Journal of Scheduling (2021) 24:329–346
https://doi.org/10.1007/s10951-021-00683-w

A hybrid evolutionary approach to job-shop scheduling with generic time lags

Madiha Harrabi[1] · **Olfa Belkahla Driss**[2] · **Khaled Ghedira**[3]

Accepted: 19 March 2021 / Published online: 1 June 2021

Abstract
This paper addresses the job shop scheduling problem including time lag constraints. This is an extension of the job shop scheduling problem with many applications in real production environments, where extra (minimum and maximum) delays can be introduced between operations. It belongs to a category of problems known as NP-hard problems due to the large solution space. Biogeography-based optimization (BBO) is an evolutionary algorithm which is inspired by the migration of species between habitats, recently proposed by Simon (IEEE Trans Evol Comput 12:702–713, 2008) to optimize hard combinatorial optimization problems. BBO has successfully solved optimization problems in many different domains and has demonstrated excellent performance. We propose a hybrid biogeography-based optimization (HBBO) algorithm for solving the job shop scheduling problem with additional time lag constraints while minimizing total completion time. In the proposed HBBO, an effective greedy constructive heuristic is adapted to generate the initial habitat population. A local search metaheuristic is investigated in the mutation step in order to improve the solution quality and enhance the diversity of the population. To assess the performance of the HBBO, a series of experiments are performed on well-known benchmark instances for job shop scheduling problems with time lag constraints. The results prove the efficiency of the proposed algorithm in comparison with various other algorithms.

Keywords BBO · Job shop · Scheduling · Optimization · Time lag

1 Introduction

The scheduling problem is one of the most commonly encountered problems in the management of production systems. It involves the allocation of a number of jobs to machines taking into consideration a set of constraints. Scheduling problems occur in all economic domains, from computer engineering to manufacturing techniques. Most scheduling problems are complex combinatorial optimization problems and very difficult to solve. The job shop scheduling problem (JSP) is a branch of production scheduling which is among the hardest combinatorial optimization problems. It is a well-known optimization problem often used in practical scheduling applications in the manufacturing sector. The JSP consists of a set of n jobs that have to be processed on a set of m machines. Each job is fully defined by an ordered sequence of operations that are associated with a particular machine while respecting certain constraints, such as:

(i) No more than one operation of any job can be executed simultaneously.
(ii) No machine can process more than one operation at the same time.
(iii) The job operations must be executed in a predefined sequence, and once an operation is started, no preemption is permitted.

The objective is to schedule each operation on the machines, taking the precedence constraints into account such that a number of optimization criteria are attained, including cost,

✉ Madiha Harrabi
madiha.harrabi@gmail.com

[1] Ecole Nationale des Sciences de l'Informatique, SOIE-COSMOS Laboratory, Université de Manouba, Manouba, Tunisie

[2] Ecole Supérieure de Commerce, SOIE-COSMOS Laboratory, Université de Manouba, Manouba, Tunisie

[3] Université Centrale de Tunis, Honoris United Universities, Eljaziri, Tunisie

run time, and makespan ($C_{\max}$: total completion time) minimization.

In this paper we are interested in the job shop scheduling problem including time lag constraints, which is a special case of resource-constrained project scheduling problems with minimum and maximum time lags between operations of different jobs (González et al. 2015).

The job shop problem including minimum and maximum time lag constraints is a generalization of the job shop scheduling problem, in which there are time relations between the starting times of operations. The job shop scheduling problem including time lag constraints involves two kinds of time lag constraints: either time lags between two successive operations of the same job (job shop scheduling problem with time lags, denoted as JSPTL) or generic time lags between pairs of operations, denoted as JSPGTL. A job shop scheduling problem including minimum and maximum time lag constraints is NP-hard (Lacomme and Tchernev 2012).

The addition of time lag constraints between operations makes even the usually simple task of finding a feasible schedule difficult, and few articles have tackled time lag constraints. In this paper we propose solving the JSPTL-JSPGTL problem with a biogeography-based optimization (BBO) algorithm combined with a greedy constructive algorithm and Tabu search metaheuristic. BBO was introduced in 2008 to solve global optimization problems. It is an evolutionary algorithm that is inspired by the migration of species between habitats. BBO has been demonstrated to be a powerful search technique because it includes both exploration and exploitation strategies based on migration. It is one of the fastest-growing nature-inspired algorithms for solving practical optimization problems. This is a result of its advantages in terms of simplicity, flexibility, and computational efficiency, as well as its stochastic nature, which does not require derivatives of the objective function. Motivated by the effectiveness of this newly emerging evolutionary algorithm in solving different kinds of optimization problems, this study represents the first reported work using the BBO for solving the JSPTL-JSPGTL problem. Our intention is to provide an adaptation of the different parameters of BBO to suit the problem characteristics. We also extend the classic BBO algorithm by adding a constructive greedy algorithm to create the initial population and using the Tabu search algorithm for the mutation step. This hybridization allows us to obtain the best results for the job shop with time lags, which is a very challenging problem for local search metaheuristics, because the classical neighborhood structures for the standard job shop lead to unfeasible schedules most of the time. We conducted an experimental study in which we compared it with the state-of-the-art approach in both the JSPTL and JSPGTL. The experimental results prove that our method is very competitive and that the proposed HBBO algorithm is both effective and efficient. The remainder of the paper is organized as follows. The next section provides related works. In Sect. 3, we give an overview of the job shop scheduling problem with minimum and maximum time lag constraints. In Sect. 4 we explain the concept and structure of the basic BBO algorithm, and we present the adaptation of the HBBO algorithm for our problem in Sect. 5. Section 6 analyzes the performance results of HBBO when applied to solve instances of benchmark problems in the literature. In Sect. 7, we summarize our contributions and propose some avenues for future research.

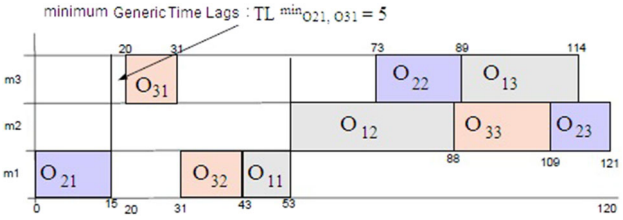

Fig. 1 Minimum time lags

2 Related works

Time lag describes the waiting-time constraints between two consecutive operations in the same job or between two operations of different jobs. Two kinds of time lag constraints can be used in several fields of industrial job shop applications, either minimum or maximum time lags. Minimum time lag constraints can correspond to overlap time, storage time, transit time, or communication time between processes of a computer system. Minimum time lags may be used when waiting time between operations is required for processing, such as cooling (Fondrevelle et al. 2006), material handling (Soukhal et al. 2005), and chemical reactions (Chu and Proth 1996). The minimum time lag between operations (i, j) and (i', j') of different jobs is denoted as $TL^{\min}_{(i,j),(i',j')}$, see Fig. 1. Maximum time lags can be used to demand that the waiting time between operations not be overly long in order to avoid deterioration of products (Hodson et al. 1985). Maximum time lag constraints can be used in the chemical sector. For example, filtration of a product must respond quickly to the formulation to prevent the precipitation of the product. In the agribusiness sector, for example, freezing of meals must occur no later than 30 minutes after cooking; otherwise the meal is declared unfit for human consumption. Similar scenarios can be found in the pharmaceutical and other sectors. The maximum generic time lag between operations (i, j) and (i', j') of different jobs is denoted as $TL^{\max}_{(i,j),(i',j')}$, see Fig. 2.

Minimum and maximum time lag constraints arise in many real-life scheduling applications. For example, in the

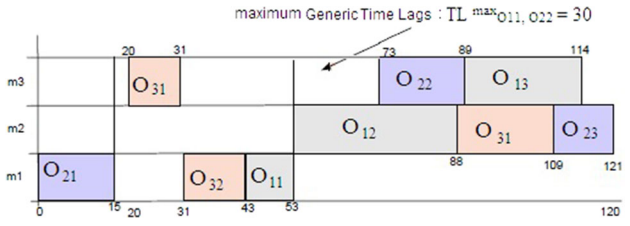

Fig. 2 Maximum time lags

steel industry, the time lag between the heating of a piece of steel and its molding should be small (Wismer 1972). Similarly when scheduling chemical reactions, the reactive component usually cannot be stored for a long period between two stages of a process, so as to avoid interactions with external elements (Rajendran 1994). Many other industrial applications can be found, such as fabrication of printed circuits (Kim et al. 1996), hoist-scheduling problems (Manier and Bloch 2003), perishable product production (Johnson 1954), biotechnology, and chemistry (Nawaz et al. 1983).

Time lag constraints have been introduced in numerous scheduling problems. Mitten (1958) employed time lag constraints for the first time in a problem with parallel machines using a polynomial algorithm to minimize the makespan for the flow shop problem with two machines. Wikum et al. (1994) studied problems with a single machine considering minimum and maximum distances between jobs and proved that these problems are NP-hard, although some particular cases are polynomially solvable. Brucker et al. (1999) showed many examples of scheduling problems that can be modeled as single-machine problems with time lags, including multiprocessor tasks, multipurpose machines, or problems with changeover costs. They also proposed a branch-and-bound method to solve the problem. Hurink and Keuchel (2001) proposed a local search approach for the single-machine problem with positive and negative time lags. Fondrevelle et al. (2006) solved permutation flow shop scheduling problems with maximum and minimum time lags. Botta-Genoulaz (2000) tackled maximum lateness minimization in hybrid flow shop scheduling with precedence constraints and time lags. Another example was presented in the thesis of Zhang (2010), where several variants of online and offline problems with time lags are studied. Time lag constraints have been introduced in many works concerned with solving the resource-constrained project scheduling problem (RCPSP). Bartusch et al. (1988) proposed a scheduling project network with resource constraints and time windows, and Brinkmann and Neumann (1996) proposed heuristic procedures for resource-constrained project scheduling with minimum and maximum time lags. Neumann et al. (2002) studied the project scheduling problem with time windows and scare resources. Heilmann (2003) proposed an exact procedure for a general RCPSP where multiple modes are available for the performance of the individual activities, as well as both minimum and maximum time lags between the different activities. The objective is to determine a mode and a start time for each activity such that all constraints are observed while minimizing the project duration. Hamdi and Loukil (2011) proposed a permutation flow shop problem with maximum and minimum time lags and makespan minimization. Nikbakhsh et al. (2012) proposed an immune algorithm for a hybrid flow shop scheduling problem with time lags and sequence-dependent setup times.

Dhouib et al. (2013) proposed a combination of a simulated annealing algorithm and a mixed-integer mathematical program for solving a permutation flow shop scheduling problem with time lag constraints. Sheikh (2013) investigated a genetic algorithm for solving a multi-objective flexible flow line problem with due window, time lag, and job rejection. Zhao et al. (2017) proposed a universal approach to resolve flow shop scheduling problems with time lags. Ye et al. (2017) studied a non-permutation flow shop scheduling problem with time lags to minimize the makespan.

Different methods have been proposed in the literature for solving the job shop scheduling problem with time lags in the literature. Caumond et al. (2005a) introduced a genetic algorithm based on an operation insertion heuristic. Caumond et al. (2004) proposed a Tabu search metaheuristic, and Caumond et al. (2005b) proposed a constructive heuristic based on Giffer and Thompson's heuristic. Deppner (2004) investigated an approach based on a construction method. Caumond et al. (2008) introduced a memetic algorithm based on a disjunctive graph that is suitable for various industrial situations. Karoui et al. (2010) investigated a climbing discrepancy search method. Artigues et al. (2011) proposed an insertion heuristic and generalized resource constraint propagation approach for the job shop problem. González et al. (2015) proposed a scatter search procedure combining path relinking and the Tabu search metaheuristic of Nowicki and Smutnicki (2005). Afsar et al. (2016) proposed a disjunctive graph model for the job shop problem with time lags and transport. Other methods have been proposed for solving the job shop scheduling problem with generic time lags. Lacomme et al. (2011) proposed a dedicated constraint propagation technique using the Bierwirth vector for presenting a solution, and Lacomme and Tchernev (2012) proposed a set of greedy randomized propagation rules. Harrabi and Belkahla Driss (2016) proposed a Tabu search metaheuristic, while Harrabi et al. (2017a, b, c) proposed a combination of a genetic algorithm and Tabu search metaheuristic. Multi-agent systems have recently been widely used for the resolution of job shop problems. Harrabi and Belkahla Driss (2015) proposed a Tabu search metaheuristic in a multi-agent model. Harrabi et al. (2017a, b, c) proposed the use of parallel Tabu searches in a multi-agent system composed of competitive agents. Harrabi et al. (2017a, b, c) proposed a multi-agent model

based on a hybrid genetic algorithm. Harrabi et al. (2018) proposed a BBO combined with a greedy algorithm for the job shop scheduling problem with time lags between successive operations of the same job. Harrabi et al. (2020) proposed a modified BBO algorithm with an improved mutation operator for the job shop scheduling problem with generic time lags. In this paper, we propose a hybrid BBO algorithm for the job shop scheduling problem with additional minimum and maximum time lag constraints with makespan minimization. Indeed, the popularity of hybrid optimization approaches is rapidly growing as an effective strategy to improve the performance of classical algorithms by combining components from various optimization methods. Population-based optimization algorithms like BBO often have good global exploration ability. However, they are generally not very efficient at local exploitation. In contrast, local search algorithms are efficient at local exploitation but are not effective at exploring the entire search space. Therefore, hybridization of a local search with population-based optimization is a promising way to synergize the advantages of both approaches in a single algorithm. The aim of this type of hybridization is to find the right trade-off between global exploration and local exploitation of the problem search space. Studies have shown that BBO performance can be enhanced through the incorporation of techniques from other metaheuristics (Ma and Simon 2017). Many works in the literature have investigated the hybridization of BBO with other algorithms. Al-Roomi and El-Hawary (2016) presented a combination of BBO and simulated annealing (SA), and Wee et al. (2016) introduced a Tabu search in the mutation step of BBO for the quadratic assignment problem. BBO is also often hybridized with other population-based algorithms. Foe example, many authors have used a hybridization of BBO and differential evolution (DE) to solve various optimization problems. Bhattacharya and Chattopadhyay (2010) and Bhattacharya and Chattopadhyay (2011) investigated an application of BBO and DE for economic emission load dispatch. Wireless sensor network power allocation was solved with BBO/DE in Boussaïd et al. (2011). In Du et al. (2009), BBO was hybridized with an evolutionary strategy (ES). Sinha et al. (2012) proposed a hybridization between BBO and ant colony optimization (ACO). The combination of BBO and particle swarm optimization (PSO) was proposed in Guo et al. (2013). Zheng et al. (2014a, b) introduced a new variation of BBO, called ecogeography-based optimization (EBO), which regards the population of islands (solutions) as an ecological system with a local topology. Two novel migration operators are designed to perform effective exploration and exploitation in the solution space, mimicking the species dispersal under ecogeographic barriers and differentiations. Lu et al. (2018) proposed a biogeography-based memetic algorithm, or BBMA, which redefines the migration and mutation operators of the BBO. They employed a local population topology to suppress premature convergence and used a critical-path-based local search operator to enhance the exploitation ability. Rifai et al. (2018) investigated a non-dominated sorting BBO for a scheduling problem of a flexible manufacturing system (FMS) having multi-loading/unloading and shortcuts infused in the reentrant characteristics. This model is formulated to identify the near optimal trade-off solutions capable of addressing the two objectives of makespan minimization and total earliness. Wu et al. (2019) used a water optimization metaheuristic for the flow shop scheduling problem using a self-adaptive local search procedure to improve the basic algorithm. Zhao et al. (2019) proposed a hybrid BBO with a variable neighborhood search for solving the no-wait flow shop scheduling problem.

3 Job shop scheduling problem with time lags: description and mathematical modeling

3.1 Problem description

The job shop problem with generic minimum and maximum time lags (JSPGTL) is a generalization of the job shop problem in which there are time constraints restricting the minimum and/or the maximum distance between two operations. The JSPGTL involves a set of jobs that should be processed on a set of machines. Each job i consists of a sequence of operations; (i, j) denotes the jth operation of job i. Machines cannot process more than one job simultaneously. Each operation should be allocated to one machine. Every machine and every job is ready at time 0. Each job has a fixed processing sequence; if the process of an operation is started, it should be finished without any interruption. For some pairs of operations (i, j) and (i', j'), there are minimum and maximum time lag constraints, respectively, denoted by $\mathrm{TL}^{\min}_{(i,j),(i',j')}$ and $\mathrm{TL}^{\max}_{(i,j),(i',j')}$, restricting the distance between the end of (i, j) and the start of (i', j') to the interval $[\mathrm{TL}^{\min}_{(i,j),(i',j')}, \mathrm{TL}^{\max}_{(i,j),(i',j')}]$. Solving the JSPGTL consists in sequencing all operations on the machines subject to a set of constraints. There are precedence constraints for operations of the same job, so each job consists of a set of operations that should be sequentially scheduled. Also, there are capacity constraints such that each machine processes at most one operation at a time, and each operation requires the uninterrupted and exclusive use of a given machine during p_{ij} time units. Finally, we consider the addition of minimum and maximum time lags constraints between operations which present the minimum and maximum waiting time between the ending time and the starting time of two operations. The objective is to find an optimal solution according to some

Table 1 Example of instance job shop J

	Operation 1	Operation 2	Operation 3
Job 1	m1, 10	m2, 35	m3, 25
Job 2	m1, 15	m3, 16	m2, 12
Job 3	m3, 11	m1, 12	m2, 21

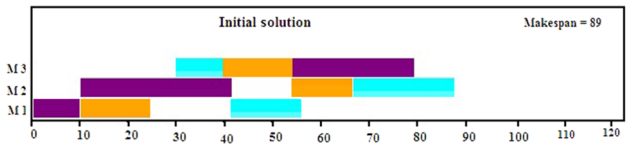

Fig. 3 Example of a Gantt chart solution

criterion, most commonly the makespan, which is the completion time of the last operation.

Example: Table 1 illustrates an example of instance J for a job shop with three jobs and three machines.

The minimum and maximum time lags between operations of different jobs are:

$$TL^{\min}_{(O11),(O13)} = 36 \quad TL^{\max}_{(O11),(O13)} = 45$$
$$TL^{\min}_{(O11),(O22)} = 20 \quad TL^{\max}_{(O11),(O22)} = 30$$
$$TL^{\min}_{(O21),(O31)} = 5 \quad TL^{\max}_{(O21),(O31)} = 20$$

Figure 3 presents an example of a feasible solution in a Gantt chart model for the instance of a job shop with generic time lags given in Table 1.

3.2 Mathematical modeling

The mathematic model of the job shop scheduling problem with generic time lags is formulated as follows:

Input variables

- M = the set of machines;
- J = the set of jobs;
- Ω = the set of operations;
- Ω_μ = the set of operations processed on machine μ.

Decision variables

- p_{ij} = the duration of operation (i, j);
- t_{ij} = the start times of operations (i, j);
- $\mathrm{TL}^{\min}_{(i,j),(i',j')}$ = minimum time lags between operations (i, j) and (i', j');
- $\mathrm{TL}^{\max}_{(i,j),(i',j')}$ = maximum time lags between operations (i, j) and (i', j');
- $x_{(ij),(i'j')}$ = corresponds to the sequencing variables;

$$x_{(ij),(i'j')} = \begin{cases} 1 & \text{if } (ij) \text{ is before } (i'j') \\ 0 & \text{Otherwise.} \end{cases}$$

- H is a large positive number.

Linear programming equations

The problem of linear formulation of a job shop with generic time lags was given by Lacomme et al. (2011) and was inspired by the linear programming formulation for job shop scheduling proposed by (Manne 1960).

$$\mathrm{Min} C_{\max} \tag{1}$$

such that

$$C_{\max} \geq t_{ij} + p_{ij}, \forall (i, j) \in \Omega \tag{2}$$
$$t_{i'j'} \leq t_{ij} + p_{ij} + H.(x_{(ij),(i'j')} - 1), \quad \forall (i, j), (i', j') \in \Omega_\mu, \forall \mu \in M \tag{3}$$
$$t_{ij} \leq t_{i'j'} + p_{i'j'} + H.x_{(ij),(i'j')}, \quad \forall (i, j), (i', j') \in \Omega_\mu, \forall \mu \in M \tag{4}$$
$$t_{i'j'} \geq t_{ij} + p_{ij} + TL^{\min}_{(i,j),(i',j')}, \forall (i, j), (i', j') \in \Omega \tag{5}$$
$$t_{i'j'} \leq t_{ij} + p_{ij} + TL^{\max}_{(i,j),(i',j')}, \forall (i, j), (i', j') \in \Omega \tag{6}$$
$$t_{ij} \geq 0, \forall (i, j) \in \Omega \tag{7}$$
$$x_{(ij),(i'j')} \in \{0, 1\}, \forall (i, j), (i', j') \in \Omega_\mu, \forall \mu \in M \tag{8}$$

Constraint (2) states that the makespan is greater than or equal to the finish time of each operation. Constraints (3) and (4) represent the machine disjunctions. Constraints (5) and (6) correspond to the time lags constraints.

When there is a classical precedence constraint between two operations, the value of the minimum time lag is set to 0, and the value of the maximum time lag is set to ∞.

When there is no constraint between two operations, both the minimum and the maximum time lags are set to ∞.

3.3 Disjunctive graph model

In the disjunctive graph introduced by Roy and Sussmann (1964), each operation is modeled by a node and an arc from operation (i, j) to operation (i', j'), and represents the minimum distance between the start time of these two operations. It corresponds to the binary constraint:

$t_{i'j'} - t_{ij} \geq l_{(ij),(i'j')}$, where $l_{(ij),(i'j')}$ is the length of the arc. The maximum time lag from an operation (i, j) to an operation (i', j') is represented by an arc with negative length which corresponds to the duration of (i, j) plus the maximum time lag value. Minimum time lag constraints are modeled

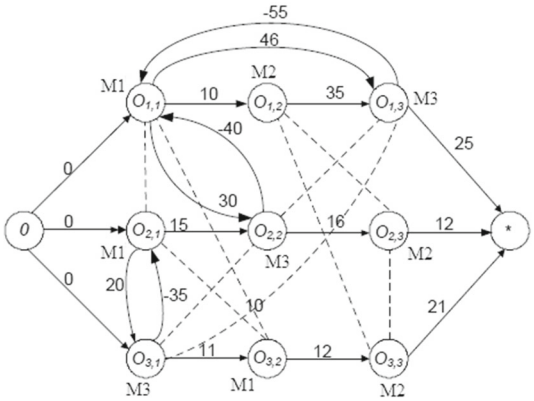

Fig. 4 Disjunctive graph

by an extra arc from operation (i, j) to operation (i', j') and weighted with the processing time of (i, j) plus the minimum time lag value. When no time lags are specified (for example, between one operation and the dummy operation of the graph), it is possible to assume, without loss of generality, that we have null minimum time lags and infinite maximum time lags. Since there is no interest in considering infinite maximum time lags, negative arcs representing infinite maximum time lags are ignored in the graphic representation. To model time lag constraints on the conjunctive/disjunctive graph, we use the formulation based only on the start times of operations $st_{Oi,j}$ (and the processing times of operations). Then, the time lag constraints given in Sect. 3.1 are modeled by:

- $46 \leq st_{O1,3} - st_{O1,1} \leq 55$
- $30 \leq st_{O2,2} - st_{O1,1} \leq 40$
- $20 \leq st_{O3,1} - st_{O2,1} \leq 35$

Figure 4 presents the disjunctive graph of the job shop scheduling problem with generic time lags given in Table 1. Maximum time lag constraints are represented by negative arc cost in the disjunctive graph from one operation to the previous one. The negative cost of the arc is equal to the duration of the previous operation plus the maximum time lag value.

4 Biogeography-based optimization

4.1 Basic concepts of the BBO algorithm

The BBO algorithm, proposed by Simon (2008), is inspired by the mathematics of biogeography, mainly from the work of MacArthur and Wilson (1967). A large number of theoretical, methodological, and practical studies on BBO have since arisen. The two main concepts of BBO are habitat suitability index (HSI) and suitability index variables (SIVs). Features that correlate with the HSI include rainfall, diversity of topographic features, land area, and temperature. SIVs are considered the independent variables of the habitat. Geographical areas that are well suited for species are said to possess a high HSI. Considering the optimization algorithm, a population of candidate solutions can be represented as vectors. Each integer in the solution vector is considered a SIV. In assessing the performance of the solutions, habitats with a high HSI are considered to be good solutions, and habitats with a low HSI are considered to be poor solutions. Therefore, the HSI is analogous to fitness in other population-based optimization algorithms. The two main operators of the BBO are migration and mutation. The main algorithm of the BBO is shown in Algorithm 1.

Algorithm 1 Main algorithm of BBO

Begin
Generate a set of habitats to a problem
Evaluate the fitness value or HSI for each habitat
When Stopping criterion do not met **do**
  Determine immigration rate λ and emigration rate μ for each habitat
  Modify habitats based on λ and μ
  For i = 1 to N (Population size) **do**
    Use λ to probabilistically decide whether to modify a habitat
    If rand (0,1)< λ_i
      Select Habitat H_j through roulette wheel method to emigration
      Perform migration on H_i and H_j
      Evaluate the fitness value or HSI for newly generated solution
      Replace the new solution with H_i
    End
    If rand (0,1) < $P_{Mutation}$
      Apply mutation on H_i
      Evaluate the fitness value or HSI for newly generated solution
    End
  End
  Update Habitat's population
End

4.2 Adaptation of the BBO algorithm in scheduling problems

The BBO algorithm has been successfully used for solving many scheduling problems. Ma et al. (2015) proposed a multi-objective BBO for automated warehouse scheduling, in which a real-world scheduling problem was presented as a constrained multi-objective optimization problem. A railway scheduling application using BBO was presented by Zheng et al. (2014a, b), which derived a mathematical model that considered multiple stations requiring supplies, source stations for storing supplies, and allocation stations for pro-

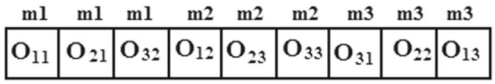

Fig. 5 Habitat representation

viding wagons. Rabiee et al. (2016) developed a modified BBO algorithm for hybrid flow shop scheduling to minimize mean tardiness under various assumptions, including machine eligibility, unrelated parallel machines, different ready times, and sequence-dependent setup times. Habib et al. (2011) introduced a new BBO algorithm for solving the flexible job shop scheduling problem. Wang and Duan (2014) proposed a HBBO algorithm for the job shop scheduling problem in which the proposed HBBO algorithm combines chaos theory and a "searching around the optimum" strategy with the basic BBO, which makes it converge to global optimum solution faster and more stably. Yang (2015) investigated a modified BBO algorithm with a machine-based shifting decoding strategy for solving the flexible job shop scheduling problem. Lin (2016) introduced a hybrid discrete BBO algorithm, or HDBBO, which combined the Nawaz, Enscore, and Ham (NEH) heuristic with opposition-based learning and BBO for the flow shop scheduling problem. The literature reports numerous applications of BBO to benchmarks and practical optimization problems and compares the performance against different well-known algorithms. The results confirm that BBO outperforms the existing algorithms and can efficiently solve most of the benchmark functions. In this work, we use the hybridization of BBO for the job shop scheduling problem with time lags and makespan minimization.

5 Hybrid biogeography-based optimization for the job shop scheduling problem with time lag constraints

In this section, we present the hybridization steps for the BBO algorithm with a Tabu search metaheuristic for solving the job shop scheduling problem with minimum and maximum time lags constraints.

5.1 Representing habitat

We represent a solution with a real number vector containing the total number of operations and the processing order of operations for each machine. Each vector presents a solution. Figure 5 presents an example of the representation of a solution from the problem instance given in Table 1. In Fig. 5, (O_{11}, O_{21}, O_{32}, O_{33}, O_{12}, O_{23}, O_{31}, O_{22}, O_{31}) is a possible habitat for an instance of a job shop scheduling problem with time lag constraints with problem size of 3×3.

We have a vector V containing the operations sequence with length $L = (n \times m)$, equal to the total number of operations ($3 \times 3 = 9$). For each machine, the processing order of operations is given. For example, the processing order of operations for machine 1 is (O_{21}, O_{11}, O_{32}). Each index represents the selected operation to be processed on the machine indicated at position p. For example, $p = 4, V(4)$ is the selected operation O_{21} to be executed on machine m2.

5.2 Initialization of population

The BBO algorithm starts with population habitats. According to the chosen parameter population size PS, an initial population containing PS individuals is generated using the greedy algorithm. This heuristic is a popular method for solving combinatorial optimization and operations research problems (i.e., generally NP-hard). Greedy heuristics are used because they are fast, produce solutions with good quality, are easy to implement, and can easily be expanded. They are commonly used to speed up research. Indeed, in most cases, greedy algorithms have a reduced polynomial time complexity, and their use often leads to better-quality local optima (Talbi 2009). The greedy algorithm starts building the solution from one operation to another. After inserting the operation to a defined position of the current solution, the different constraints are checked. If all constraints are satisfied, we proceed to the next operation. Otherwise this operation is deleted in the current position and added to another position that respects the different constraints, see Algorithm 2.

Algorithm 2 The greedy algorithm

Begin
solution ω ; Set = {operation}
While (Not-empty(Set) and Not-solution.complete())
Select (x, Set) x∈ Set
If $solution.add - possible\ (x) =$ true then solution.add(x)
$x \in$ Set
Else Set.det(x) x∈ Set
End while
End

5.3 Selecting strategies

This step is one of the distinctive steps of BBO with other algorithms, and is executed through two different strategies, one for migration and one for mutation.

5.3.1 Selecting migration strategies

Solutions are selected for immigrating or emigrating according to the immigration rate λ_i and the emigration rate μ_j. According to the concept of the BBO algorithm, during the

migration process, we face two types of selection. Firstly, we should determine whether or not a special habitat H_i should immigrate. To do so, a simple comparison of λ_i with a random number is done. Secondly, we should select habitat H_i for emigrating to H_j. Details of the selection algorithm for migration are shown in Algorithm 3.

Algorithm 3 Selecting migration strategy

```
Begin
Select Rand (Rand ∈ [0,1])
If  rand (0,1) < λi then
    For j= 1 to n
      Select Hj through roulette wheel process
      If Hj is selected
       Using Hi and Hj, the migration process is done
    End if
End if
End
```

5.3.2 Selecting mutation strategies

According to the concept of the BBO algorithm, during the mutation process, a habitat H_i should or should not be mutated. For this, a simple comparison of the mutation probability with a random number is done. Algorithm 4 explains how the mutation selection strategy is performed in the BBO algorithm.

Algorithm 4 Selecting mutation strategy

```
Begin
Select Hi(SIV) according to mutation probability
If Hi(SIV) is selected
    The mutation operator is done
End
```

5.4 Migration operator

Migration is a probabilistic operator that is used for modifying each solution H_i by sharing features among different solutions. The idea of a migration operator is based on the migration in biogeography which shows the movement of species among different habitats. Solution H_i is selected as immigrating habitat with respect to its immigration rate λ_i, and solution H_j is selected as emigrating habitat with respect to its emigration rate μ_j. This means that a solution is selected for immigrating or emigrating depending on its immigration rate λ_i or emigration rate μ_j; the migration process can be shown as:

$H_i(\text{SIV}) \leftarrow H_j(\text{SIV})$ After calculating the HSI for each solution H_i, the immigration rate λ_i and the emigration rate

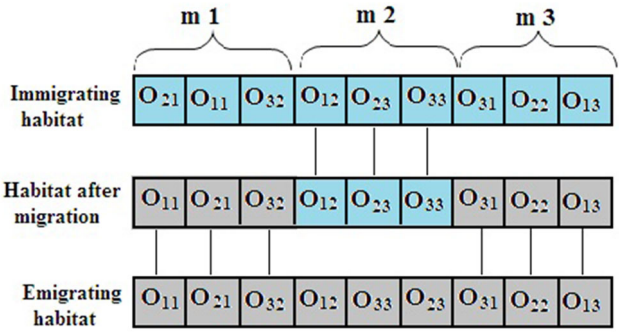

Fig. 6 Migration operator of the BBO algorithm

μ_j can be evaluated as follows:

$$\lambda_i = I\left(1 - \frac{k_i}{n}\right) \tag{9}$$

$$\mu_j = E\left(\frac{k_i}{n}\right) \tag{10}$$

In Eqs. (9) and (10), k_i represents the rank of the ith habitat after sorting all habitats according to their HSIs, and n represents the size of the population. It is clear that since higher HSI indicates a better solution, higher k_i represents the better solution. Therefore, the first solution is the worst, and the nth solution is the best. In the equations, n is the number of habitats in the population, while I represents the maximum immigration rate and E the maximum emigration rate, which are both usually set to 1. The two rates λ_i and μ_j are the functions of fitness or HSI of the solution. Since, according to the biogeography, the SIVs of a high-HSI solution tend to emigrate to low-HSI solutions, a high-HSI solution has a relatively high μ_j and low λ_i, while in a poor solution, a relatively low μ_j and a high λ_i are expected. The migration process can be presented as Algorithm 5 (Wang and Duan 2014).

Algorithm 5 The migration process of BBO

```
Begin
For i = 1 to N
    Use λi to probabilistically decide whether to emigrate to Hi
    If rand (0,1) < λi then
      Select the emigrating island Hj with probability αμj, j ∈ [1,N]
      If rand (0,1) < μj then
      Hj (SIV) ← Hi(SIV)
      End if
    End if
End for
End
```

Figure 6 illustrates an example of migration operator application for our problem, see Fig. 6.

As mentioned earlier, the SIVs from a good habitat tend to migrate into a poor habitat. This migration oper-

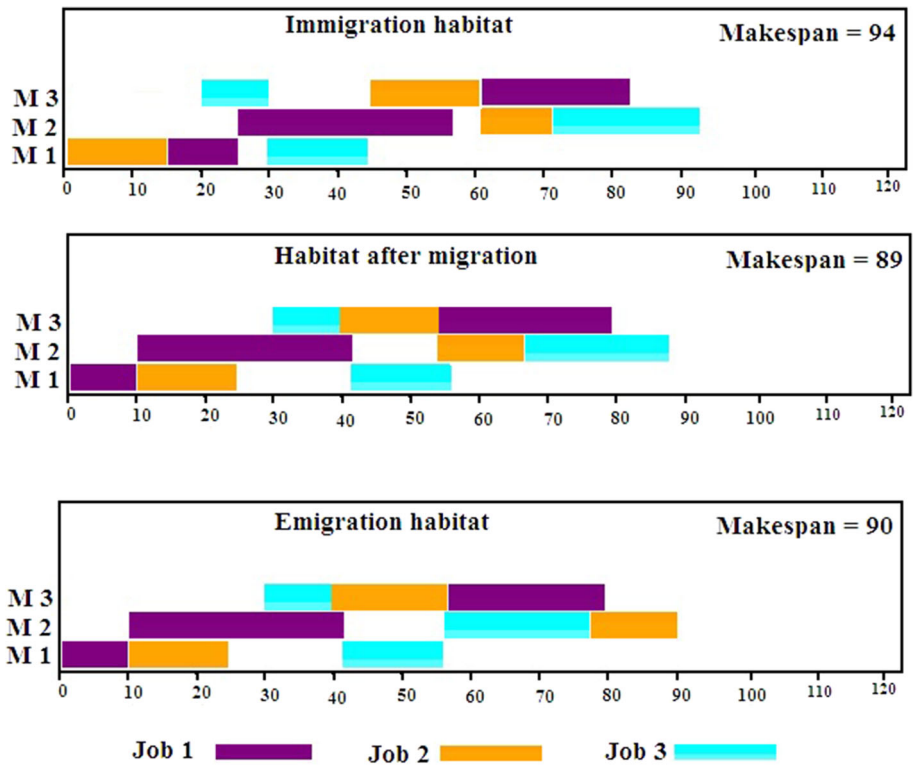

Fig. 7 Gantt charts of migration process solutions

ator is performed probabilistically based on immigration and emigration rates. In this example, we will explain how the migration is implemented in our BBO algorithm. Consider dealing with an instance of a job shop scheduling problem with time lags with three jobs, three operations, and three machines. Suppose, based on immigration and emigration rates, that an immigrating habitat $H_i = (O_{21}, O_{11}, O_{32}, O_{12}, O_{23}, O_{33}, O_{31}, O_{22}, O_{31})$and an emigrating habitat $H_e = (O_{11}, O_{21}, O_{32}, O_{12}, O_{33}, O_{23}, O_{31}, O_{22}, O_{31})$. The migration process is: H_e (SIV) $\leftarrow$ H_i (SIV)

SIVs of H_i will be randomly selected and replace randomly selected SIVs of H_e. Assuming that SIVs of H_i (O_{12}, O_{23}, O_{33}) are selected to replace SIVs of H_e (O_{12}, O_{33}, O_{23}), the migration process consists of the following steps:

(1) SIVs of machine 2 from H_i (O_{12}, O_{23}, O_{33}) migrate into H_e to replace SIVs of H_e (O_{12}, O_{33}, O_{23}).
(2) SIVs (O_{12}, O_{23}, O_{33}) replace SIVs (O_{12}, O_{33}, O_{23}).
(3) SIVs (O_{11}, O_{21}, O_{32}) and (O_{31}, O_{22}, O_{31}) of machine 1 and machine 3 from H_e remain at original places.
(4) Therefore, the new habitat, $H_n = (O_{11}, O_{21}, O_{32}, O_{12}, O_{23}, O_{33}, O_{31}, O_{22}, O_{31})$ is produced, see Fig. 7.

5.5 Mutation operator

Mutation is used to enhance the diversity of the population, which helps to decrease the chances of becoming trapped in the local optima. Solutions with very high HSI and very low HSI are both equally improbable, while medium-HSI solutions are relatively probable to mutate. Namely, a randomly generated SIV replaces a selected SIV in the solution H_i according to a mutation probability. Note that an elitism approach is employed to save the features of the habitat that has the best solution in the BBO process and guarantees the survival of the best individual(s). The habitat with the best solution has a mutation rate of 0. Mutation is a probabilistic operator that randomly modifies a solution's SIV based on its priori probability of existence. The probability Ps, which represents that the habitat contains exactly species S, changes from time t to time $t + \Delta t$.

It is updated as follows:

$$P_s(t+\Delta t) = P_s(t)(1-\lambda_s \Delta t - \mu_s \Delta t) + P_{s-1}\lambda_{s-1}\Delta t + P_{s+1}\mu_{s+1}\Delta t \quad (11)$$

Species count probabilities P_s computed from λ_i and μ_j with equation (11) are used to determine the mutation rates. Suppose a habitat with S species is selected to execute the mutation operation, then randomly modify a chosen variable (SIV) based on its associated probability P_s. The mutation rate $m(s)$ can be computed according to the following function proportional to P_s:

$$m(s) = m_{\max}\left(1 - \frac{P_s}{P_{\max}}\right) \quad (12)$$

Suppose that habitat $H_i \in \text{SIV}^R$ represents a feasible solution to some problem; the mutation process can be presented in Algorithm 6 (Wang and Duan 2014).

Algorithm 6 The mutation process of BBO

```
Begin
  For i = 1 to R
    Compute the probability P_s with equation 11
    Compute the mutation rate m(s) with equation 12
    Select H_j with probability α m(s)
    If H_j is selected then
      Replace H_j with randomly generated SIV
    End if
  End for
End
```

If a solution is selected for mutation, it is replaced by a randomly generated new solution set. This random mutation affects the exploration ability of the BBO algorithm (Feng et al. 2017). The mutation habitats are kept in the population only if the quality is better than the original habitats. However, this is not practical when solving a complex optimization problem such as job shop scheduling with time lag constraints. Most of the time, the resulting habitats from a simple mutation operator are unlikely to be better than the original habitats, especially as the algorithm converges. To overcome this weakness of the classical BBO algorithm, we propose replacing the mutation operator with a Tabu search (TS) procedure. Proposed by Glover (1986), TS is a metaheuristic which performs a local search based on the information in the memory. TS is both a neighborhood-based and iterative procedure. At each iteration, the current solution will make a move to the neighborhood solution with the best objective function value. To avoid trapping in local optima, the move that has been made will be stored in a Tabu list, and a reverse move to previous solutions is forbidden. The performance of Tabu search is highly dependent on the neighborhood type used and Tabu list implementation. The Tabu search metaheuristic has been successfully used for solving different combinatorial optimization problems. The advantages of replacing the mutation operator of the classical BBO algorithm with TS are twofold. First, the original aim of the mutation process is maintained, which is to increase the diversity of the population. At the same time, the quality of the resulting habitats is prevented from being degraded (Wee et al. 2016). Different parameters of the TS metaheuristic are presented in the following sections.

5.5.1 Initial solution

The initial solution is the starting step used for the algorithm to begin the search for better configurations in the search space. In this implementation, the initial solution used is the resulting solution of the migration step of the BBO algorithm; the Tabu search process then proceeds iteratively to visit series of locally best configurations following a neighborhood function, see Fig. 8.

5.5.2 Neighborhood structure

A neighborhood is a set of solutions (neighbors) created by a specific operator.

A move is a function transforming a solution into one of its neighbors. In the literature, we can find many types of moves. Among the most common we can cite:

Swapping moves: this type of neighborhood is created by reversing two successive elements in the current solution. This movement generates a neighborhood of size $(n-1)$, which can be explored in $O(n^2m)$, see Fig. 9.

Exchange moves: this type of neighborhood is created by swapping the positions p_i and p_j of any two elements. This movement generates a neighborhood of size $n(n-1)/2$, and exploring the neighborhood reaches $O(n^3m)$, see Fig. 10.

Insertion moves: this type of neighborhood is created b moving an element from its original position p_i to a new position p_j. This movement generates a neighborhood of size $(n-1)^2$, but it can be evaluated in $O(n^2m)$, see Fig. 11.

Deroussi et al. (2006) proposed a study of different neighborhood structures and showed that local search algorithms based on swap moves do not allow one to reach good-quality solutions, and local search algorithms based on exchange moves provide good-quality local minima, but the neighborhood exploration is in $O(n^3m)$, which can cause considerable computation time for large instances.

In this implementation, we use the insertion move structure, which is considered the most effective and most commonly used in several approaches described in the literature because of its efficiency in terms of quality of the solution and running time. Figure 12 illustrates an example of an insertion move neighborhood structure for the job shop scheduling problem with time lag constraints.

As mentioned earlier, the insertion move neighborhood structure is performed by simply moving a selected operation from its original position p_i to a new position p_j. Assuming that the operation O_{13} is chosen to be moved, the resulting neighbor solution produced is shown in Fig. 13.

5.5.3 Tabu List

The Tabu list (TL) can be used to avoid searching the previously tested solutions of the minimization or maximization problem. The TL stores the attributes of the moves that have been made. When the length of the list reaches the maximum fixed value $L_{\max}$, before adding the next element, the oldest one is deleted. When, in a given step, all the moves are for-

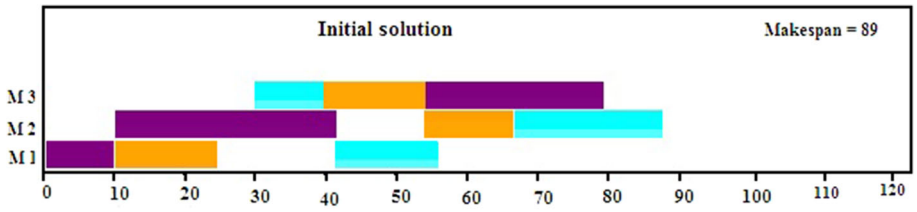

Fig. 8 Gantt chart of initial solution

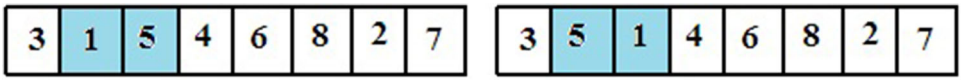

Fig. 9 Swapping moves

Fig. 10 Exchange moves

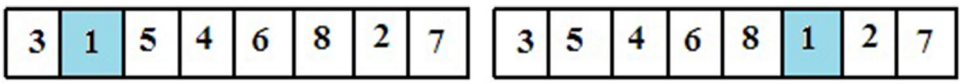

Fig. 11 Insertion moves

bidden by the Tabu list, the oldest elements are deleted from the list until at least one move is allowed.

5.5.4 Diversification

In order to ensure sufficient diversity in the search, a specific diversification must be incorporated to complement the neighborhood. The diversification step is activated when the number of iterations continuously increases without any improvement in the current solution, which means that the best solution found has not been replaced by one of these neighbors for some time, which is a sign that the Tabu search was probably trapped in a local optimum. In this case, a simple effective method is used to achieve diversity. A new schedule is generated using a new insertion order of jobs in order to explore a new region of the search space, and the resolution process is started again. Next, the number of iterations diversification, i.e., the number of iterations after the last improvement, is reset, and the research process continues by considering the solution obtained by diversification phase as a new current solution until reaching the stopping criterion.

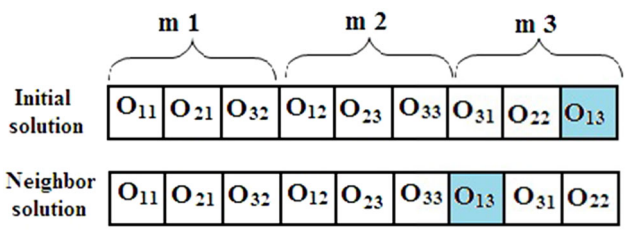

Fig. 12 Insertion moves structure

6 Experimental results

In order to evaluate the performance of the proposed hybrid biogeography-based optimization algorithm for the job shop scheduling problem with minimum and maximum time lag constraints, we give in this section the results of the HBBO algorithm for the job shop scheduling problem with time lags between successive operations of the same job (JSPTL) and the results of the job shop scheduling problem with generic time lags between whatever pairs of operations of different jobs (JSPGTL). Several experiments were conducted on a set of benchmarks for job shop problems with additional time lag constraints existing in the literature.

6.1 Experimental results for JSPTL

For the JSPTL problem, we use the instances of Fisher and Thompson (1963), Lawrence (1984), and Carlier (1978). For instances of Fisher and Thompson (1963) and Lawrence (1984), we compare the results of the HBBO algorithm with:

- Generalized disjunctive constraint propagation based on a job insertion heuristic JI (Artigues et al. 2011)
- Approach based on operation insertion heuristic OI (Caumond et al. 2005b)
- Tabu Search algorithm TS (Caumond et al. 2004)
- Multi-agent model based on Tabu search metaheuristic MATS (Harrabi and Belkahla Driss 2015)
- Competitive agents implementing parallel Tabu searches CAPTS (Harrabi et al. 2017a, b, c)
- Greedy biogeography-based optimization GBBO (Harrabi et al. 2018)

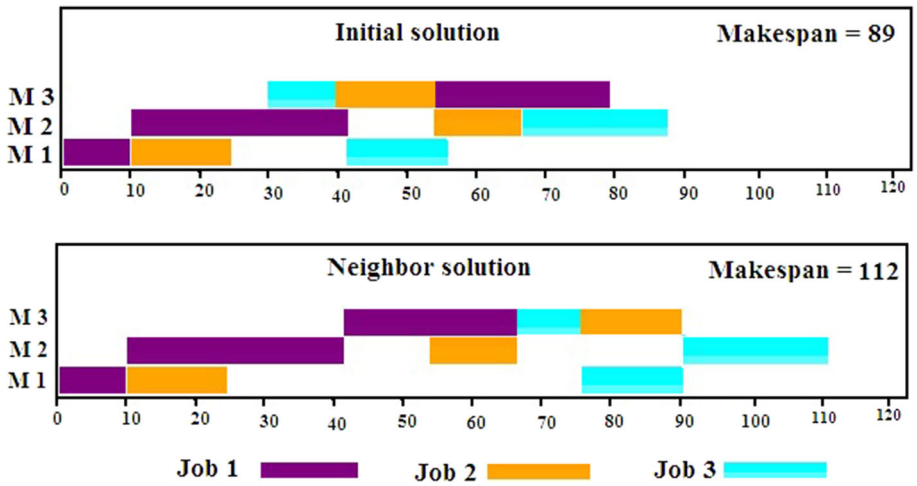

Fig. 13 Gantt charts of insertion moves neighborhood

Table 2 Results for JI/OI/TS/MATS/CAPTS/GBBO/HBBO for Fisher and Thompson instances

Instances	$n * m$	Lower bound	JI heuristic	OI heuristic	TS	MATS-JSTL	CAPTS-JSTL	Greedy BBO	HBBO
ft06_0_0	$6 * 6$	73	96	83	94	80	73	83	73*
ft06_0_0.5	$6 * 6$	63	72	109	81	69	66	73	69
ft06_0_1	$6 * 6$	58	72	58	61	64	60	61	60
ft06_0_2	$6 * 6$	55	70	55	55	60	57	55	55*

For instances of Carlier (1978), we compare the results of the HBBO algorithm with:

- Tabu search algorithm TS (Caumond et al. 2004)
- Memetic algorithm Mem (Caumond et al. 2008)
- Multi-agent model based on Tabu search metaheuristic MATS (Harrabi and Belkahla Driss 2015)
- Competitive agents implementing parallel Tabu searches CAPTS (Harrabi et al. 2017a, b, c)
- Greedy biogeography-based optimization GBBO (Harrabi et al. 2018)

For all instances, $\mathrm{TL}^{\min}_{(i,j),(ij+1)} = 0$.
For each instance, $\mathrm{TL}^{\max}_{(i,j),(ij+1)} = \{0, 0.5, 1, 2\}$.
Instances are designed with Name$\mathrm{TL}^{\min}_{(i,j),(ij+1)}\,\mathrm{TL}^{\max}_{(i,j),(ij+1)}$. For example, ft06-0-0.5 is the instance of Fisher and Thompson 6 with $\mathrm{TL}^{\min}_{(i,j),(ij+1)} = 0$ and $\mathrm{TL}^{\max}_{(i,j),(ij+1)} = 0.5$. For some instances, HBBO gives an optimal solution with a makespan value equal to the lower bound. We mark the values of these instances with "*".

6.1.1 Comparison results of Fisher and Thompson instances for the JSPTL problem

Table 2 presents results for JI/OI/TS/MATS/CAPTS/GBBO/HBBO for Fisher and Thompson instances.

For the ft06 instances, the proposed HBBO algorithm gives better results than the job insertion heuristic method in 100% of instances and better results than TS, MATS, and GBBO in 75%. Compared with CAPTS, HBBO gives better results in 25% and gives 50% better than the OI heuristic (see Table 2).

6.1.2 Comparison results of Lawrence instances for the JSPTL problem

Table 3 presents results for JI/OI/TS/MATS/CAPTS/GBBO/HBBO for Lawrence instances.

For Lawrence instances with ten jobs and five machines, results show that the BBO algorithm gives better results than the job insertion heuristic and Tabu search heuristic in 95% of instances, and better results than the operation insertion heuristic, MATS, CAPTS, and Greedy BBO in 100% of instances, see Table 3.

6.1.3 Comparison results of Carlier instances for the JSPTL problem

Table 4 presents results for TS/Mem/MATS/CAPTS/GBBO/HBBO for Carlier instances.

For Carlier's instances, results show that the HBBO algorithm gives better makespan values than the Tabu search in 87.5% of instances and better results than the memetic algorithm in 40% of instances. Compared with CAPTS and GBBO, HBBO gives better results in 25% and better results than MATS in 50%. See Table 4.

Table 3 Results for JI/OI/TS/MATS/CAPTS/GBBO/HBBO for Lawrence instances

Instances	$n * m$	Lower bound	JI heuristic	OI heuristic	TS	MATS- JSTL	CAPTS- JSTL	Greedy BBO	HBBO
la01_0_0	5 * 10	971	1258	1504	1473	1020	1012	1133	984
la01_0_0.5	5 * 10	758	1063	1474	758	980	924	959	817
la01_0_1	5 * 10	683	928	1114	916	907	891	934	751
la01_0_2	5 * 10	666	967	948	732	894	856	889	666*
la02_0_0	5 * 10	937	1082	1416	1436	1085	1034	1103	954
la02_0_0.5	5 * 10	742	1011	1207	1153	973	952	1019	895
la02_0_1	5 * 10	686	935	1136	900	694	686	874	686*
la02_0_2	5 * 10	673	928	895	681	874	673	801	673*
la03_0_0	5 * 10	820	1081	1192	1108	1178	1018	1059	917
la03_0_0.5	5 * 10	679	930	1085	1052	946	874	966	843
la03_0_1	5 * 10	640	886	931	847	894	718	837	689
la03_0_2	5*10	630	808	787	671	761	630	651	630*
la04_0_0	5*10	887	1207	1346	1275	1393	1237	1250	914
la04_0_0.5	5*10	703	870	1156	1106	1234	1015	906	882
la04_0_0	5*10	646	1010	857	950	915	826	867	784
la04_0_ 2	5*10	619	892	838	642	827	783	804	724
la05_0_0	5*10	757	1080	1224	1128	1168	868	1049	758
la05_0_0.5	5*10	622	935	1208	957	962	817	976	634
la05_0_1	5*10	593	814	964	761	803	624	736	614
la05_0_2	5*10	593	749	683	600	763	593	700	593*

Table 4 Results for TS/MEM/MATS/CAPTS/GBBO/HBBO for Carlier instances

Instances	$n * m$	Lower bound	TS	MEM	MATS-JSTL	CAPTS-JSTL	GBBO	HBBO
Car5_0_0	10*6	7821	11495	7821	8346	8145	9406	9218
Car 5_0_0.5	10*6	7821	10293	7821	7815	7762	8854	8753
Car 5_0_1	10*6	7805	8910	7821	7748	7724	7805	7805
Car 5_0_2	10*6	7700	8281	7702	7732	7702	7894	7702*
Car 6_0_0	8*9	8313	11243	8313	8407	8367	8453	8411
Car 6_0_0.5	8*9	8300	9100	8330	8387	8324	8312	8306
Car 6_0_1	8*9	8323	9248	8313	8362	8319	8313	8323*
Car 6_0_2	8*9	8305	8467	8505	8357	8313	8417	8406
Car 7_0_0	7*7	6558	7704	6558	6704	6617	7528	7467
Car 7_0_0.5	7*7	6558	6953	6558	6624	6593	6958	9928
Car 7_0_1	7*7	6573	6590	6558	6602	6574	6609	6573*
Car 7_0_2	7*7	6558	6573	6558	6568	6561	6572	6564
Car 8_0_0	8*8	8407	10144	8407	8409	8394	8423	8407*
Car 8_0_0.5	8*8	8407	8856	8407	8367	8338	8459	8446
Car 8_0_1	8*8	8279	8833	8264	8326	8287	8304	8279
Car 8_0_2	8*8	8259	8586	8279	8302	8264	8285	8267

6.2 Experimental results for the JSPGTL problem

For the JSPGTL problem, we use the instances of Lawrence (1984) and Carlier (1978).

For instances of Carlier, we compare the results of the proposed HBBO algorithm with:

- Dedicated constraint propagation DCP (Lacomme et al. 2011)
- Greedy randomized propagation rules modified ARP-MD Lacomme and Tchernev (2012)
- Tabu search (TS) metaheuristic (Harrabi and Belkahla Driss 2016);

Table 5 Results for DCP/GR-ARP-MD/TS/GATS/MAHGA/GBBO/HBBO for Carlier instances

Instances	$n * m$	Lower bound	DCP	GR-ARP-MD	TS	GATS	MAHGA	GBBO	HBBO
Car1	11 * 5	7038	8574	13788	9817	9264	8960	8628	8610
Car2	13 * 4	7166	7777	/	6358	7633	7633	7725	7693
Car3	12 * 5	7312	9025	/	8217	9025	8964	8961	8946
Car4	14 * 4	8003	8787	/	8624	8619	8513	8619	8600
Car5	10 * 6	7702	9867	13597	8415	10163	10163	10127	10084
Car6	8 * 9	8313	9404	/	8634	9683	9608	9467	9428
Car7	7 * 7	6558	8746	10948	9215	8911	8853	8738	8719
Car8	8 * 8	8264	11317	16130	10820	12257	12173	11308	11283

- Genetic algorithm combined with the Tabu search (GATS) metaheuristic (Harrabi et al. 2017a, b, c);
- Multi-agent model based on the hybrid genetic algorithm (MAHGA) (Harrabi et al. 2017a, b, c);
- Greedy biogeography-based optimization (GBBO) (Harrabi et al. 2018).

6.2.1 Comparison results of Lawrence instances for the JSPGTL problem

Table 5 presents results for DCP/GR-ARP-MD/TS/GATS/MAHGA/GBBO/HBBO for Carlier instances.

For Carlier instances, results show that the HBBO algorithm gives better results than the dedicated constraint propagation in 62% of instances and better results than the greedy randomized propagation rules and greedy biogeography-based optimization in 100% of instances. Compared with the Tabu search, HBBO gives better results in 37%, and compared with the hybrid genetic algorithm, the HBBO algorithm gives better results in 87% of instances and better results in 75% than the multi-agent model based on the hybrid genetic algorithm.

6.2.2 Comparison results of Lawrence instances for the JSPGTL problem

Table 6 presents results for DCP/GR-ARP-MD/TS/GATS/MAHGA/GBBO/HBBO for Lawrence instances.

For Lawrence instances, results show that the HBBO algorithm gives better results than the dedicated constraint propagation in 82% of instances and better results than the greedy randomized propagation rules and greedy biogeography-based optimization in 100% of instances. Compared with the Tabu search, HBBO gives better results in 65%, and compared with hybrid genetic algorithm, the HBBO algorithm gives better results in 85% of instances and better results in 62% than the multi-agent model based on the hybrid genetic algorithm.

6.3 Analysis of results

The proposed hybrid biogeography-based optimization was evaluated by using 88 well-studied instances with different sizes of problem; 40 instances for a job shop with time lags and 48 instances for a job shop with generic time lags in order to prove its efficiency and effectiveness. These problem instances are commonly utilized for benchmarking job shop scheduling with time lags with the objective of minimizing makespan.

Results show that hybrid BBO gives 14 optimal solutions from 80 instances compared with the lower bound found using CPLEX, proposed (by Lacomme and Tchernev 2012)

The proposed hybrid biogeography-based optimization was able to achieve 11 optimal solutions for the following instances of a job shop with time lags: ft06-0-0, ft06-0-2, la01-0-2, la02-0-1, la02-0-2, la03-0-2, Car5-0-1, Car5-0-2, Car6-0-1, Car 7-0-1, Car 8-0-1.

In addition, hybrid BBO was able to achieve three optimal solutions for the following instances of a job shop with generic time lags: la01, la12, la22.

Moreover, hybrid BBO was able to achieve near-optimal schedules for the majority of all problem instances. It is clear that the proposed optimization technique achieved 36% of the optimal solutions for JSPTL and near-optimal solutions for the rest of instances.

For the job shop with generic time lags, the best-known solution considered is the DCP approach. The proposed hybrid biogeography-based optimization gives the best-known solutions in some instances (la01, la04) and gives for the remaining instances the near-optimal schedules in which they were generally better than those of the other algorithms.

Based on the above results, it appears that the proposed technique was able to provide optimal schedules in some cases and near-optimal schedules in most cases. Based on these results, the proposed approach can be considered as an efficient algorithm for solving job shop scheduling problems with time lags and generic time lags.

Table 6 Results for DCP/GR-ARP-MD/TS/GATS/MAHGA/GBBO/HBBO for Carlier instances

Instances	$n * m$	Lower bound	DCP	GR-ARP-MD	TS	GATS	MAH GA	GBBO	HBBO
la01	10 * 5	666	666	875	654	684	684	694	666*
la02	10 * 5	655	697	897	718	854	811	772	753
la03	10 * 5	597	636	/	684	738	738	658	627
la04	10 * 5	590	713	/	627	713	713	737	713
la05	10 * 5	573	593	878	634	622	622	588	576
la06	15 * 5	926	926	/	867	1132	1104	1085	1037
la07	15 * 5	780	894	1123	837	852	834	847	829
la08	15 * 5	663	907	/	883	962	943	728	704
la09	15 * 5	851	951	/	904	928	906	917	917
la10	15 * 5	858	958	/	923	924	911	903	903
la11	20 * 5	1022	1222	/	1304	1054	1022	1032	1024
la12	20 * 5	1059	1039	1575	986	1168	1136	1074	1059*
la13	20 * 5	1005	1150	/	1024	918	918	1043	1027
la14	20 * 5	1192	1292	1584	1137	1234	1207	1258	1241
la15	20 * 5	1109	1207	1593	1194	1109	1093	1121	1109
la16	10 * 10	945	1114	1599	1084	1114	1114	1208	1181
la17	10 * 10	787	1091	1292	1128	1128	1104	1124	1108
la18	10 * 10	848	1076	/	1036	1186	1186	1094	1094
la19	10 * 10	842	1050	1403	964	1050	1050	1059	1052
la20	10 * 10	902	1142	1635	1019	1419	1376	1212	1204
la21	15 * 10	1046	1181	1795	1146	1093	1038	1084	1078
la22	15 * 10	927	1028	/	984	973	924	934	927*
la23	15 * 10	1032	1054	/	1035	1267	1219	1071	1064
la24	15 * 10	935	1054	/	1078	1017	993	1023	1009
la25	15 * 10	977	1069	1736	1031	1028	1028	1062	1053
la26	20 * 10	1018	1306	1877	1269	1154	1079	1114	1107
la27	20 * 10	1035	1408	/	1227	1186	1128	1154	1116
la28	20 * 10	1135	1325	1997	1271	1216	1191	1185	1167
la29	20 * 10	1057	1308	/	1294	1286	1264	1273	1238
la30	20 * 10	1135	1395	/	1319	1267	1238	1237	1218
la31	30 * 10	1704	1890	2543	1722	1718	1718	1721	1718
la32	30 * 10	1750	1986	2500	1837	1822	1786	1771	1753
la33	30 * 10	1519	1790	/	1819	1604	1592	1619	1604
la34	30 * 10	1721	1962	/	1894	1874	1874	1891	1864
la35	30 * 10	1888	2128	/	2089	2063	2037	2038	2016
la36	15 * 15	1268	1350	1747	1309	1350	1350	1354	1327
la37	15 * 15	1397	1566	2452	1617	1604	1604	1570	1564
la38	15 * 15	1196	1295	1725	1238	1218	1218	1229	1207
la39	15 * 15	1233	1390	/	1367	1286	1217	1256	1248
la40	15 * 15	1122	1320	/	1286	1224	1194	1208	1194

7 Conclusion

Over the past few decades, a number of new types of algorithms have been developed for solving optimization problems. Biogeography-based optimization is a new simulated bio-inspired intelligent algorithm which has some features that are distinct from other biology-based optimization methods. We propose an improved biogeography-based optimization algorithm hybridized with the Tabu search metaheuristic for solving the job shop scheduling problem including minimum and maximum time lag constraints, which is an NP-hard problem, and obtaining a feasible solu-

tion is a difficult problem. According to an analysis and comparisons of the test results of HBBO through different instances of job shop scheduling problems with additional time lag constraints described in the literature, this algorithm is better able to solve well-known benchmarks. Compared with other proposed algorithms, the HBBO algorithm has significantly better performance. Based on the good results of the proposed HBBO algorithm, it can be used to solve other extensions of our problem. We can develop the hybridization of BBO with other algorithms such as ACO, PSO, or GA in order to solve the same problem. We can also adopt the distributed BBO via the multi-agent system.

References

Afsar, H. M., Lacomme, P., Ren, L., Prodhon, C. & Vigo, D. (2016). Resolution of a job-shop problem with transportation constraints: A master/slave approach. In *IFAC conference on manufacturing modelling, management and control*.

Al-Roomi, A., & El-Hawary, M. (2016). Metropolis biogeography-based optimization. *Information Sciences*, *360*, 73–95.

Artigues, C., Huguet, M., & Lopez, P. (2011). Generalized disjunctive constraint propagation for solving the job shop problem with time lags. *Engineering Applications of Artificial Intelligence*, *24*, 220–231.

Bartusch, M., Mohring, R. H., & Rademacher, F. J. (1988). Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, *16*, 201–240.

Bhattacharya, A., & Chattopadhyay, P. (2010). Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Transactions on Power Systems*, *25*(4), 1955–1964.

Bhattacharya, A., & Chattopadhyay, P. (2011). Hybrid differential evolution with biogeography-based optimization algorithm for solution of economic emission load dispatch problems. *Expert Systems With Applications*, *38*(11), 14001–14010.

Botta-Genoulaz, V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*, *64*, 101–111.

Boussaïd, I., Chatterjee, A., & Siarry, P. (2011). Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, *60*(5), 2347–2353.

Brinkmann, K., & Neumann, K. (1996). Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: The minimum project-duration and resource-levelling problems. *Journal of Decision Systems*, *5*, 129–156.

Brucker, P., Hilbig, T., & Hurink, J. (1999). A branch and bound algorithm for a single machine scheduling with positive and negative time-lags. *Discrete Applied Mathematics*, *120*, 77–99.

Carlier, J. (1978). Ordonnancements á contraintes disjunctives. *RAIRO Recherche operationelle, 12*, 333–351.

Caumond A. A., Lacomme, P. & Tchernev, N. (2005a). Proposition of genetic algorithm for job-shop with time-lags. ROADEF'05 (pp. 183–200).

Caumond, A., Gourgand, M., Lacomme, P., & Tchernev, N. (2004). Metaheuristics for job-shop with time lags. In *Proceedings of the conference Mosim's 04* (pp. 939–946). Nantes, France.

Caumond, A., Lacomme, P., & Tchernev, N. (2005b). Feasible schedule generation with extension of the Giffler and Thompson's heuristic for the job shop problem with time lags. In *International conference of industrial engineering and systems management* (pp. 489–499).

Caumond, A., Lacomme, P., & Tchernev, N. (2008). A memetic algorithm for the job-shop with time-lags. *Computers & Operations Research*, *35*, 2331–2356.

Chu, C., & Proth, J. M. (1996). Single machine scheduling with chain structured precedence constraints and separation time windows. *IEEE Transactions on Robotics and Automation*, *12*, 835–843.

Deppner, F. (2004). *Ordonnancement d'atelier avec contraintes temporelles entre opérations*. Ph.D. thesis, Institut National Polytechnique de Lorraine.

Deroussi, L., Gourgand, M., & Norre, S. (2006). New effective neighborhoods for the permutation flow shop problem.

Dhouib, E., Teghem, J., & Loukil, T. (2013). Lexicographic optimization of a permutation flow shop scheduling problem with time lag constraints. *International Transactions in Operational Research*, *20*(2), 213–232.

Du, D., Simon, D., & Ergezer, M. (2009). Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In *Proceedings of IEEE international conference on systems, man and cybernetics, San Antonio* (pp. 997–1002).

Feng, S.-L. Yang, Z.-Q. Huang, M.-X. (2017). Hybridizing adaptive biogeography-based optimization with differential evolution for multi-objective optimization problems. *Information, 8*, 83.

Fisher, H., & Thompson, G. L. (1963). *Probabilistic learning combination of local job shop scheduling rules. Industrial scheduling* (pp. 225–251). Upper Saddle River: Prentice Hall.

Fondrevelle, J., Oulamaraa, A., & Portmann, M.-C. (2006). Permutation flowshop scheduling problems with maximal and minimal time lags. *Computers & Operations Research*, *33*, 1540–1556.

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

González, M. A., Oddi, A., Rasconi, R., & Varela, R. (2015). Scatter search with path relinking for the job shop with time lags and set up times. *Computer & Operation Research*, *60*, 37–54.

Guo, W., Li, W., Zhang, Q., Wang, L., & Wu, Q. (2013). Biogeography-based particle swarm optimization with fuzzy elitism and its applications to constrained engineering problems. *Engineering Optimization*, *46*(11), 1–20.

Habib, S., Rahmati, A., & Zandieh, M. (2011). A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *58*(9), 1115–1129.

Hamdi, I., & Loukil, T. (2011). Minimizing the makespan in the permutation flowshop problem with maximal and minimal time lags. In *Communications, computing and control applications (CCCA)* (pp. 1–6).

Harrabi, M., & Belkahla Driss, O. (2015). MATS-JSTL: A multi-agent model based on Tabu search for the job shop problem with time lags. In *International computational collective intelligence technologies and applications ICCCI* (pp. 39–46).

Harrabi, M., & Belkahla Driss, O. (2016). Tabu Search metaheuristic for job shop scheduling problem with generic time lags. In *International conference on decision aid sciences and applications DASA*.

Harrabi, M., Belkahla Driss, O., & Ghedira, K. (2017a). Competitive agents implementing parallel Tabu searches for job shop scheduling problem with time lags. In *IASTED international conference on modelling, identification and control* (pp. 848–052).

Harrabi, M., Belkahla Driss, O., & Ghedira, K. (2017b). Combining genetic algorithm and Tabu search for job shop scheduling problem with time lags. In *IEEE international conference on engineering & MIS*.

Harrabi, M., Belkahla Driss, O., & Ghedira, K. (2017c). A multi-agent model based on hybrid genetic algorithm for job shop schedul-

ing problem with generic time lags. In *ACS/IEEE international conference on computer systems and applications AICCSA*.

Harrabi, M., Belkahla Driss, O., & Ghedira, K. (2018). A greedy biogeography-based optimization algorithm for job shop scheduling problem with time lags. In M. Graña, et al. (Eds.), *International joint conference SOCO'18-CISIS'18-ICEUTE'18. SOCO'18-CISIS'18-ICEUTE'18. Advances in intelligent systems and computing* (Vol. 771). Springer.

Harrabi, M., Belkahla Driss, O., & Ghedira, K. (2020). A modified biogeography-based optimization algorithm with improved mutation operator for job shop scheduling problem with time lags. *Logic Journal of the IGPL*, jzaa037.

Heilmann, R. (2003). A branch-and-bound procedure for the multimode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, *144*(144), 348–365.

Hodson, A., Muhlemann, P., & Price, D. H. R. (1985). A microcomputer based solution to a practical scheduling problem. *Journal of the Operational Research Society*, *36*(10), 903–914.

Hurink, J., & Keuchel, J. (2001). Local search algorithms for a single machine scheduling problem with positive and negative time-lags. *Discrete Applied Mathematics*, *112*, 179–197.

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics*, *1*, 61–68.

Karoui, W., Huguet, M. J., Lopez, P., & Haouari, M. (2010). Méthode de recherche a divergence limitée pour les *problèmes* d'ordonnancement avec contraintes de délais. In *Conférence Internationale de Modélisation et Simulation* (Vol. 8, pp. 10–12).

Kim, Y. D., Lim, H. G., & Park, M. W. (1996). Search heuristics for a flowshop scheduling problem in a printed circuit board assembly process. *European Journal of Operational Research*, *91*, 124–143.

Lacomme, P., Huguet, MJ., & Tchernev, N. (2011). Dedicated constraint propagation for job-shop problem with generic time-lags. In *16th IEEE conference on emerging technologies and factory automation IEEE catalog number: CFP11ETF-USB*.

Lacomme, P., & Tchernev, N. (2012). Job-shop with generic time lags: A heuristic based approach. In *9th international conference of modeling, optimization and simulation—MOSIM*.

Lawrence, S. (1984). *Supplement to resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques*. Pittsburgh: Graduate School of Industrial Administration, Carnegie Mellon University.

Lin, J. (2016). A hybrid discrete biogeography-based optimization for the permutation flow shop scheduling problem. *International Journal of Production Research*, *54*(16), 1–10.

Lu, X., Du, Y., Yang, X., & Zheng, Y. (2018). A biogeography-based memetic algorithm for job-shop scheduling. In *International conference on bio-inspired computing: theories and applications* (pp. 273–284).

Ma, H., Su, S., Simon, D., & Fei, M. (2015). Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling. *Engineering Applications of Artificial Intelligence*, *44*, 79–90.

Ma, H., & Simon, D. (2017). Biogeography-based optimization: A 10-year review. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *1*(5), 391–407.

MacArthur, R., & Wilson, E. (1967). *The theory of biogeography*. Princeton, NJ: Princeton University Press.

Manier, M. A., & Bloch, C. (2003). A classification of hoist scheduling problems. *International Journal of Flexible Manufacturing Systems*, *15*(1), 37–55.

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

Mitten, L. G. (1958). Sequencing *n* jobs on two machines with arbitrary time lags. *Management Science*, *5*(3), 293–298.

Nawaz, M., Enscore, J. E. E., & Ham, I. (1983). A heuristic algorithm for the *n*-job, *m* machine sequencing problem. *Management Science*, *16/B*, 630–637.

Neumann, K., Schwindt, C., & Zimmermann, J. (2002). *Project scheduling with time windows and scarce resources*. Berlin: Springer.

Nikbakhsh, J. P., Mohammad, F. M., & Mohammad, K. (2012). An immune algorithm for hybrid flow shop scheduling problem with time lags and sequence dependent setup times. *International Journal of Advanced Manufacturing Technology*, *63*(1–4), 337–348.

Nowicki, E., & Smutnicki, C. (2005). An advanced Tabu search algorithm for the job shop problem. *Journal of Scheduling*, *8*, 145–159.

Rabiee, M., Jolai, F., Asefi, H., & Fattahi, P. (2016). A biogeography-based optimization algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimize mean tardiness. *International Journal of Computer Integrated Manufacturing*, *29*, 1007–1024.

Rifai, A. P., Nguyen, H., Aoyama, H., Dawal, S. Z., & Masruroh, N. A. (2018). Non-dominated sorting biogeography-based optimization for bi-objective reentrant flexible manufacturing system scheduling. *Applied Soft Computing*, *62*, 187–202.

Rajendran, C. (1994). A no-wait flow shop scheduling heuristic to minimize makespan. *The Journal of the Operational Research Society*, *45*, 472–478.

Roy, B., & Sussmann, B. (1964). *Les problèmes d'ordonnancement avec contraintes disjonctives*. SEMA: Technical report.

Sheikh, S. (2013). Multi-objective flexible flow lines with due window, time lag and job rejection. *International Journal of Advanced Manufacturing Technology*, *64*(9–12), 1423–1433.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*, 702–713.

Sinha, S., Bhola, A., Panchal, V., Singhal, S., & Abraham, A. (2012). Resolving mixed pixels by hybridization of biogeography based optimization and ant colony optimization. In *IIEEE Congress on Evolutionary Computation*, Brisbane, QLD (pp. 1–6).

Soukhal, A., Oulamara, A., & Martineau, P. (2005). Complexity of flow shop scheduling problems with transportation constraints. *European Journal of Operational Research*, *161*, 32–41.

Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. Hoboken, NJ: Wiley.

Wang, X., & Duan, H. (2014). A hybrid biogeography-based optimization algorithm for job shop scheduling problem. *Computers & Industrial Engineering*, *73*, 96–114.

Wee, L., Antoni, W., & Mohammad, D. (2016). A biogeography-based optimization algorithm hybridized with Tabu search for the quadratic assignment problem. *Computational Intelligence and Neuroscience*,. https://doi.org/10.1155/2016/5803893.

Wikum, E. D., Llewellyin, D. C., & Nemhauser, G. L. (1994). One-machine generalized precedence constrained scheduling problem. *Operations Research Letters*, *16*, 87–99.

Wismer, A. D. (1972). Solution of the flow shop scheduling problem with no intermediate queues. *Operations Research*, *20*, 689–697.

Wu, J., Wu, X., Lu, X., Lu, X., Du, Y., & Zhang, M. (2019). Water wave optimization for flow-shop scheduling (Vol. 771–783).

Yang, Y. (2015). A modified biogeography-based optimization for the flexible job shop scheduling problem. In *Mathematical problems in engineering*.

Ye, S., Zhao, N., Li, K., & Lei, C. (2017). Efficient heuristic for solving non-permutation flow-shop scheduling problems with maximal and minimal time lags. *Computers & Industrial Engineering*, *113*, 160–184.

Zhang, X. (2010). Scheduling with time lags. Ph.D. Thesis, Erasmus Research Institute of Management.

Zhao, F., Qin, S., & Zhang, Y. (2019). A hybrid biogeography-based optimization with variable neighborhood search mechanism for

no-wait flow shop scheduling problem. *Expert Systems with Applications*, *126*, 321–339.

Zhao, N., Ye, S., Li, K., & Chen, S. (2017). Effective iterated greedy algorithm for flow-shop scheduling problems with time lags. *Chinese Journal of Mechanical Engineering*, *30*, 652–662.

Zheng, Y., Ling, H., Shi, H., & Chen, H. (2014a). Emergency railway wagon scheduling by hybrid biogeography-based optimization. *Computers & Operations Research*, *43*(1), 1–8.

Zheng, Y., Ling, H., & Xue, J. (2014b). Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations. *Computers & Operations Research*, *50*, 115–127.