Received: November 3, 2020. Revised: February 8, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Chaotic Student Psychology based Optimization Algorithm for Bi-Objective Permutation Flowshop Scheduling Problem

Ayomi Sasmito[1] **Asri Bekti Pratiwi**[2]*

[1]*Postgraduate Programs of Mathematics, Gadjah Mada University, Yogyakarta, Indonesia*
[2]*Department of Mathematics, Faculty of Science and Technology, Universitas Airlangga, Surabaya, Indonesia*
* Corresponding author's Email: asri.bekti@fst.unair.ac.id

Abstract: This paper proposes an efficient optimization algorithm called Chaotic Student Psychology Based Optimization (CSPBO) to solve bi-objective permutation flowshop scheduling problem (BPFSP). The SPBO algorithm does not require any tunning parameters which makes it simpler in computational experiments. The original SPBO classifies students related to their efforts in improving their performances. In this paper, we use chaotic maps to enhance student efforts in each category and present different strategic approaches. Logistic, iterative, sine, tent, and singer maps are integrated using proposed strategies to find the best map and also its strategy, named CSPBO. To prove its performance, the CSPBO is compared with several well-known metaheuristic algorithms. The comparison results are evaluated in regard to mean, standard deviation, best terms and ARPD. Since BPFSP is an NP-hard problem, computational experiments on BPFSP instances are carried out according to type size of the dataset. Three types of dataset, small, medium and large are tested. Based on the experiment result, the SPBO integrated with logistic map has significantly given better performance compared to other chaotic maps. Furthermore, CSPBO shows compatible performance compared with other metaheuristic algorithms in solving BPFSP.

Keywords: SPBO, BPFSP, Chaotic maps, Optimization.

1. Introduction

Optimization algorithms have evolved based on nature observation ideas in determining the best solution from feasible alternative solutions with regard to given problems. Either heuristic and metaheuristic approaches can be an alternative technique in optimization algorithms. Evolutionary algorithms in metaheuristic approach generally require parameters, besides population size, which are highly sensitive taking effect to the algorithm performances, such as genetic algorithm, differential evolution, and harmony search [1, 2]. Genetic algorithm (GA) has two basic parameters, crossover and mutation probability, where some research has recommended and analyzed the parameter values to improve its performance [3]. Giving different values of each parameter will also provide different results which sometimes can lead to bad results. Another algorithm, harmony search, also has constant parameters playing a critical role in obtaining the near-optimal solution where higher chance of finding the global optima [4, 5]. Apart from the evolutionary algorithm, the metaheuristic approach based on swarm behavior also takes tuning parameters to control the efficiency of the given problems.

Artificial bee colony algorithm proposed by Karaboga [6] has *limit* as parameter changing the performance of algorithms. As the core parameter of the algorithm in determining scout bee occurrence, a different value of control parameter *limit* gives different result in behavior in the population [7]. Since algorithm performances depend on parameters, determine the appropriate values for those required parameters in solving optimization problem always become a challenging topic. Inappropriate values of algorithm parameters which are adjusted in the initial process computation can bring unsatisfactory result in optimization process. Finding the significant value of parameters with aims to improve performances certainly takes time in computational experiments. Therefore, this paper considers a new population-based algorithm proposed by Das et. al. which not requires any control parameters in the algorithm, called student psychology-based optimization (SPBO) algorithm [8].

SPBO algorithm is adapted from student psychology who always gives efforts to obtain the highest grade in subject evaluation. Although students always try to get high marks/grades in their evaluation but only capability, efficiency and interest to the subject offered which can affect the performance of students. On condition that a student interests to a subject offered, then the grades can be easily improved. Accordingly, each student aims to improve own performance by giving the best effort to obtain good grades so that become the best student in the class [8]. There are four student classifications in SPBO, i.e. best student, good students, average students, and students who try to improve randomly.

Permutation flowshop scheduling problem (PFSP) is well-known as NP-hard problem since it commonly involves more than two machines in real systems. The PFSP may be stated as a set of jobs which is processed in a set of machines in the same order. The time of processing of job i $(i = 1,2, \dots n)$ on machine j $(j = 1,2, \dots m)$, p_{ij}, is known. The aim of PFSP is finding a jobs sequence minimizing the maximum completion time of overall process, C_{max}, called as makespan. Generally, at most one job can be processed by each machine and at most one machine can process each job, as well as preemption is not allowed. Therefore, a jobs sequence for each machine is particularly the same [9]. In some research, PFSP is considered only one scheduling objective, i.e. makespan. However, in real system situation, decision-makers need think about multiple objectives simultaneously. The tardiness term is important enough in systems of production, because the penalty costs will incur if the jobs cannot be completed before the due dates. Therefore, the due date of job i is usually given [10]. The PFSP with two objectives is called as bi-objective PFSP. There are various kinds of algorithm used in resolve PFSP with two objectives, such as the branch and bound algorithm to optimize total flow time and total tardiness [11] and ant colony optimization to optimize makespan and total flowtime [12]. Some metaheuristics have been proposed to solve this problem.

An investigation on pareto-ant colony optimization (pareto-ACO) in solving bi-PFSP shows that pareto is a competitive method for multi-objective criterion [13]. Comparing with GA, ACO, simulated annealing and particle swarm optimization (PSO), the improved cuckoo search algorithm (CSA) has better performance to minimize hybrid FSP makespan [14]. Research in job scheduling problem using teaching learning-based optimization (TLBO) was conducted to minimize expenses. The experiment result shows that TLBO has considerable potential to solve PFSP [15]. Moreover, according to [8] SPBO gives better performance in given benchmark function tests compared with other algorithms including TLBO [16] where tuning parameters also are not needed.

Similar to other metaheuristics, there are still unstable performances of SPBO algorithm in solution qualities, early convergence, as well as there is a possibility of local optima trapped when solving global optimization problems. Hence, this paper tries to settle the SPBO issues, convergence and solution qualities, by modifying in order to enhance the performances. On the other hand, a chaotic optimization algorithm has gained popularity in recent years among optimization areas [17]. The chaotic concepts and functions can enhance algorithm performance in handling early convergences problem, diversity and providing more efficient and effective results [18]. A modification on PSO using improved chaotic successfully gain a good balance between global and local search capabilities where chaotic motion can control the particle position representing solution [19]. Chaotic maps especially logistic map can raise the quality of solution which escapes from local optimum [20].

In this work, five different variants of chaotic SPBO algorithm are proposed based on chaotic maps [19]: logistic, iterative, sine, tent, and singer maps. Chaotic maps are used to enhance the effort of students in each category. Besides the enhancement, different strategic approaches are proposed to find the best performance correspond to its map and strategy, namely CSPBO. Moreover, CSPBO is compared to the performance of four optimization algorithms, such as the original SPBO, TLBO [16], CSA [21], squirrel search algorithm [22] and pareto-SPBO [23, 24]. The performance is analyzed based on optimum result obtained as well as its convergence mobility. The rest of this paper is organized as follows. Section 2 explains the bi-objective PFSP model. The next section, Section 3, describes the proposed Chaotic SPBO algorithm. Section 4 discusses the experiment result. Lastly, Conclusion is presented in Section 5.

2. Bi-objective PFSP

As explained in the previous section, real system problem in scheduling sometimes involves more than

one objective. A multi-objective optimization model can be described as follow [25]:
Optimize which is can be minimized/maximized

$$F(X) = \{f_1(X), f_2(X), \ldots, f_p(X)\} \quad (1)$$

while the constraints are given as

$$H(X) = 0 \quad (2)$$

$$G(X) \geq 0 \quad (3)$$

where $F(X)$ is the set of functions to be optimized and a set of decision variables is denoted as vector X. Three situations can be found in in multi-objective optimizations, listed as follow:

1. To minimize all p objectives
2. To maximize all p objectives
3. To minimize r objectives and maximize $p - r$ objective

A weighted sum creates a single-objective model from multi-objectives model by assigning a weight to each objective function. Therefore, the single-objective model can be provided below:
Optimize which is can be minimized/maximized

$$F(X) = \sum_{k=1}^{p} \omega_k f_k(X) \quad (4)$$

while the constraints are given as

$$H(X) = 0 \quad (5)$$

$$G(X) \geq 0 \quad (6)$$

$$0 \leq \omega_k \leq 1, \qquad for\ k = 1,2, \ldots, p \quad (7)$$

$$\sum_{k=1}^{p} \omega_k = 1 \quad (8)$$

where $\omega_k, k = 1,2, \ldots, p$ are the weights assigned to each objective function. The simplest model of multi-objectives optimization is bi-objectives optimization model where permutation flowshop scheduling problem with two objectives in minimizing makespan and tardiness are discussed in this paper.

A permutation flowshop scheduling problem with two objectives called bi-objectives permutation flowshop scheduling problem (BPFSP) belongs to NP-hard problems. BPFSP finds a sequence of n jobs processed by m machines where each job can only be assigned to any one of machines at given time. Each job is processed sequentially on machines with the same order of machines and the sequence of job processing is same for all machines. A positive processing duration of job $i(i = 1,2, .., n)$ on machine $j(j = 1,2, .., m)$ is deterministic known denoted as p_{ij}. Beside a due date, d_i, by which job i should ideally finished, each job i has a positive weight, w_i, taking effect to tardiness. In this paper, the BPFSP is to find the minimum of the total time needed to completely process all the n jobs, called as makespan, and total weighted tardiness [11].

Let $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$ denotes a set of job processing sequences in each machine, while $C_{i,j}$ denotes the completion time of job i on machine j, and T_i denotes the tardiness of job i. Then, the makespan denoted as $C_{\max}$, and the total weighted tardiness denoted as Tw can be determined as follow [26]:

$$C_{\pi(1),1} = p_{\pi(1),1} \quad (9)$$

$$C_{\pi(1),j} = C_{\pi(1),j-1} + p_{\pi(1),j}\ , j = 2,3, \ldots m \quad (10)$$

$$C_{\pi(i),1} = C_{\pi(i-1),1} + p_{\pi(i),1}\ , i = 2,3, \ldots n \quad (11)$$

$$C_{\pi(i),j} = \max\{C_{\pi(i-1),j}, C_{\pi(i),j-1}\} + p_{\pi(i)),j}, \\ i = 2,3, \ldots n; j = 2,3, \ldots, m \quad (12)$$

$$C_{\max} = \max\{C_{\pi(n),m}\} \quad (13)$$

$$T_{\pi(i)} = \max\{C_{\pi(i),m} - d_{\pi(i)}, 0\}, i = 1,2, \ldots n (14)$$

$$Tw = \sum_{i=1}^{n} T_{\pi(i)} w_i \quad (15)$$

Therefore, based on the weight sum method, the single objective optimization form of BFSP can be restated as follow:

$$Minimize\ F = \omega_1 C_{\max} + \omega_2 Tw \quad (16)$$

where ω_1, ω_2 are the weight coefficients and $\omega_1 + \omega_2 = 1$. Since there are two objectives, the priority between both objectives takes part in determining the weight coefficients. In this paper, the priority between makespan and total weighted tardiness are assumed to be the same. Therefore, the weight coefficients $\omega_1 = \omega_2 = 0.5$.

3. Chaotic SPBO algorithm

A metaheuristic algorithm proposed by Das et.al based on student psychology in undertaking to reach the highest grade in the subject evaluation. Each student always tries to get good grade in every subject offered but the performance depends on the capability, efficiency and interest to the subject offered (see [8]). If a student has high interest to subject offered then there is a higher possibility that the student will obtain a good grade in the subject evaluation. Moreover, students will give their best effort to improve their performances as well as trying hard to be the best student in the class. Based on their evaluation results, the students are classified into four categories: best student, good students, average students, and students who try to improve their performance randomly. Besides the best student, student categorization is conducted in randomly process according to student psychology whether the student is categorized into good student, average student or student with random improvement. The overall grade obtained by a student is appeared as the fitness, $f(X_i)$.

Chaos is deterministic, random-like behavior found in nonlinear dynamical system that is sensitive dependence on the initial condition [27]. The nature of chaos systems is apparently random and unpredictable. A discrete-time dynamical system for chaotic state can be formulated as follows:

$$x_{k+1} = f(x_k), 0 < x_k < 1, k = 0,1,2,\ldots \quad (17)$$

where $x_k \{k = 0,1,2,\ldots\}$ is the chaotic state or chaotic sequence, which can be utilized as spread-spectrum sequence as random number sequence. Chaotic sequences are easy to generate and fast in computation which require less memory because they only need chaotic map and an initial condition.

It has been studied that the modification process in each category of students in SPBO algorithm can affect the algorithm convergence. The SPBO algorithm uses random numbers to improve the solution qualities. In order to control the random numbers effect on the algorithm performance, the SPBO algorithm is hybridized with chaotic maps. Unordered random numbers are generated using chaotic maps through random function as the random number generator which have particular pattern. Chaotic maps integration into optimization algorithm is needed to enhance its performance [28].

A research conducted by [19] discuss the comparison performance between chaotic maps. Based on the simulation result in twelve types of problem, the logistic map gives best performance in increasing solution quality comparing with other twelve chaotic maps such as gauss map, etc. Sine and iterative maps also perform better than sinusoidal, liebovitch, etc. Although tent and singer maps less superior than sine and iterative maps, their achievement is better than gauss, intermittency, and piecewise maps. Therefore, this paper proposes five chaotic maps: logistic, tent, sine, singer and iterative to enhance the SPBO quality solutions.

Table 1. Chaotic maps characteristics used in experiment

Chaotic Map	**Equation**	**Parameter**
Logistic [14]	$x_{k+1} = ax_k(1 - x_k)$	$a = 4, x_0 = 0.3$
Iterative [14]	$x_{k+1} = \sin(\frac{a\pi}{x_k})$	$a = 0.7,$ $x_0 = 0.3$
Sine [14]	$x_{k+1} = \frac{a}{4}\sin(\pi x_k)$	$a = 4, x_0 = 0.3$
Tent [14]	$x_{k+1} = \begin{cases} \frac{x_k}{0.7}, & x_k < 0.7 \\ \frac{10}{3}(1 - x_k), & x_k \geq 0.7 \end{cases}$	$x_0 = 0.3$
Singer [14]	$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.3x_k^4)$	$\mu = 1.07,$ $x_0 = 0.3$

The proposed chaotic maps are hybridized into SPBO algorithm through five different techniques. The initial value of the chaotic maps is determined randomly between 0 and 1. Since in chaotic systems early period generations are not sufficient important then the simulation choose initial value $x_0 = 0.3$ for all types of chaotic maps in order to compare their performance. The explanation for each chaotic map is presented in Table 1 while Fig. 1 presents the chaotic value distribution for five maps using $x_0 = 0.3$ and number of iterations is 500 in range (0,1).

As seen in Fig. 1, each chaos variable is randomly dispensed between 0 and 1 in 500 iterations. The five different techniques of modification are explained as follows.

SPBO-I

Parameter in for the best student, $rand$, is updated using chaotic maps during entire execution of experiment. The update performance of best student category can be rewritten as

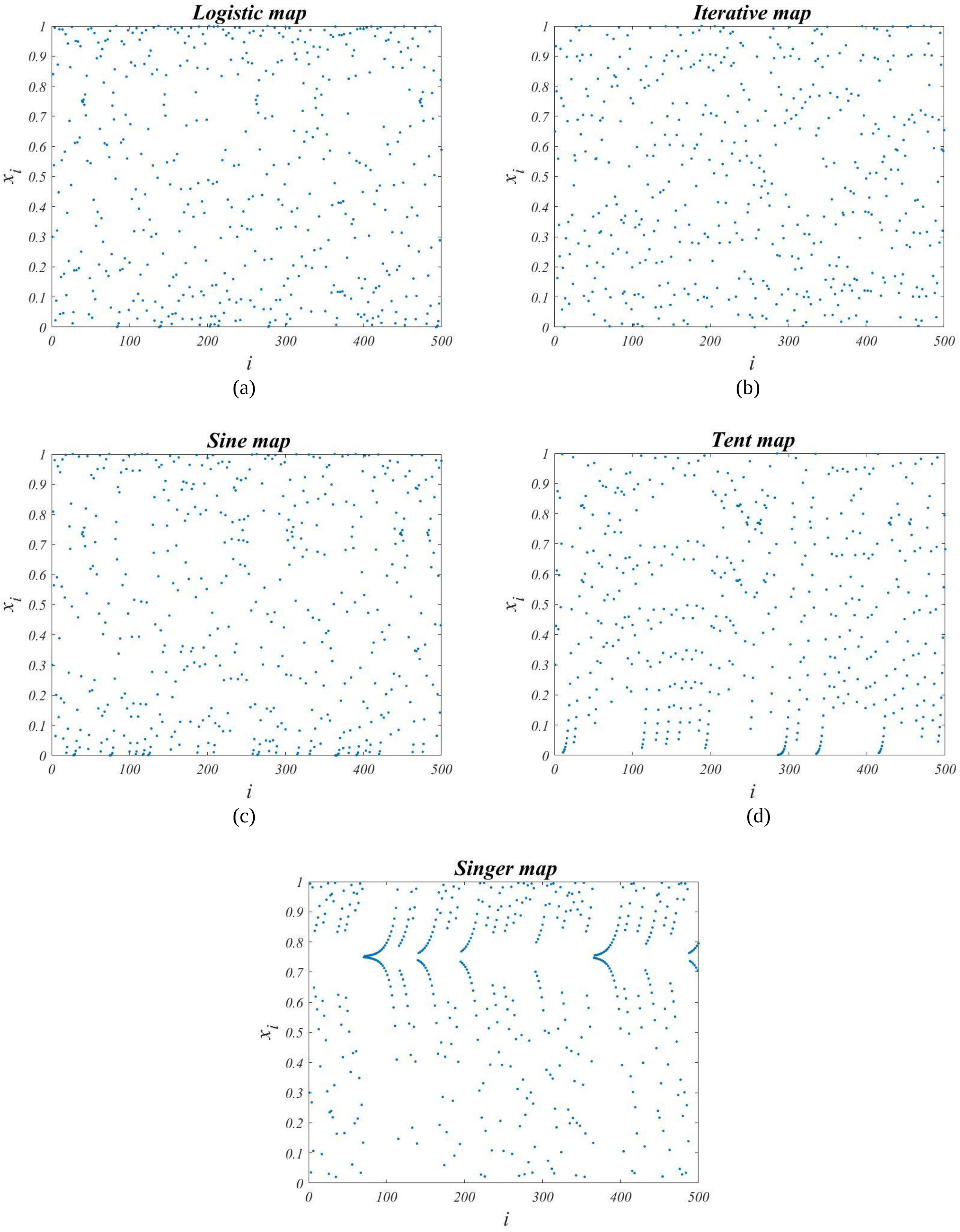
$$X_{best}^{new} = X_{best} + (-1)^k \times chaos(t) \times \left(X_{best} - X_j\right) \quad (18)$$

In SPBO algorithm, $rand$ is a random number between 0 and 1, whereas in SPBO-I, $chaos(t)$ presents a chaotic number between 0 and 1 in iteration t.

SPBO-II

Parameter in good student category, $rand$, is updated using chaotic maps during entire execution of experiment. The update performance of good student category can be rewritten as

$$X_i^{new} = X_{best} + [chaos(t) \times (X_{best} - X_i)] \quad (19a)$$



(a)

(b)

(c)

(d)

(e)

Figure. 1 Random number generated through chaotic maps in 500 iterations: (a) logistic map, (b) iterative map, (c) sine map, (d) tent map, and (e) singer map

$$X_i^{new} = X_i + [chaos(t) \times (X_{best} - X_i)] + [chaos(t) \times (X_i - X_{mean})] \quad (19b)$$

SPBO-III

Parameter in average student category, $rand,$ is updated using chaotic maps during entire execution of experiment. The update performance of average student category can be rewritten as

$$X_i^{new} = X_i + [chaos(t) \times (X_{mean} - X_i)] \qquad (20)$$

In SPBO algorithm $rand$ is a random number between 0 and 1, whereas in SPBO-III, $chaos(t)$ presents a chaotic number between 0 and 1 in iteration t.

SPBO-IV

Parameter in student who try to improve randomly, is updated using chaotic maps during entire execution of experiment. The update performance of student who try to improve randomly can be rewritten as

$$X_i^{new} = X_{min} + [chaos(t) \times (X_{max} - X_{min})] \qquad (21)$$

In SPBO algorithm $rand$ is a random number between 0 and 1, whereas in SPBO-IV, $chaos(t)$ presents a chaotic number between 0 and 1 in iteration t.

SPBO-V

The strategy on SPBO-V is combination of the SPBO I-IV explained as follows:

Algorithm : SPBO-V

```
Step 1 : Initialize the population in class $X_i$ ($i = 1,2, \dots, number\ of\ students$), number of subjects, number of maximum iteration (max_$iter$) and $chaos(0)$ value for each students.
Step 2 : Evaluate students performance in class $f(X_i)$
Step 3 : $t = 1$
Step 4 :
While $t \le$ max_$iter$ do
for $s = 1$ to number of subjects
    Define each student category in population
    for $i = 1$ to number of students
        if student $i$ is the best student then
            Modify performance using (18)
        else if student $i$ is the good student then
            if student $i$ gives effort based on the best
            student only then
                Modify performance using (19a)
            else
                Modify performance using (19b)
            end if
        else if student $i$ is the average student then
            Modify performance using (20)
        else
            Modify performance using (21)
        end if
    end for
    Evaluate students performance in class $f(X_i^{new})$
    If the new performance $f(X_i^{new})$ is better than the older
    $f(X_i)$ then
        Replace the old performance with the new one :
        $X_i = X_i^{new}$
    else
        Keep the old performance
    end if
end for
$t = t + 1$
end while
Step 5 : Display the performance of the best student
```

4. Experiment result

This section discusses about the analysis and results of BPFSP using the proposed techniques. To evaluate the performance of the Chaotic SPBO (CSPBO) algorithm in solving BPFSP, the CSPBO is coded in C++ language and run on a PC with AMD Ryzen 3 3200U @2.4GHz / 12GB RAM under Windows 10. The test instances are generated benchmark set used in Bi-Objective Permutation Flowshop Scheduling Problem (BPFSP) data which is used in this paper were taken from http://soa.iti.es/files/Instances_PFSP_SDST.7z. Data consists number of jobs, number of machines, time processing, due date, and weight. The data taken classified into three types, namely small (20 jobs with 10 machines), medium (50 jobs with 5 machines), and large dataset (100 jobs with 10 machines).

In order to evaluate the proposed CSPBO performance, three types of data were observed. The CSPBO algorithm was combined with five different chaotic maps: logistic, iterative, sine, tent, and singer [14], using five different approaches to obtain the best chaotic approach in SPBO algorithm. In this work, population size for algorithms was set to 50. The maximum number of iterations was taken as 500 and each data set was evaluated with 30 independent run.

Table 2, Table 3 and Table 4 summarize the average of the objective function values using logistic, iterative, sine, tent, and singer maps associate to these approaches using small dataset, medium dataset and large dataset, respectively, to analyze the best CSPBO algorithm used to solve BPFSP in finding the optimum solution. The first row presents the best objective value found so far corresponds to its chaotic

Received: November 3, 2020. Revised: February 8, 2021.

Table 2. Performances in small dataset

Logistic	Iterative	Sine	Tent	Singer
SPBO - I				
1427.5	1501.3	1361.7	1454.5	1443.5
2	5	1	4	3
SPBO - II				
1501.9	1546.6	1534.5	1438.4	1514.1
2	5	4	1	3
SPBO - III				
1446.1	1393.2	1471.9	1391.4	1476.3
3	2	4	1	5
SPBO - IV				
1407.9	1514.8	1421.7	1436.9	1435.5
1	5	2	4	3
SPBO - V				
1398.5	1493.1	1431.8	1440.7	1470.0
1	5	2	3	4

Table 3. Performances in medium dataset ($\times 10^2$)

Logistic	Iterative	Sine	Tent	Singer
SPBO - I				
702.41	710.85	686.98	717.60	726.39
2	3	1	4	5
SPBO - II				
706.95	706.64	725.64	732.47	732.30
2	1	3	5	4
SPBO - III				
703.98	706.82	721.34	726.23	700.68
2	3	4	5	1
SPBO - IV				
717.37	725.96	758.49	743.77	737.75
1	2	5	4	3
SPBO - V				
699.79	700.78	682.13	720.20	724.49
2	3	1	4	5

Table 4. Performances in large dataset ($\times 10^3$)

Logistic	Iterative	Sine	Tent	Singer
SPBO - I				
560.36	561.03	572.73	563.45	571.86
1	2	5	3	4
SPBO - II				
541.43	532.74	536.67	552.38	545.05
3	1	2	5	4
SPBO - III				
566.73	567.92	573.13	567.34	571.88
1	3	5	2	4
SPBO - IV				
578.23	572.55	573.77	574.22	580.60
4	1	2	3	5
SPBO - V				
532.54	550.99	533.77	557.33	544.75
1	4	2	5	3

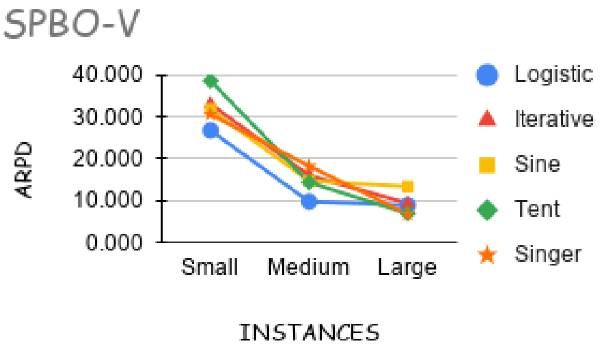


Figure. 2 Chaotic maps performance

map while the second row shows performance rank of each map corresponds to value obtained.

Considering the all strategies proposed as seen in Tables 2, 3, and 4, the SPBO-V provides smallest objective function in solving BPFSP indicating that SPBO-V is the most suitable strategy to enhance the original SPBO performance. By modifying each student category using chaotic search can optimize the exploitation process in the algorithm. Further analysis is needed to find better map to hybridize with the SPBO-V.

The effectiveness of the chaotic map studied is analyzed using Average Relative Percentage Deviation (ARPD), determined in equations below [15]:

$$ARPD = \sum_{i=1}^{R}\left(\frac{(S_i - B)\times 100}{B}\right)/R \qquad (22)$$

where S_i denotes the best result generated by the algorithm in run-i, B denotes best solution reached among all compared algorithms, and R denotes number of runs. The simulation result for each chaotic map using SPBO-V is exposed in Fig. 2.

Fig. 2 illustrates that implementation SPBO-V strategy in solving BPFSP for small (20 jobs with 10 machines), medium (50 jobs with 5 machines), and large dataset (100 jobs with 10 machines) using the logistic map gives better performance using all strategies and dataset types. Logistic map which is adapted from s-curve logistic function uses a nonlinear difference equation to look at discrete time steps [15]. Its behavior has brought the SPBO to find the best random number to gain the near-optimal solution. Apparently, growth parameter set $a = 4$ becomes good combination with its initial value. Since too low growth rate causes smaller result for the next step value. Otherwise, fluctuate value can be obtained using higher growth rate which bring to global optimization toward the SPBO algorithm.

It is observed that among all chaotic maps and strategies, logistic map with SPBO-V strategy is the

Table 5. Performance comparison ($\times 10^3$)

Algorithm	Terms	Small	Medium	Large
SPBO	**Average**	1.52	68.83	556.0
	St.Dev	0.29	4.64	20.86
	Best	1.12	62.61	507.8
CSA	**Average**	1.65	77.23	593.0
	St.Dev	0.18	4.27	9.68
	Best	1.27	65.82	566.3
TLBO	**Average**	1.74	69.52	555.9
	St.Dev	0.37	5.19	47.48
	Best	1.27	61.26	484.0
SSA	**Average**	1.94	78.12	580.6
	St.Dev	0.47	15.48	17.45
	Best	1.24	89.27	537.7
Pareto - SPBO	**Average**	1.93	91.34	608.6
	St.Dev	0.25	3.19	9.10
	Best	1.27	91.34	589.2
CSPBO	**Average**	1.40	69.98	532.5
	St.Dev	0.20	4.06	16.82
	Best	1.10	63.74	506.6

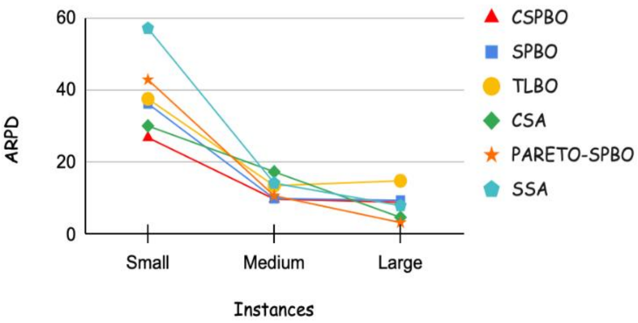


Figure. 3 CSPBO performance in ARPD term

Table 6. Average ARPD

Terms	Average (ARPD)
CSPBO	15.165
SPBO	18.521
TLBO	21.948
CSA	17.362
Pareto-SPBO	18.932
SSA	26.392

most efficient method to be applied in dealing with BPFSP. Therefore, SPBO-V is called as Chaotic-SPBO (CSPBO). Next, the simulation result of proposed strategies is compared with five different meta-heuristic algorithms: SPBO [8], CSA [21], TLBO [16], Squirrel Search Algorithm (SSA) [22] and Pareto-SPBO [23,24]. As explained in the first section, the CSA, TLBO and Pareto have been applied to solve FPSP in various constraints and strongly produce superior performances [13-15]. The parameters setting of mentioned algorithms are taken based on the corresponding literatures [8,16,21-24]. The experiment result is presented in the terms of average values, standard deviation values, and best solution obtained after 30 run and shown in Table 5. According to Table 5, based on the result of BPFSP objective function values using small dataset, medium dataset and large dataset show that the CSPBO algorithm have superior performance compared with other algorithms. The CSPBO algorithm carries out better results in average term with smaller standard deviations. Unfortunately, CSPBO algorithm has failed to obtain best solution comparing with TLBO, which does not have any tunning parameter similar with SPBO, when applying in medium and large datasets. During verified its performance, ARPD calculation is revealed in Fig. 3.

CSPBO algorithm profitably gains best result in ARPD term compared with other algorithms in small dataset but less superior in large dataset, as seen in Fig. 3. The average ARPD for small, medium, and large instances shows in Table 6 describes that the CSPBO algorithm has the lowest average ARPD value. Logistic map has displaced the $rand$ operator in each student category modification. Although chaotic works in the same range real number between 0 and 1, the generated number will be not repeatable and same patterns can exist at every scale. These behaviors improve wider search area in finding candidate solution in search space, therefore applicable to solve BPFSP.

5. Conclusion

This paper presents the Chaotic-SPBO for solving BPFSP. Based on the experiment result, SPBO algorithm using logistic map as enhancement in all student categories modification has profitably given best performance in solving BPFSP. Logistic map is the most suitable map to be hybridized with SPBO algorithm where successfully exploits the search space. The CSPBO also provides great performance compared with other metaheuristic algorithms. Comparing with the original SPBO, the CSPBO algorithm works in any types of dataset well. Random number generated by chaotic map can enhance wider search space in stable system.

Conflicts of Interest

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of this article.

Author Contributions

All authors conceived and designed the study. Ayomi Sasmito conducted the experiments. Asri Bekti Pratiwi analyzed the data and wrote the paper.

References

[1] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm", *Applied Soft Computing*, Vol. 61, pp. 1041-1059, 2017.

[2] B. Si, Z. Tian, X. Jin, X. Zhou, and X. Shi, "Ineffectiveness of optimization algorithms in building energy optimization and possible causes", *Renewable Energy*, Vol. 134, pp. 1295-1306, 2019.

[3] A. R. Ayad, H. A. Awad, and A. A. Yassin, "Parametric analysis for genetic algorithms handling parameters", *Alexandria Engineering Journal*, Vol. 52, Issue 1, pp. 99-111, 2013.

[4] N. F. Mansor, Z. A. Abas, A. F. N. A. Rahman, A. S. Shibghatullah, and S. Sidek, "A New HMCR Parameter of Harmony Search for the Better Exploration", In: J. Kim, Z. Geem. (eds) Harmony Search Algorithm. *Advances in Intelligent System and Computing*, Vol. 382. Springer, Berlin, Heidelberg. 2016.

[5] V. Kumar, J. K. Chhabra, and D. Kumar, "Effect of Harmony Search Parameters' Variation in Clustering", *Procedia Technology*, Vol. 6, pp. 265-274, 2012.

[6] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", *Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department*, 2005.

[7] B. Akay, and D. Karaboga, "Parameter Tuning for the Artificial Bee Colony Algorithm", In: N. T. Nguyen, R. Kowalczyk, S. M. Chen. (eds) Computational Collective Intilligence. *Semantic Web, Social Networks and Multiagent Systems*. ICCCI 2009. Lecture Notes in Computer Science, Vol. 5796. Springer, Berlin, Heidelberg. 2009.

[8] B. Das, V. Mukherjee, and D. Das, "Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems", *Advances in Enginering Software*, Vol. 146, 102804, 2020.

[9] E. Mokotoff, "Multi-Objective Simulated Annealing for Permutation Flow Shop Problems", In: Chakraborty U.K. (eds), Computational Intelligence in Flow Shop Scheduling, *Studies in Computational Intelligence*, Vol. 230, Springer, Berlin, Heidelberg, 2009.

[10] J. Deng and L. Wang, "A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem", *Swarm and Evolutionary Computation*, Vol. 32, pp. 121-131, ISSN 2210-6502, 2017.

[11] W. C. Lee and C. C. Wu, "Minimizing The Total Flow Time and The Tardiness in a Two Machine Flow Shop", *International Journal of Systems Science*, Vol. 32, Issue 3, pp. 365-373, 2001.

[12] M. M. Yenisey and B. Yagmahan, "Multi-objective Permutation Flowshop Schedulling Problem: Literature Review, Classification and Curent Trends", *Omega*, Vol 45, pp. 119-135, 2014.

[13] J. M. Pasia, R. F. Hartl, and K. F. Doerner, "Solving a Bi-Objective Flowshop Scheduling Problem by Pareto-Ant Colony Optimization", In: M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stützle. (eds) *Ant Colony Optimization and Swarm Intelligence*. ANTS 2006. Lecture Notes in Computer Science, Vol. 4150. Springer, Berlin, Heidelberg, pp. 294-305, 2006.

[14] M. K. Marichelvam, T. Prabaharan, and X. S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan", *Applied Soft Computing*, Vol 19, pp. 93-101, 2014.

[15] A. Mishra and D. Shrivastava, "A TLBO and a Jaya heuristics for permutation flow shop scheduling to minimize the sum of inventory holding and batch delay costs", *Computers & Industrial* Engineering, Vol 124, pp. 509-522, 2018.

[16] R. Rao, V. J. Savsani, and J. Balic, "Teaching-learning-based optimization algorithm: A novel method for constrained mechanical design optimization problems", *Computer-Aided Design*, Vol. 43, pp.303-315, 2012.

[17] L. Dos Santos Coelho, and V. C. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization", *Expert Systems with Applications*, Vol. 34, No. 3, pp.1905-1913, 2008.

[18] S. Talatahari, B. F. Azar, R. Sheikholeslami, and A. H. Gandomi, "Imperialist competitive algorithm combined with chaos for global optimization", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 17, No. 3, pp. 1312-1319, 2012.

[19] Z. Ma, X. Yuan, S. Han, D. Sun, and Y. Ma, "Improved Chaotic Particle Swarm Optimization Algorithm with More Symmetric Distribution for Numerical Function Optimization", *Symmetry*, Vol 11, 876, 2019.

[20] M. El-Shorbagy, A. Mousa, and S. Nast, "A chaos-based evolutionary algorithm for general

nonlinear programming problems", *Chaos Solitons Fractals*, Vol. 85, pp. 8–21, 2016.

[21] X. S. Yang and S. Deb, "Cuckoo Search via Lévy Flights", In: *Proc. of 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, pp. 210-214, 2009.

[22] X. S. Yang, "Flower Pollination Algorithm for Global Optimization", In: Durand-J. Lose, N. Jonoska (eds) *Unconventional Computation and Natural Computation*. UCNC 2012. Lecturer Notes in Computer Science, Vol. 7445, Springer, Berlin, Heidelberg. 2012.

[23] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm", *Swarm and Evolutionary Computation*, Vol. 44, pp.148-175, 2019.

[24] K. H. Chang, *Design Theory and Method using CAD/CAE*, 1st Edition, Academic Press, Elsevier, 2014.

[25] Y. Donoso, and R. Fabregat, *Multi-Objective Optimization in Computer Networks Using Metaheuristics*, Auerbach Publication, Broken Sound Parkway, 2007.

[26] K. R. Baker, dan D. Trietsch, *Principles of Sequencing and Scheduling*, John Wiley & Sons, Inc., Hoboken, NJ, USA. 2019.

[27] G. Heidari-Bateni and C. D. McGillem, "A chaotic direct-sequence spread spectrum communication system", *IEEE Transaction on Communication*, Vol. 42, pp. 1524-1527, 1994.

[28] Y. Kumar and P. K. Singh, "A chaotic teaching learning based optimization algorithm for clustering problems", *Applied Intelligence*, Vol. 49, pp. 1036-1062, 2019.