Whitley D, Starweather T, Shaner D (1990) The traveling salesman and sequence scheduling: quality solutions using genetic edge recombination. In: L. Davis (Ed.), Handbook of genetic algorithms, (pp. 350–372). New York