Arab J Sci Eng (2013) 38:3593–3604
DOI 10.1007/s13369-013-0625-y

RESEARCH ARTICLE - SYSTEMS ENGINEERING

Development of Both the AIS and PSO for Solving the Flexible Job Shop Scheduling Problem

Amir Sadrzadeh

Received: 18 December 2011 / Accepted: 3 July 2013 / Published online: 24 July 2013

Abstract The flexible job shop scheduling problem (FJSP) is to assign each operation to an appropriate machine and to sequence the operations on the machines. The paper describes the development and the application of the artificial immune system (AIS) and the particle swarm optimization (PSO) for solving the flexible job shop scheduling problem with sequence-dependent setup times (SDST-FJSP). A series of the experiments have been designed using the analysis of variance to recognize best settings of parameters. Finally, 30 examples of the different sizes in the SDST-FJSP with the objective of minimizing makespan and mean tardiness have been used to verify the performance of the proposed algorithms, and to compare them with the existing meta-heuristic algorithms in the literature, such as the genetic algorithm (GA), the parallel variable neighborhood search (PVNS), and the variable neighborhood search (VNS). The obtained results show that the proposed PSO outperforms the GA and the PVNS approaches. It is found that the average best-so-far solutions obtained from the proposed AIS are better than those produced by the GA, the PVNS, the VNS, and the PSO algorithms for all the examples.

Keywords Bi-criteria scheduling · Sequence-dependent setup time · Artificial immune system · Particle swarm optimization

A. Sadrzadeh (✉)
Department of Industrial Engineering,
Urmia University of Technology, Urmia, Iran
e-mail: a.sadrzadeh@ine.uut.ac.ir; amir.sadrzade@gmail.com

الخلاصة

إن مشكلة جدولة ورشة العمل المرنة (FJSP) هي في تعيين كل عملية للجهاز المناسب، وفي تسلسل العمليات على الآلات ، وتصف الورقة العلمية تطوير وتطبيق النظام المناعي الاصطناعي (AIS) وتحسين سرب الجسيمات (PSO) من أجل حل مشكلة جدولة ورشة العمل المرنة مع أوقات الإعداد التي تعتمد على التسلسل (FJSP-SDST) ، وتم تصميم سلسلة من التجارب باستخدام تحليل التباين (ANOVA) للتعرف على أفضل إعدادات للمعاملات ، وأخيرا تم استخدام ثلاثين مثالا من أحجام مختلفة في FJSP-SDST بهدف تقليل وقت التشغيل ، وتم استخدام متوسط التأخر للتحقق من أداء الخوارزميات المقترحة، ومقارنتها بخوارزميات الكشف عن مجريات الأمور الفوقية الموجودة في الأدب، مثل الخوارزمية الجينية (GA)، وبحث الجوار المتغير الموازي (PVNS)، وبحث الجوار المتغير (VNS)، وتشير النتائج التي تم الحصول عليها إلى أن تحسين سرب الجسيمات المقترح يتفوق على نهوج GA و PVNS ، ووجد أن أفضل متوسط للحلول تم الحصول عليه من AIS المقترحة هو أفضل من ذاك الذي تنتجه خوارزميات GA، PVNS، VNS، و PSO لجميع الأمثلة.

1 Introduction

The job shop scheduling problem (JSP) plays a key role in the manufacturing system, and also it has been considered by prior researchers. As an extension of the classical JSP, the flexible job shop scheduling problem (FJSP) is to assign each operation to an appropriate machine and to sequence the operations on the machines [1,2].

Considering the assumption of sequence-dependent setup times (SDST) in the real world scheduling settings, the setup operations must be performed after completing the process of one job and before beginning the process of the next job [3–6], and also the sequence of the jobs significantly affects the setup times [7].

The flexible job shop scheduling problem with sequence-dependent setup times (SDST-FJSP) belongs to the class of the NP-hard problems [1,8], i.e., the amount of required computation increases exponentially with the problem size.

The meta-heuristic algorithms are capable of obtaining optimal or near optimal solutions for this type of problems in an acceptable computational time. With due attention to the literature review related to the application of the meta-heuristic algorithms in the SDST-FJSP, it is understandable that few studies have addressed the SDST-FJSP. Imanipour [9] modeled the SDST-FJSP as the nonlinear mixed integer programming and used the tabu search to minimize the makespan. Saidi-Mehrabad and Fattahi [10] also developed the tabu search for solving the SDST-FJSP to minimize the makespan and compared their results with the results of the LINGO software. Bagheri and Zandieh [1] applied the variable neighborhood search (VNS) algorithm based on an integrated approach for solving the SDST-FJSP. These authors adjusted the parallel variable neighborhood search (PVNS) algorithm of Yazdani et al. [11] and the genetic algorithm (GA) of Pezzella et al. [12] to the SDST-FJSP as the previous meta-heuristic algorithms used in the FJSP literature. Finally, Bagheri and Zandieh [1] verified the efficacy of the VNS algorithm in comparison with the two mentioned algorithms.

Among the different stochastic search methods, the particle swarm optimization (PSO) and the artificial immune systems (AIS) have been successfully used to solve the complex combinatorial optimization problems [13,14]. However, no research has been reported that has used these algorithms to solve the SDST-FJSP; as a result, the present paper develops both the AIS and PSO for solving the SDST-FJSP. In order to demonstrate the effectiveness of the proposed algorithms, it has been used on 30 examples to compare with the VNS, the PVNS and the GA algorithms.

The remainder of this paper has been organized as follows: the SDST-FJSP is presented in Sect. 2. The procedures of the AIS and the PSO are described in Sects. 3 and 4, respectively. Section 5 considers the design of experiment (DOE) to investigate the proper settings of the AIS parameters. In order to verify the efficiency of the proposed algorithms, Sect. 6 illustrates the comparative studies with the other approaches. Finally, the conclusion is drawn in Sect. 7.

2 The SDST-FJSP

In the flexible job shop scheduling problem with sequence-dependent setup times (SDST-FJSP), each job is produced by a predetermined sequence of the operations one after another. This problem, which has been developed by Bagheri and Zandieh [1], works as assigning each operation to an available machine and sequencing the assigned operations on all the machines. The assumptions and the objective functions in this problem are summarized as below:

2.1 Assumptions

1. Jobs are independent of each other.
2. The precedence constraints must be considered among the operations of the same job.
3. Each operation must be completed without interruption.
4. At a given time, a machine can only execute one operation.
5. Machines are independent of each other.
6. Setup times are dependent on the sequence of jobs.
7. A dummy job without the setup and processing times is performed before actual starting process of operations executed on each machine.
8. The processing and setup times are deterministic and preselected.
9. All jobs and machines are available at time 0.

2.2 Objective Functions

Production scheduling can be characterized by the objective functions of minimizing makespan, mean tardiness, maximum lateness, maximum machine workload, and total workload. Among them, the makespan criterion has been studied more by prior researchers [1]. The makespan ($C_{\max}$), denoting the period required to complete all jobs [15], can be calculated using Eq. (1):

$$F_1 = C_{\max} = \text{Max}\{C_1, C_2, \ldots, C_n\} \tag{1}$$

where C_i is the completion time of job i and n is the total number of the jobs.

The mean tardiness for all jobs is shown as follows [1]:

$$F_2 = \frac{\sum_{i=1}^{n} \text{Max}\,\{C_i - d_i, 0\}}{n} \tag{2}$$

where d_i is the due date of job i.

In this study, the objective function (Z) achieves minimization of makespan and mean tardiness, which can be formulated as follows:

$$\text{Minimize } Z = \alpha \times F_1 + (1-\alpha) \times F_2, \quad 0 \le \alpha \le 1 \tag{3}$$

where α is the relative importance of makespan and mean tardiness.

3 AIS Procedures for the SDST-FJSP

The immune system uses two types of response mechanisms, including the innate (non-specific) and the acquired (specific) [16,17], to protect body from foreign pathogens [18]. The innate immune response is obtained through the evolution from generation to generation, and the acquired immune response is learned through its own encounters with the antigens [19].

The artificial immune system (AIS) [20] was invented by Farmer et al. [13] as a new computational intelligence

Fig. 1 Pseudo-code of the proposed AIS

```
Input the value of the AIS parameters (the number of antibodies (NA) and the antibody
elimination percentage (%B));
Generate a population of the NA antibodies;
While the stopping criteria are not met:
        Calculate the affinity value (fitness) of each antibody (solution) in the population;
        For each antibody,
                Apply the inverse mutation operator to create a better antibody;
                If the mutated antibody is not better than the original antibody,
                        Apply the pair-wise interchange mutation operator to create a
                        better antibody;
        Eliminate the worst %B of the antibodies in the population;
        Create randomly the new antibodies to replace the eliminated antibodies;
```

technique [21]. The AIS has been developed to give the defensive properties within a computing context [22], and has been studied widely in the fields of the artificial intelligence due to its deep inspiration to real world sciences and engineering problems [8,19,23]. The AIS models can be listed as follows [24]: The clonal selection [25], the immune networks [13], the danger theory [26], the negative selection [27], the bone marrow, and also the somatic hypermutation. Among these models, the clonal selection together with the affinity maturation processes [28] has been applied to explain how the immune system responds to the pathogens, and how it improves its capability of killing the invaders [29–31].

The pseudo-code of the proposed AIS with the clonal selection and the affinity maturation processes for the SDST-FJSP are outlined in Fig. 1. The main processes of the AIS (see Fig. 2) are described in the following subsections:

3.1 Problem Encoding

A solution in the FJSP can be expressed by the assignment of operations to machines, and the sequence of the assigned operations on machines [1]. In the present paper, a linear encoding known as the task sequencing list [1,32] has been used to represent the antibody (solution). The encoded sub-antibody has been shown by the triple (i, j, k), where i, j and k are the number of the job, the operation and the machine, respectively. The length of an antibody is equal to the total number of the operations for all jobs. The left-to-right ordering of the sub-antibodies indicates the sequence of the operations on the machines. Figure 3 shows a typical antibody representation for an example with four jobs and three machines.

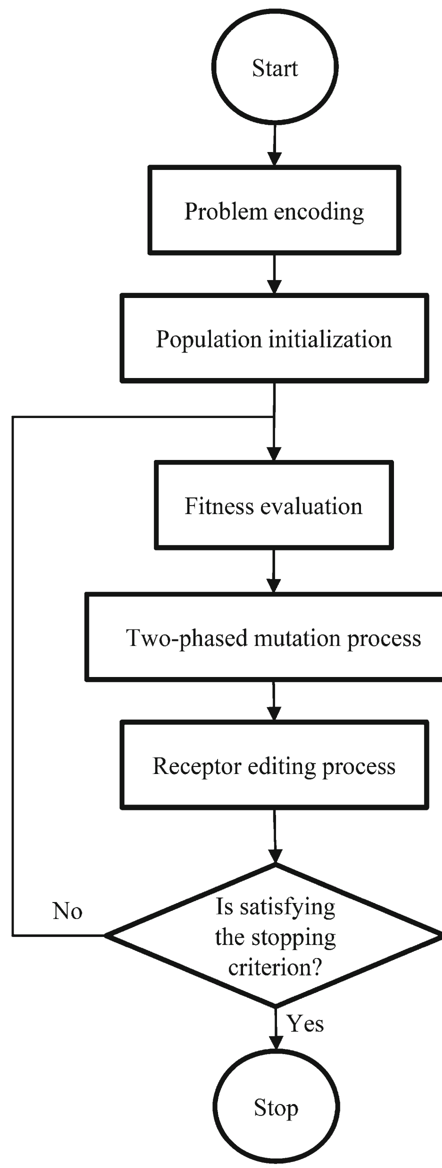


Fig. 2 Proposed AIS procedures for solving the SDST-FJSP

3.2 Population Initialization

The proposed AIS produces the encoded sub-antibodies with the random assignment of the operations to the appropriate machines. The created sub-antibodies are arranged according to the precedence constraints (assumption #2 in the Sect. 2.1) to generate the initial population of the antibodies.

(1,1,1)	(2,1,3)	(3,1,2)	(3,2,1)	(1,2,2)	(2,2,3)	(4,1,1)	(4,2,3)	(2,3,1)	(4,3,2)	(3,3,3)	(2,4,2)

Fig. 3 Antibody representation

Fig. 4 Pseudo-code of the proposed PSO

```
Input the value of the PSO parameters (the number of the particles (M), the inertia weight
(w(t)), and the learning factors (c1 and c2));
Generate a swarm of the M particles;
While the stopping criteria are not met:
        Calculate the fitness value of each particle in the swarm;
        For each particle,
                Calculate the best fitness of particle (local best);
                Calculate the best fitness of the all particles (global best);
                Update the particle velocity using the swap and insert operators based
                on the local best and the global best;
                Update the particle position using the swap and insert operators based
                on the new velocity;
```

3.3 Fitness Evaluation

With the clonal selection, each antibody (solution) has an affinity value (fitness) [33]. The affinity value of the antibodies is determined by the bi-criteria performance measure, including minimization of makespan and mean tardiness [see Eq. (3)].

3.4 Two-Phased Mutation Process

As one of the affinity maturation processes, the mutation is a critical process of the AIS and is applied to determine the quantity of exploration within the search space [33]. The two-phased mutation process is an inverse mutation operator followed by a pair-wise interchange mutation operator and is used to generate a clone (new offspring) from an antibody [34]. The number of clones is determined by the number of antibodies (NA) and the affinity value of the antibody [33].

The inverse mutation operator is implemented once, to each antibody in the population [24]. After random selection of two sub-antibodies, a clone is obtained by inverse replacement of the sequence of the sub-antibodies between the two selected sub-antibodies. The mutated clone replaces the original antibody if it presents a better performance. Otherwise, the pair-wise interchange mutation operator is applied to produce a clone with a better affinity value by interchanging the two selected sub-antibodies. The two-phased mutation process is repeated until all the antibodies are mutated.

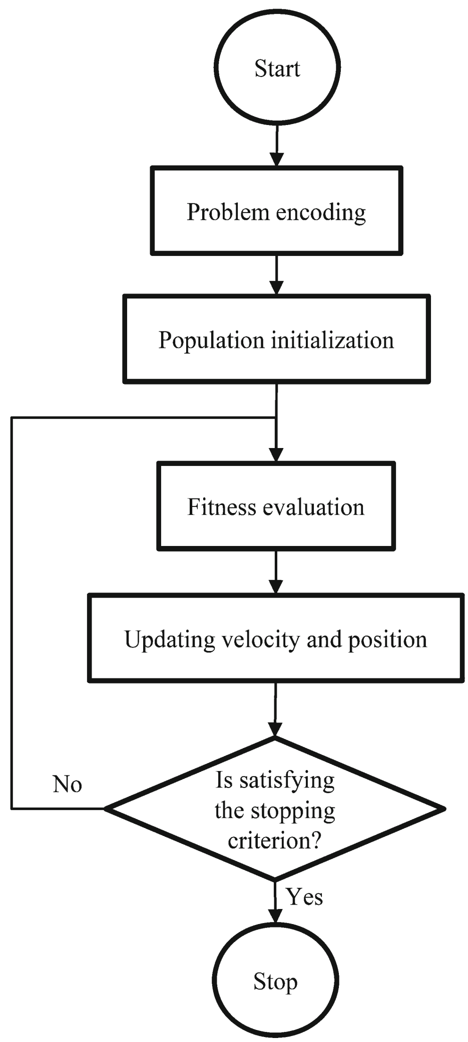


Fig. 5 Proposed PSO procedures for solving the SDST-FJSP

3.5 Receptor Editing Process

As another process of the affinity maturation processes, the receptor editing with the antibody elimination percentage (%B) permits exploring more promising regions of the search space [24]. A fraction of the antibodies (worst %B of the whole population) are eliminated and replaced by the same number of randomly created new antibodies.

3.6 Stopping Criterion

The stopping criterion of the algorithm is determined by a specified CPU time limit. The algorithm steps are repeated until this criterion is satisfied.

4 PSO Procedures for the SDST-FJSP

The swarm intelligence is based upon a simulation of the social behavior such as bird flocking to a promising region for food [35,36]. The application of the swarm intelligence in the optimization was first designed by Eberhart and Kennedy [37] under the name of the PSO [38–40]. The PSO is computationally flexible, efficient, robust and adaptive [41,42], and has great capability of escaping local optimum solution [43–45]. Like the evolutionary algorithms, the PSO conducts the searching process using a swarm (population) of particles (individuals) [46–48]. During exploration for the optimum solution, each particle moves to the next position by conducting a velocity in the direction of its own best previous position (local best) and the best position discovered by the whole particles (global best) [33,49,50].

The N-dimensional position and velocity vectors for the pth particle in the tth iteration can be described as $X_p(t) = x_{p1}(t), x_{p2}(t), \ldots, x_{pN}(t)$ and $V_p(t) = v_{p1}(t), v_{p2}(t), \ldots, v_{pN}(t)$, respectively [48,51]. The following equations can represent the updating rules of a swarm of the particles [52]:

$$V_p(t) = w(t) \times V_p(t-1) + c_1 \times r_1 \times (X_p^L - X_p(t-1)) + c_2 \times r_2 \times (X^G - X_p(t-1)) \quad (4)$$

$$X_p(t) = V_p(t) + X_p(t-1) \quad (5)$$

Table 1 Experimental factors and levels considered

Factors	Levels	Value
Number of the antibodies (NA)	4	100, 500, 1,000, 1,500
Antibody elimination percentage (%B)	7	5, 10, 25, 40, 50, 70, 85

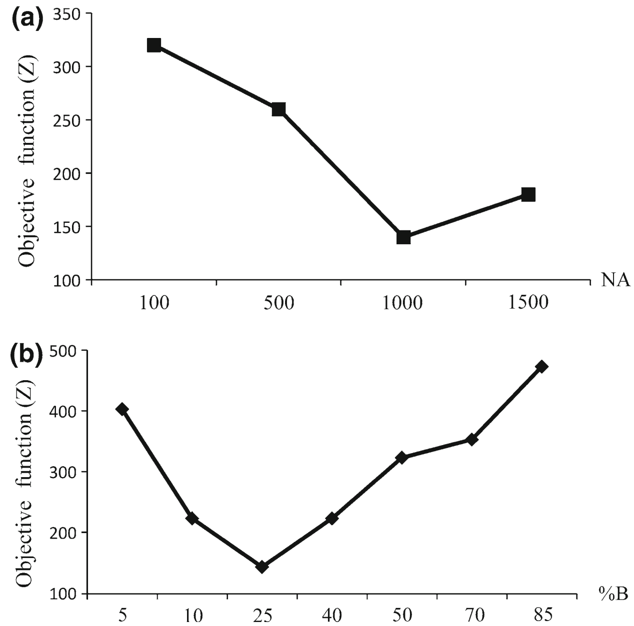


Fig. 6 Proposed AIS with the different values of the NA and the %B

where '$p = 1, 2, \ldots, M$' is the particle number; t is the iteration number; $X_p^L = \{x_{p1}^L, x_{p2}^L, \ldots, x_{pN}^L\}$ and $X^G = \{x_1^G, x_2^G; \ldots; x_N^G\}$ are the local best of the pth particle and the global best, respectively; c_1 and c_2 are the positive constants known as the learning factors; r_1 and r_2 are two random numbers uniformly distributed in the interval [0, 1] [48]; $w(t)$ is the inertia weight used to control the impact of the previous velocities on the current velocity [53].

The pseudo-code of the proposed PSO for the SDST-FJSP is outlined in Fig. 4. The PSO consists of five main processes (see Fig. 5): (1) problem encoding, (2) population initialization, (3) fitness evaluation, (4) updating velocity and position, and (5) stopping criterion. The four procedures of the PSO are similar to the AIS procedures (see Sects. 3.1, 3.2, 3.3, and 3.6). The extra process is described in the following section.

4.1 Updating Velocity and Position

In each iteration, every particle updates its current velocity and position in the search space using the values of the local and global best [33]. In the proposed PSO, this

Table 2 Analysis of variance (ANOVA) on the experimental results

Source	Sum of squares	Degree of freedom	Mean square	F	P value
Number of the antibodies (NA)	36.729	4–1 = 3	12.243	25.69	0.000
Antibody elimination percentage (%B)	51.603	7–1 = 6	8.6005	18.05	0.000
Error	662.498	1,390	0.4766		
Total	750.83	4×7×50 – 1 = 1,399			

Table 3 The characteristic of classes

Class number	Number of jobs (n)	Number of operations for each job (n_i)	Number of of machines (m)	Number of available machines for each operation	Processing times	Sequence-dependent setup times	Dummy jobs
1	10	5	5	$[U(1,6)]$	$U(20,100)$	$U(20,60)$	$U(20,40)$
2	15	5	8	$[U(1,9)]$	$U(20,100)$	$U(20,60)$	$U(20,40)$
3	10	10	5	$[U(1,6)]$	$U(20,100)$	$U(20,60)$	$U(20,40)$
4	15	10	10	$[U(1,11)]$	$U(20,100)$	$U(20,60)$	$U(20,40)$
5	20	10	8	$[U(1,9)]$	$U(20,100)$	$U(20,60)$	$U(20,40)$
6	20	15	10	$[U(1,11)]$	$U(20,100)$	$U(20,60)$	$U(20,40)$

U (a, b): Uniform distribution between (a, b), [x]: The greatest integer which is less than the real number x

Table 4 Average relative percentage deviation (RPD) of the algorithms for $\alpha = 0.25$

Class number	Example number	GA-2008	PVNS-2009	VNS-2011	Proposed PSO	Proposed AIS
1	1	23.58	23.98	10.43	15.46	7.35
	2	25.89	25.85	9.44	13.4	6.63
	3	10.48	18.44	4.78	6.17	2.54
	4	30.84	21.32	3.44	5.29	1.03
	5	34.38	23.5	11.84	16.12	8.26
2	6	23.05	20.46	7.44	6.35	3.13
	7	31.84	23.95	10.34	12.17	5.21
	8	18.57	20.85	5.3	6.64	1.03
	9	19.43	17.84	5.95	7.93	2.47
	10	22.75	20.53	4.16	5.98	1.54
3	11	18.44	32.87	10.75	8.43	5.45
	12	36.95	25.85	11.44	7.35	6.29
	13	18.23	19.74	8.49	4.73	3.02
	14	24.78	24.59	6.93	3.05	2.34
	15	22.76	26.83	8.23	6.15	3.38
4	16	15.84	14.4	5.85	5.27	0.64
	17	14.3	18.56	4.98	2.06	1.55
	18	11.4	14.38	3.49	2.64	1.1
	19	15.75	22.05	10.87	6.53	4.74
	20	17.85	20.44	6.92	4.12	2.87
5	21	26.96	29.84	12.74	8.34	5.55
	22	25.95	31.84	10.72	7.34	2.07
	23	22.96	36.84	11.3	7.82	4.15
	24	26.87	31.75	12.85	6.19	3.35
	25	21.87	28.54	9.43	8.32	4.79
6	26	21.67	22.54	13.65	17.36	9.33
	27	19.4	23.59	12.86	14.44	7.43
	28	20.82	22.98	11.6	15.63	6.25
	29	18.41	24.62	13.21	14.87	9.42
	30	21.97	20.56	10.93	12.65	7.9

Table 5 Average relative percentage deviation (RPD) of the algorithms for $\alpha = 0.5$

Class number	Example number	GA-2008	PVNS-2009	VNS-2011	Proposed PSO	Proposed AIS
1	1	13.85	11.74	4.75	6.56	2.85
	2	15.96	13.08	3.09	4.34	2.14
	3	17.5	12.46	4.95	5.29	1.44
	4	16.94	11.74	4.19	6.22	2.76
	5	18.43	12.84	5.86	7.5	3.85
2	6	17.02	14.89	5.38	7.24	4.19
	7	13.3	12.94	5.22	6.34	3.87
	8	14.74	13.38	7.84	9.85	3.56
	9	17.54	11.64	5.84	7.14	2.13
	10	13.29	12.28	7.85	7.98	4.76
3	11	13.89	8.83	5.63	3.46	1.26
	12	16.84	12.43	6.28	4.88	3.94
	13	12.27	10.46	5.73	3.67	1.85
	14	15.13	17.42	7.37	4.18	3.39
	15	14.58	11.53	8.14	6.95	2.84
4	16	11.73	12.53	5.93	3.57	2.33
	17	11.3	8.63	5.83	4.84	3.74
	18	9.73	13.64	6.31	3.85	3.46
	19	10.63	14.26	5.63	4.53	1.84
	20	9.31	11.46	7.54	4.14	1.65
5	21	17.37	10.83	7.83	6.94	3.94
	22	15.83	9.62	6.32	4.82	3.82
	23	16.93	9.63	7.63	5.15	1.74
	24	21.94	12.94	10.38	8.29	2.93
	25	17.3	11.43	8.63	6.66	4.94
6	26	16.93	14.87	10.83	12.94	8.73
	27	24.03	17.83	9.75	15.83	3.8
	28	19.74	13.24	7.71	9.81	3.35
	29	18.54	14.76	9.7	12.44	5.83
	30	22.38	15.54	9.83	13.84	4.22

updating is done with the aid of swap operator [54] and insert operator [55]. The difference between these operators is that the insert operator removes a node from its original location and inserts it into another location, whereas the swap operator exchanges two nodes in the different locations.

5 Design of Experiment

The aim of this experiment was to investigate the appropriate settings of the proposed AIS parameters, including the number of antibodies (NA) and the antibody elimination percentage (%B), for the SDST-FJSP with 10 jobs, 5 operations for each job, and 5 machines. The full factorial DOE [56] and range of the values are shown in Table 1. The computational runs are replicated 50 times with random numbers. The results of the analysis of variance (ANOVA) are presented in Table 2. It can be discovered that both the NA and %B are statistically significant under the confidence level of 0.95. The AIS has been tested on the different values of these two parameters in Fig. 6. It can be seen that the best settings of the NA and %B are 1,000 and 25, respectively.

6 Comparison of the Results

Six classes of the examples have been used to demonstrate effectiveness and competitiveness of the proposed AIS and PSO algorithms for solving the SDST-FJSP, and to compare with the other existing algorithms, including (1) VNS-2011: the variable neighborhood search approach presented by Bagheri and Zandieh [1]; (2) PVNS-2009: the parallel variable neighborhood search presented by Yazdani et al. [11], and adjusted by Bagheri and Zandieh [1]; and (3) GA-2008: the genetic algorithm presented by Pezzella et al. [12], and adjusted by Bagheri and Zandieh [1]. The characteristic of each class with different numbers of the jobs, the operations and the machines is shown in each row of Table 3. The first four rows of this table are studied by Bagheri and Zandieh [1], while the fifth and sixth rows of this table have been proposed in the present study. The five examples have been generated for each class, in addition to the 30 examples.

The proposed AIS and PSO algorithms have been coded in MATLAB programming language. All the examples have been run on a 2.4-GHz Pentium IV PC with 4 GB of RAM. The settings of the AIS parameters have been stated in the previous section. Similar to the population size in the VNS-2011 algorithm, the number of particles (M) also in the PSO is set to be 1,000. The inertia weight ($w(t)$) and the learning factors (c_1 and c_2) are set to be 1 and 2, respectively. Like the

Table 6 Average relative percentage deviation (RPD) of the algorithms for $\alpha = 0.75$

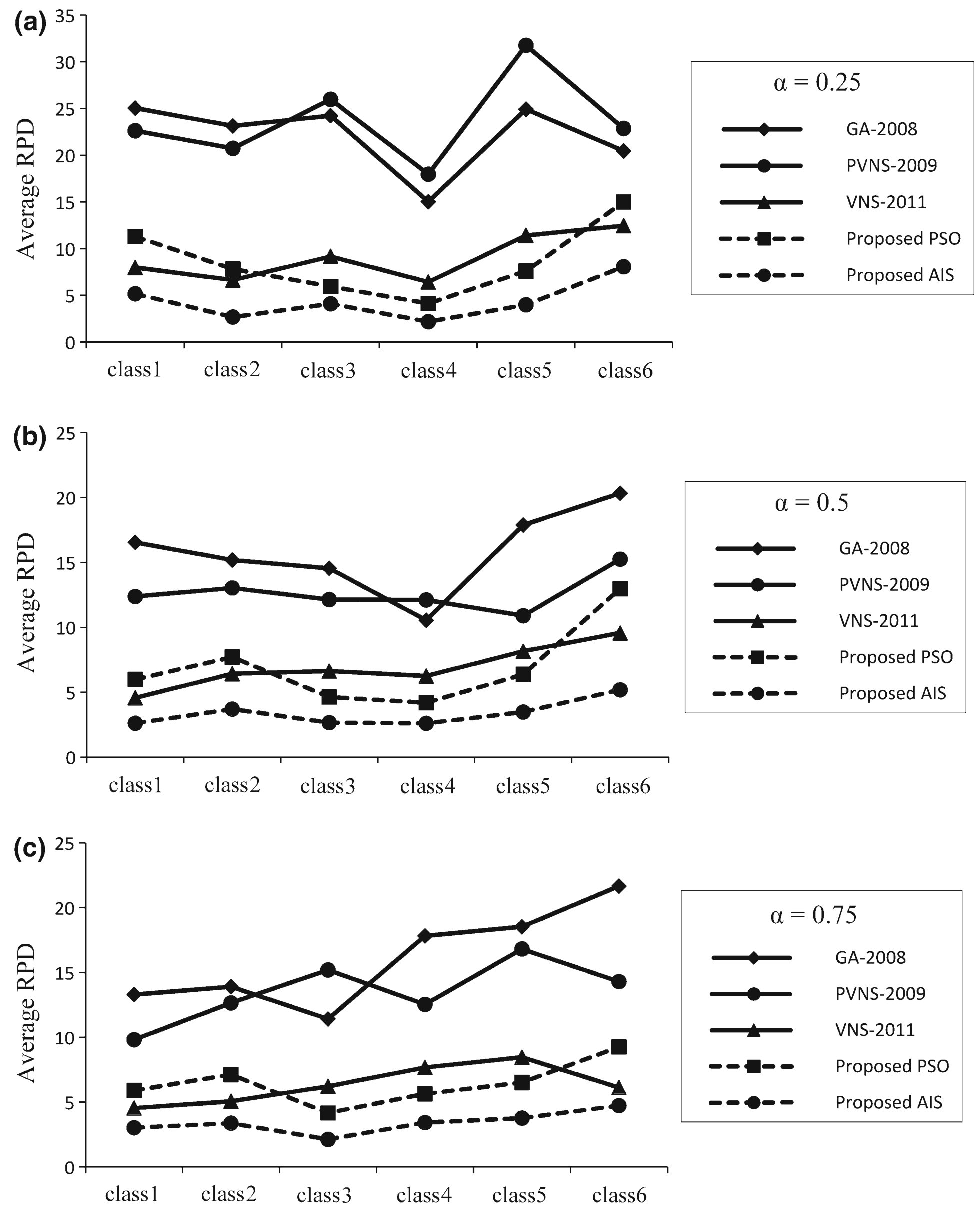
Class number	Example number	GA-2008	PVNS-2009	VNS-2011	Proposed PSO	Proposed AIS
1	1	11.64	8.83	2.43	4.83	3.24
	2	12.74	10.53	3.87	5.91	2.65
	3	14.52	11.86	6.73	7.31	4.96
	4	11.84	8.23	5.84	6.26	2.54
	5	15.75	9.63	3.84	5.19	1.74
2	6	11.73	9.94	4.83	6.98	2.45
	7	12.84	11.52	6.73	7.3	4.75
	8	15.13	14.64	5.73	10.3	3.65
	9	15.49	14.73	3.21	5.93	3.53
	10	14.3	12.43	4.82	5.09	2.5
3	11	9.73	15.3	6.72	2.65	1.86
	12	11.37	17.73	7.12	4.85	3.13
	13	8.3	9.18	4.37	3.17	0.87
	14	16.93	15.93	6.93	5.65	2.01
	15	10.74	17.84	5.93	4.52	2.75
4	16	17.63	11.84	6.94	4.91	2.35
	17	21.39	17.94	11.93	7.84	5.43
	18	17.26	9.74	6.1	5.43	3.19
	19	16.43	11.3	7.38	3.14	2.68
	20	16.38	11.84	6.01	6.88	3.45
5	21	19.16	16.84	7.85	6.83	3.41
	22	20.53	17.73	8.4	5.31	4.83
	23	17.74	15.83	8.83	6.93	3.95
	24	16.37	16.3	9.3	7.28	2.48
	25	18.84	17.36	7.94	6.23	4.16
6	26	19.36	14.83	6.33	8.83	2.84
	27	21.73	12.93	4.9	9.37	3.92
	28	24.85	17.83	7.94	10.84	6.93
	29	18.51	9.93	5.26	7.93	5.11
	30	23.85	15.94	6.17	9.3	4.83

work presented by Bagheri and Zandieh [1], in the current research, a CPU time limit (i.e., $n \times n_i \times m \times 0.1$ s) has been considered as the stopping criterion for testing all examples with all algorithms. And also, ten replications have been performed for each example to achieve more reliable results. In order to compare the performance of the algorithms in the three values 0.25, 0.5 and 0.75 of α, the relative percentage deviation (RPD) in each example has been formulated as follows [57]:

$$\mathrm{RPD} = \frac{Z_{\mathrm{alg}} - Z_{\mathrm{best}}}{Z_{\mathrm{best}}} \times 100 \tag{6}$$

where Z_{alg} is the objective function (Z) obtained for each replication of the given algorithm and Z_{best} is the objective

Fig. 7 Average RPD of the algorithms in the three values 0.25, 0.5 and 0.75 of α



function (Z) of the best solution obtained among the ten replications of all algorithms.

Table 4 shows the average RPD of the GA-2008, the PVNS-2009, the VNS-2011, the PSO, and the AIS algorithms for the ten replications of each example for $\alpha = 0.25$. The results for $\alpha = 0.5$ and 0.75 are shown in Tables 5 and 6, respectively. As can be seen, the proposed AIS performs better than the other algorithms for all values of α. Only in the problems #1 and #9 for $\alpha = 0.75$ (see Table 6), the VNS-2011 algorithm has acquired the better results in comparison with the AIS.

To study the effect of the problem size on the performance of algorithms, the average RPD in the six classes of the examples for $\alpha = 0.25$, 0.5 and 0.75 is plotted in Fig. 7. The results obtained in all classes and all values of α show that the proposed AIS works better than the other algorithms, while the proposed PSO performs better than the GA-2008 and the PVNS-2009 algorithms. For the classes #3, #4, and #5, the PSO does better than the VNS-2011.

The convergence behavior of all algorithms for the classes #5 and #6 (large-sized problems) is shown in Figs. 8 and 9, respectively. From these figures, it can be concluded that the initial solution of the proposed AIS is better than the other algorithms. Also, the proposed AIS hastens the convergence and improves the final solution.

Figure 10 presents the variance of the RPD for all algorithms and all examples in $\alpha = 0.25, 0.5,$ and 0.75. The results show that the proposed AIS is obviously better

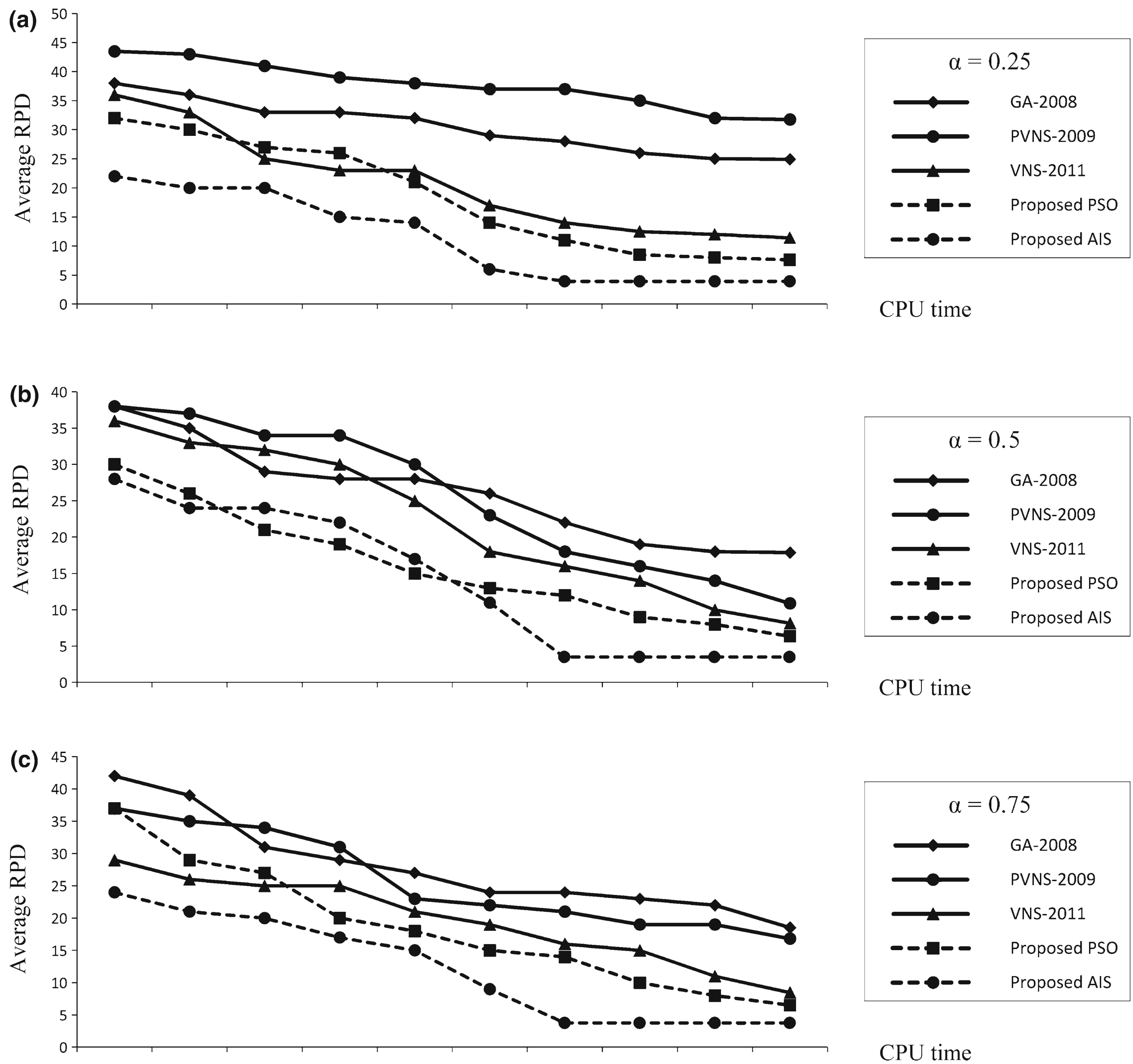


Fig. 8 Convergence curve of the algorithms for the class #5 in the three values 0.25, 0.5 and 0.75 of α

than the four other algorithms in terms of the scheduling stability.

7 Conclusion

The present paper describes the successful development of the artificial immune system (AIS) and the particle swarm optimization (PSO) for solving the SDST-FJSP by minimizing the objective function of makespan and mean tardiness. In order to verify the efficacy of the proposed algorithms with the parameters obtained in the design of experiment, 30 examples with different numbers of jobs, operations and machines have been used to compare with the genetic algorithm (GA) of Pezzella et al. [12], the parallel variable neighborhood search (PVNS) of Yazdani et al. [11], and the variable neighborhood search (VNS) of Bagheri and Zandieh [1]. The results show that while the proposed PSO works better than both GA and PVNS approaches, the proposed AIS performs the best.

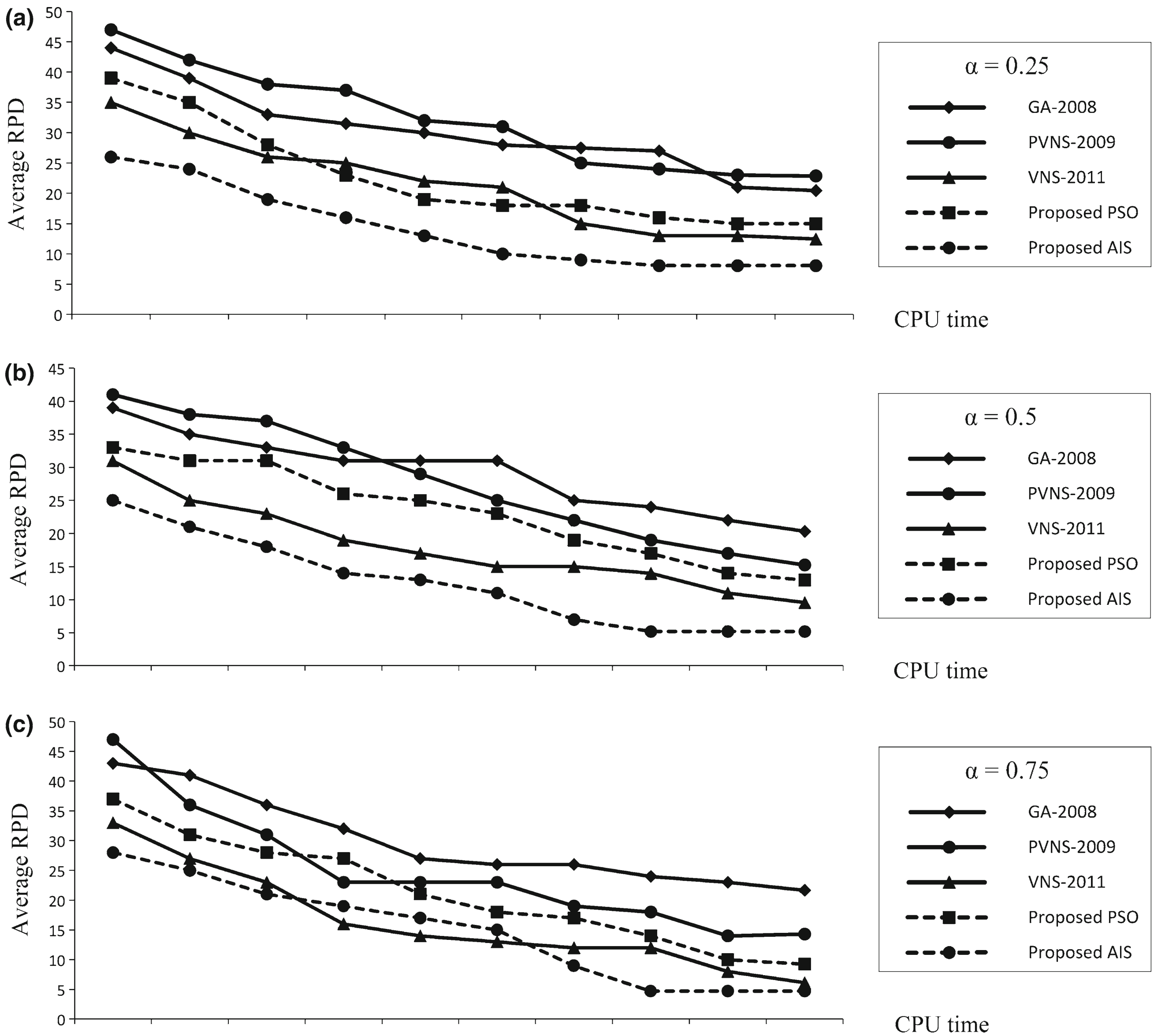
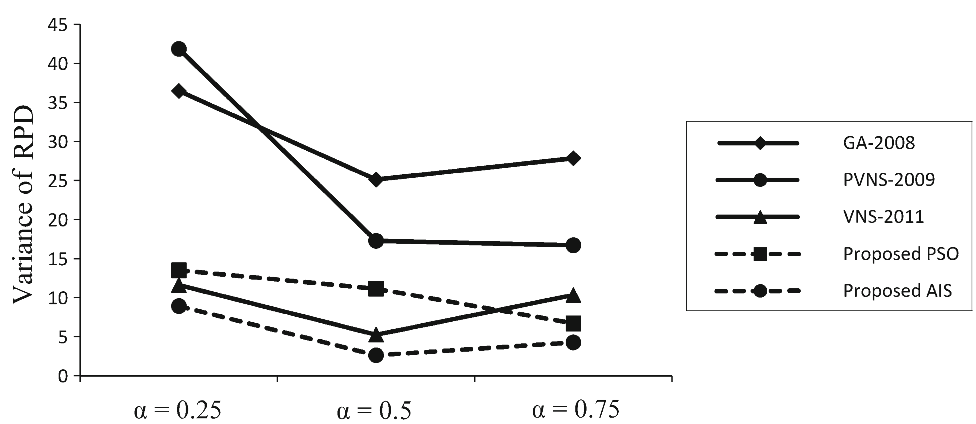


Fig. 9 Convergence curve of the algorithms for the class #6 in the three values 0.25, 0.5 and 0.75 of α

Fig. 10 Variance of the RPD of the algorithms



Acknowledgments The author thanks the Managing Editor and the anonymous referees for their useful comments which have led to considerable improvement in the quality of this manuscript.

References

1. Bagheri, A.; Zandieh, M.: Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—variable neighborhood search approach. J. Manuf. Syst. **30**, 8–15 (2011)
2. Xia, W.; Wu, Z.: An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Comput. Indus. Eng. **48**, 409–425 (2005)
3. Kurz, M.E.; Askin, R.G.: Comparing scheduling rules for flexible flow lines. Int. J. Product. Econ. **85**(2), 371–388 (2003)
4. Luh, P.B.; Gou, L.; Zhang, Y.; Nagahora, T.; Tsuji, M.; Yoneda, K.; et al.: Job shop scheduling with group dependent setups, finite buffers, and long time horizon. Ann. Oper. Res. **76**, 233–259 (1998)
5. Sule, D.R.: Industrial Scheduling. PWS, Boston (1997)
6. Zandieh, M.; Mozaffari, E; Gholami, M.: A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. J. Intell. Manuf. **21**, 731–743 (2010)
7. Manikas, A.; Chang, Y.L.: A scatter search approach to sequence-dependent setup times job shop scheduling. Int. J. Prod. Res. **1**, 1–20 (2008)
8. Li, B.; Wu, S.; Yang, J.; Zhou, Y.; Du, M.: A three-fold approach for job shop problems: a divide-and-integrate strategy with immune algorithm. J. Manufac. Syst. doi:10.1016/j.jmsy.2011.05.005 (2011)
9. Imanipour, N.: Modeling and solving flexible job-shop problem with sequence dependent setup times. In: Proceedings of the International Conference on Service Systems and Service Management, pp. 1205–1210 (2006)
10. Saidi-Mehrabad, M.; Fattahi, P.: Flexible job shop scheduling with tabu search algorithms. Int. J. Adv. Manufac. Technol. **32**, 563–570 (2007)
11. Yazdani, M.; Zandieh, M.; Amiri, M.: Flexible job-shop scheduling with parallel variable neighborhood search algorithm. Expert Syst. Appl. **37**(1), 678–687 (2009)
12. Pezzella, F.; Morganti, G.; Ciaschetti, G.: A genetic algorithm for the flexible job-shop scheduling problem. Comput. Oper. Res. **35**(10), 3202–3212 (2008)
13. Farmer, J.D.; Packard, N.H.; Perelson, A.S.: The immune system, adaption, and machine learning. Physica D **22**, 187–204 (1986)
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, Reading (1989)
15. Ko, C.H.; Wang, S.F.: Precast production scheduling using multi-objective genetic algorithms. Expert Syst. Appl. **38**, 8293–8302 (2011)
16. Khoo, L.P.; Situmdrang, T.D.: Solving the assembly configuration problem for modular products using an immune algorithm approach. Int. J. Prod. Res. **41**, 3419–3434 (2003)
17. Murphy, K.; Travers, P.; Walport, M.: Janeway's Immunobiology. Garland Science, London (2007)
18. Musilek, P.; Lau, A.; Reformat, M.; Wyard-Scott, L.: Immune programming. Inform. Sci. **176**(8), 972–1002 (2006)
19. Tavakkoli-Moghaddam, R.; Rahimi-Vahed, A.; Mirzaei, A.H.: A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. Inform. Sci. **177**, 5072–5090 (2007)
20. De Castro, L.N.; Timmis, J.: Artificial Immune Systems: A New Computational Intelligent Approach. Springer, Berlin (2002)
21. Dasgupta, D.: An overview of artificial immune systems and their applications. In: Artificial Immune Systems and their Applications, pp. 3–18. Springer, Berlin (1998)
22. Gu, F.; Greensmith, J.; Aickelin, U.: Exploration of the dendritic cell algorithm using the duration calculus. In: Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS), pp 54–66 (2009)
23. Dasgupta, D.: Advances in artificial immune systems. IEEE Comput. Intell. Magazine **1**, 40–49 (2006)
24. Ulutas, B.H.; Islier, A.A.: A clonal selection algorithm for dynamic facility layout problems. J. Manufac. Syst. **28**, 123–131 (2009)
25. De Castro, L.N.; Von Zuben, F.J.: The clonal selection algorithm with engineering applications. In: Proceedings of the Workshop on Artificial Immune Systems and Their Applications, Las Vegas, pp. 36–37 (2000)
26. Matzinger, P.: The danger model: a renewed sense of self. Science **296**, 301–305 (2002)
27. Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R.: Self–nonself discrimination in a computer. In: Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, pp. 202–212 (1994)
28. Zandieh, M.; Fatemi Ghomi, S.M.T.; Moattar Husseini, S.M.: An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. Appl. Math. Comput. **180**(1), 111–127 (2006)
29. Ada, G.L.; Nossal, G.J.V.: The clonal selection theory. Sci. Am. **257**, 50–57 (1987)
30. Burnet, F.M.: The Clonal Selection Theory of Acquired Immunity. Vanderbilt University Press, Nashville (1959)
31. Engin, O.; Doyen, A.: Artificial immune systems and applications in industrial problems. Gazi Univ. J. Sci. **17**(1), 71–84 (2004)
32. Kacem, I.; Hammadi, S.; Borne, P.: Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Trans. Syst. Man Cybern. Part C **32**(1), 1–13 (2002)
33. Thapatsuwan, P.; Pongcharoen, P.; Hicks, C.; Chainate, W.: Development of a stochastic optimisation tool for solving the multiple container packing problems. Int. J. Prod. Econ. doi:10.1016/j.ijpe.2011.05.012 (2011)
34. Engin, O.; Doyen, A.: A new approach to solve hybrid flow shop scheduling problems by artificial immune system. Future Gener. Comput. Syst. **20**, 1083–1095 (2004)
35. Clerc, M.; Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
36. Eberhart, R.; Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the IEEE congress on evolutionary computation (CEC), Seoul, Korea, pp. 94–107 (2001)
37. Eberhart, R.; Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
38. Al-Duwaish, H.N.: Identification of Hammerstein models with known nonlinearity structure using particle swarm optimization. Arab. J. Sci. Eng. **36**(7), 1269–1276 (2011)
39. Kuo, R.J.; Lin, L.M.: Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. Decis. Support Syst. **49**, 451–462 (2010)
40. Yang, I.T.: Performing complex project crashing analysis with aid of particle swarm optimization algorithm. Int. J. Proj. Manage. **25**, 637–646 (2007)
41. Bonabeau, E.; Meyer, C.: Swarm intelligence: a whole new way to think about business. Harv Bus. Rev. **79**(5), 107–114 (2001)
42. Li, Y.; Yao, D.; Yao, J.; Chen, W.: A particle swarm optimization algorithm for beam angle selection in intensity-modulated radiotherapy planning. Phys. Med. Biol. **50**(15), 3491–3514 (2005)

43. Eberhart, R.; Shi, Y.: Comparison between genetic algorithms and particle swarm optimization. In: Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming, San Diego, CA, pp. 611–666 (1998)
44. Haixiang, G.; Kejun, Z.; Chang, D.; Lanlan, L.: Intelligent optimization for project scheduling of the first mining face in coal mining. Expert Syst. Appl. **37**, 1294–1301 (2010)
45. Maghsoudi, M.J.; Ibrahim, Z.; Buyamin, S.; Rahmat, M.F.: Data clustering for the DNA computing readout method implemented on lightcycler and based on particle swarm optimization. Arab. J. Sci. Eng. doi:10.1007/s13369-012-0196-3 (2012)
46. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of Congress on Evolutionary Computation, Washington, DC, pp. 1951–1957 (1999)
47. Taleizadeh, A.A.; Akhavan Niaki, S.T.; Haji Seyedjavadi, S.M.: Multi-product multi-chance-constraint stochastic inventory control problem with dynamic demand and partial back-ordering: a harmony search algorithm. J. Manufac. Syst. doi:10.1016/j.jmsy.2011.05.006 (2011)
48. Zhang, H.; Li, H.; Tam, C.M.: Particle swarm optimization for resource-constrained project scheduling. Int. J. Proj. Manage. **24**, 83–92 (2006)
49. Fan, H.: A modification to particle swarm optimization algorithm. Eng. Comput. **19**(8), 970–989 (2002)
50. Kennedy, J.; Eberhart, R.; Shi, Y.: Swarm Intelligence. Morgan Kaufman Publishers, San Francisco (2001)
51. Shi, Y.; Eberhart, R.: Parameter selection in particle swarm optimization. In: Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming, New York, pp. 591–600 (1998)
52. Kennedy, J.; Eberhart, R.: Particle swarm optimisation. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
53. Shi, Y.; Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
54. Wang, K.P.; Huang, L.; Zhou, C.G.; Pang, W.: Particle swarm optimization for traveling salesman problem. In: Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xian, pp. 1583–1585 (2003)
55. Cuiru, W.; Jiangwei, Z.; Jing, Y.; Chaoju, H.; Jun, L.: A modified particle swarm optimization algorithm and its application for solving traveling salesman problem. In: Proceedings of the International Conference on Neural Networks and Brain, ICNN&B '05, pp. 689–694 (2005)
56. Kök, M.: Modeling and assessment of some factors that influence surface roughness for the machining of particle reinforced metal matrix composites. Arab. J. Sci Eng **36**(7), 1347–1365 (2011)
57. Sadrzadeh, A.: A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. Comput. Indus Eng. **62**(4), 1055–1064 (2012)