

**A Hybrid Discrete Bat Algorithm with Krill Herd based advanced planning and scheduling tool for the capital goods industry**

**Sirikarn Chansombat<sup>1</sup>, Ponnapa Musikapun<sup>2</sup>, Pupong Pongcharoen<sup>1,\*</sup>,  
and Christian Hicks<sup>3</sup>**

<sup>1</sup> Centre of Operations Research and Industrial Applications (CORIA),  
Faculty of Engineering, Naresuan University, Phitsanulok 65000, Thailand.

Email: sirikarnc53@email.nu.ac.th ORCID ID: 0000-0003-4873-0478

Email: pupongp@nu.ac.th and Scopus author ID: 17435722900

<sup>2</sup> School of Logistics and Supply Chain,

Naresuan University, Phitsanulok 65000, Thailand

Email: ponnepam@nu.ac.th and Scopus author ID: 56644715500

<sup>3</sup> Newcastle University Business School, 5 Barrack Road,

Newcastle upon Tyne, NE1 7RU, UK.

Email: chris.hicks@ncl.ac.uk and Scopus author ID: 7102667331

\* Corresponding author: pupongp@nu.ac.th

**Abstract**

Capital goods companies produce high value products such as power plant or ships, which have deep and complex product structures, with components having long process routings. Contracts usually include substantial penalties for late delivery. The high value of items can lead to substantial holding costs. Efficient schedules minimise earliness and tardiness costs and need to satisfy assembly and operation precedence constraints as well as finite capacity. This paper presents the first advanced planning and scheduling (APS) tool

for the capital goods industry that uses a Discrete Bat Algorithm (DBA), modified DBA (MDBA) and hybrid DBA with Krill Herd algorithm (HDBK) to optimise schedules. The tool was validated using four datasets obtained from a collaborating capital goods company. A sequential experimental strategy was adopted. The first experiment identified appropriate parameter settings for the DBA. The second experiment evaluated and compared the performance of the proposed HDBK algorithm with an Artificial Bee Colony, Krill Herd (KH), Modified KH, DBA and MDBA metaheuristics. The experimental results revealed that the HDBK performed best in terms of the minimum penalty cost for all problem sizes and achieved up to a 47.837% reduction in mean total penalty costs of extra-large problem size.

**Keywords:** Advanced planning and scheduling; Capital Goods; Bat Algorithm; Krill Herd; Artificial Bee Colony.

## 1 Introduction

Suppliers of capital goods are an important sector of the world economy that enhances the productivity and supports the diffusion of superior technologies (Fauceglia 2014). The main business activities of capital goods companies are the design, manufacture and construction of plant. Typical products include cranes, large steam turbines, offshore production facilities, oil platforms and ships. These products are important because they underpin manufacturing, services, trade and distribution (Acha et al. 2004).

Scheduling is “a decision-making process that plays an important role in most manufacturing and service industries” (Pinedo and Chao 1999, p.2). It can enhance the productivity of a production process (Gen and Lin 2014). Scheduling is one of the most popular research topics in the area of production and operations management (Chaudhry and Luo 2005). Production scheduling problems may be categorised as: single machine, parallel machines, flow shop, job shop, open shop and others (Pinedo and Chao 1999). Most production scheduling research has focused on single machine, parallel machines or flow shops (Lei 2009). Most of the production scheduling literature is theoretical and does not

model the many of the complexities experienced in practice (Fuchigami and Rangel 2017). There is a limited literature that has taken into account multiple-level assembly relationships (Na and Park 2014).

In the capital goods industry, production scheduling is a complex combinatorial optimisation (CO) problem. This is because there are a large number of components and subassemblies and the product structures are usually deep and complex. Major subassemblies require a range of components which are produced using a mix of jobbing, batch, assembly and flow processes. Many components require numerous machining operations which take place on many types of machine (Hicks 1998). Production scheduling must take into account operation and assembly precedence relationships and finite capacity (Hicks and Braiden 2000; Hicks 1998). Effective production schedules minimise production lead-time and meet customer due dates whilst satisfying resource constraints (Chen, Ji, and Wang 2011; Dayou, Pu, and Ji 2009). Production scheduling problems are non-deterministic polynomial (NP) hard combinatorial optimisation problems which means that the amount of computation required increases exponentially with problem size (Blum and Roli 2003).

Metaheuristics are particularly suitable for solving very large combinatorial problems, however, it is impossible to search the whole solution space, therefore an optimal solution cannot be guaranteed (Nagar, Haddock, and Heragu 1995). Metaheuristic algorithms may be classified in alternative ways (Talbi 2009; Yang 2010a). Single-point algorithms are trajectory methods that use local search heuristics e.g. Tabu Search (TS) (Glover 1990), Simulated Annealing (SA) (Kirkpatrick, Gelatt, and Vecchi 1983), Multi-start local search (MS), Greedy Randomised Adaptive Search Procedure (GRASP) and Iterated Local Search (ILS) (Lourenco, Martin, and Stützle 2003). They intensify search in the local region, which is also called exploitation oriented search (Gen and Cheng 1997). Population-based algorithms produce multiple solutions that explore the whole search space to produce greater

diversity. Well-known algorithms include Genetic Algorithms (GA) (Goldberg 1989), Ant Colony Optimisation (ACO) (Dorigo 1992) and Particle Swarm optimisation (PSO) (Kennedy and Eberhart 1995).

The literature reports the application of many established nature-inspired optimisation algorithms (see Table 1), which are broadly classified into five categories (Nanda and Panda 2014; Gupta and Sharma 2016; Fister Jr et al. (2013); Zambonelli and Viroli 2011): evolutionary-based, physics and chemistry based, swarm intelligence based, bio-inspired based and other algorithms.

Table 1. Classification of nature-inspired metaheuristic algorithms.

<b>Types</b>	<b>Algorithms</b>	<b>References</b>
Evolutionary based	Genetic Algorithm (GA)	Holland (1975)
	Genetic programming (GP)	Koza (1990)
	Evolutionary Strategy (ES)	Rechenberg (1965)
	Evolutionary Programming (EP)	Fogel, Owens, and Walsh (1966)
	Differential Evolution (DE)	Storn and Price (1997)
Physics and Chemistry based	Simulated Annealing (SA)	Kirkpatrick et al. (1983)
	Memetic Algorithm (MA)	Moscato and Norman (1992)
	Harmony Search (HS)	Geem et al. (2001)
Swarm Intelligence based	Shuffled Frog-Leaping Algorithm	Eusuff, Lansey, and Pasha (2006)
	Ant Colony Optimisation (ACO)	Dorigo (1992)
	Particle Swarm Optimisation (PSO)	Kennedy and Eberhart (1995)
	Artificial Bee Colony (ABC)	Karaboga (2005)
	Firefly Algorithm (FA)	Yang (2010a)
	Bat Algorithm (BA)	Yang (2010a)
	Krill Herd (KH)	Gandomi and Alavi (2012)
	Earthworm Optimisation Algorithm (EOA)	Wang, Deb, and Coelho (2016)
	Monarch Butterfly Optimisation (MBO)	Wang, Deb, and Cui (2015)
	Moth search algorithm (MSA)	Wang (2016)
	Cuckoo Search (CS)	Yang and Deb (2009)
	Chaotic Cuckoo Search (CCS)	Wang et al. (2016)
	Elephant Herding Optimisation (EHO)	Wang et al. (2016)
	Bio-inspired based	Flower Algorithm (FA)
Dolphin Echolocation Algorithm (DEA)		Lenin, Reddy, and Kalavathi (2014)
Japanese Tree Frogs Calling		Hernandez and Blum (2012)
Atmosphere Clouds Model		Yan, Hao, and Xie (2013)
Other algorithms	Backtracking Optimisation Search (BOS)	Civicioglu (2013b)
	League Championship Algorithm (LCA)	Kashan (2009)
	Social Emotional Optimisation (SEO)	Xu, Cui, and Zeng (2010)
	Artificial Cooperative Search (ACS)	Civicioglu (2013a)

Advanced planning and scheduling (APS) systems are based on optimisation and constraint-based planning algorithms that aim to meet customer requirements whilst

satisfying specified constraints (Hvolby and Steger-Jensen 2010). APS systems aim to manage the supply chain to improve customer satisfaction, increase efficiency and reduce costs (Dayou, Pu, and Ji 2009). APS systems have been based upon GA (Chen, Ji, and Wang 2011) and GA with local search (Pu et al. 2007), but there are no reports of the BA being used to for APS.

The objectives of this paper were to: (i) review Swarm Intelligence based Algorithms including, the Artificial Bee Colony (ABC), Krill Herd (KH) and Bat Algorithms (BA); (ii) explain a novel APS scheduling tool that meets the requirements of the capital goods industry that manufacture complex products with multi-level assemblies. The tool incorporates a novel Discrete Bat Algorithm (DBA), a Modified Discrete Bat Algorithm (MDBA) and a Hybrid Discrete Bat Algorithm with Krill Herd algorithm (HDBK) for optimisation; (iii) conduct a series of computational experiments that identified appropriate parameter settings for the DBA; iv) outline the development of the MDBA and the HDBK; and v) compare the performance of the proposed methods (DBA, MDBA and HDBK) with other approaches (ABC, KH and modified KH).

The next section explains the characteristics of population-based metaheuristics. Section 3 describes the development of advanced planning and scheduling (APS) tool. Section 4 presents the experimental design and analyses the results. The last section provides conclusion and suggestions for future research.

## **2 Population-based metaheuristics**

Population-based algorithms have been widely used to solve real world problems. They simultaneously consider multiple potential solutions and tend to perform better than single-point algorithms (Manda, Satapathy, and Poornasatyanarayana 2012). Prugel-Bennett (2010) identified five mechanisms that help give population-based algorithms an advantage:

(i) building blocks from different solutions are combined; (ii) the crossover operator focuses the search and can dramatically reduce the time taken to find a solution; (iii) the population acts as a low pass filter, which ignores local distractions; (iv) a population has the ability to search different parts of the search space simultaneously, which hedges against bad luck in the initial position; and (v) it is possible to identify parameter values that make an appropriate balance between exploitation and exploration.

Pongcharoen (2001) developed a comprehensive Genetic Algorithm (GA) tool for scheduling the production of capital goods using the objective function shown equation (1) that aggregates earliness and tardiness costs (Pongcharoen, Hicks, and Braiden 2004). This objective function was also used to solve the same problems using the Artificial Bee Colony (Pongcharoen et al. 2012) and the Krill Herd (Puongyeam, Pongcharoen, and Vitayasak 2014) algorithms.

$$\text{Total penalty cost} = \sum_{j=1}^C \sum_{k=1}^P Pe(E_{jk}) + \sum_{k=1}^P Pe(E_k) + \sum_{k=1}^P Pt(T_k) \quad (1)$$

Notation:

$j$	assembly or component $j$ ( $j = 1, 2, \dots, C_{max}$ )
$k$	final product $k$ ( $k = 1, 2, \dots, P_{max}$ )
$Pe$	earliness penalty rate (currency units per day)
$Pt$	tardiness penalty rate (currency units per day)
$E_k$	earliness of product $k$
$E_{jk}$	earliness of component $j$ in product $k$
$T_k$	tardiness of product $k$

## **2.1 Artificial Bee Colony (ABC) algorithm**

The Artificial Bee Colony (ABC) algorithm is a popular swarm intelligence-based algorithm developed by Karaboga (Karaboga 2005; Karaboga and Akay 2009; Karaboga et al. 2014). This approach is based on the collective foraging behaviour of a bee colony, which includes three categories of bee: employed bees, which forage for nectar; onlookers waiting in the hive; and scouts, which undertake random search. There is only one bee that visits each source, so the number of employed bees is the same as the number of food sources. Once a food source is identified (a candidate solution), the nectar (fitness) is identified and computed. The scouts share information with the onlooker bees. Onlooker bees choose their food source depending on the probability of the food occurring. If bees are unable to improve the fitness of the food source, their solutions are rejected (see Karaboga and Basturk 2007).

De Oliveira and Schirru (2011) developed an ABC for combinatorial optimisation that used random keys (Bean 1994) for mapping discrete variables to continuous variables. Cui and Gu (2012) developed a discrete ABC for hybrid flow shop scheduling that included a three-step differential evolution scheme (mutation, crossover and selection) for allocating employed bees to food sources. They used the algorithm developed by Nawaz, Ensore, and Ham (1983) in their procedure. Pansuwan, Rukwong, and Pongcharoen (2010) developed a scheduling tool for capital goods companies that used a discrete ABC together with the objective function shown in equation (1).

## **2.2 Krill Herd (KH) algorithm**

The Krill Herd (KH) algorithm (Gandomi and Alavi 2012) is a swarm intelligence algorithm which is based on the herding of the krill swarms. The time-dependent position of an individual krill is determined by three main actions: (i) movement induced other krill; (ii) foraging action; and (iii) random diffusion.

Initially, a swarm of krill are randomly generated in the search space. Krill try to maintain a high density and move according to their mutual effects (Gandomi and Alavi 2012). Each krill moves through  $n$  dimensional search space to look for a potential solution by moving towards the highest density of food. To improve the performance and convergence speed, crossover and mutation genetic operations were incorporated into the algorithm. The iterative search is ended when the termination criteria are met.

Wang, Deb, and Thampi (2015) developed a discrete krill herd method for flexible job shop scheduling. Puongyeam, Pongcharoen, and Vitayasak (2014) developed a discrete krill herd for scheduling in the capital goods industry which used the objective function shown in equation (1).

### **2.3 Bat Algorithm (BA)**

In 2010, Yang (2010b) presented a new metaheuristic algorithm, called the Bat Algorithm (BA) which is based on the echolocation capability of the micro-bats. In nature bats fly randomly in their search for prey with velocities  $v_i$  at positions  $x_i$  with varying wavelength/frequency ( $\lambda/f$ ), pulse rate  $r_i$  and loudness  $A_0$ . The position of each bat represents a possible solution. Depending on the proximity of the prey, bats can automatically adjust their wavelength/frequency and pulse emission rate  $r_i \in [0,1]$ . The loudness can vary from a minimum loudness ( $A_{min}$ ) to a maximum loudness ( $A_0$ ) with a typical range of [1,2]. Frequencies are normally in the range 25kHz to 150kHz (Yang 2010b).

Yang (2010b) outlined the BA as follows. The process starts by initialising a swarm (population) of  $n$  virtual bats, each of which has a random initial position (initial solution), where the ranges are problem specific, together with random values for pulse rate, loudness and frequency. Then, all of the bats move from their initial positions seeking a global best solution. Each individual bat randomly selects a frequency ( $f_i$ ) where  $f \in [0, f_{max}]$ , using

equation (2), where  $\beta \in [0,1]$  is a random number drawn from a uniform distribution. The velocity of each bat  $i$  is updated using equation (3), where  $t$  is the iteration number,  $t_{max}$  is the maximum number of iterations ( $0 \leq t \leq t_{max}$ ),  $v_i^{t-1}$  is the previous velocity,  $x_i^t$  is the current position of bat  $i$  in iteration  $t$ , and  $x^*$  is the position of the best-so-far bat. Equation (4) calculates the new position based upon the previous position and current velocity. For local search, once a solution is selected among the current best solutions, a new solution  $x_{new}$  for each bat is generated locally using random walk using equation (5), where  $x_{old} = x_i^t$ ,  $\varepsilon \in [-1,1]$  is a random number,  $A^t$  is the arithmetic mean loudness of all bats in the current iteration. Equation (6) updates the loudness  $A_i^{t+1}$  for each individual bat, where  $\alpha$  is the loudness coefficient, a parameter in the range  $[0,1]$ . The pulse emission rate  $r_i^{t+1}$  is updated by equation (7), where  $r_i^0$  is the initial pulse emission rate for the bat and  $\gamma$ , the pulse rate emission coefficient is a parameter that is greater than 0. This process is repeated until the maximum number of iterations  $t_{max}$  has been completed.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (2)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (3)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (4)$$

$$x_{new} = x_{old} + \varepsilon A^t \quad (5)$$

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (7)$$

The Bat Algorithm (BA) is a continuous optimisation algorithm, whereas combinatorial optimisation problems require discrete optimisation. There are two ways to apply the BA to discrete problems: i) use continuous optimisation, if it is possible to map the problem to a continuous variable; or ii) develop a discrete BA (Luo et al. 2014). Marichelvam and Prabaharam (2012) used the mapping approach to solve flow shop scheduling problems.

Luo et al. (2014) used the BA for solving permutation flow shop scheduling problems. Random keys (Bean 1994) were used to map from discrete to continuous variables. This was an approach that had previously been adopted by Tasgetiren et al. (2007) for use in Particle Swarm Optimisation. Dao, Pan, and Pan (2018) developed a parallel BA that used random key mapping for job shop scheduling.

## 2.4 A comparison of the proposed population-based metaheuristics

The concept, terminology and parameters of metaheuristics vary. Table 2 provides a summary of the population-based metaheuristics (ABC, KH and BA) presented in this work.

Table 2. Concept and terminology comparison of ABC, KH and BA.

Comparison	Artificial Bee Colony (ABC)	Krill Herd (KH)	Bat Algorithm (BA)
Natural inspiration	Foraging behaviour of a bee colony	Herding behaviour of krill colony	Echolocation behaviour of micro-bats
Solution initialisation	Random	Random	Random
Candidate solution	Food source individual's position	Krill individual's position	Bat individual's position
Old solution	Old food source position	Old krill position	Old bat position
New solution	New food source position	New krill position	New bat position
Best solution	Any food source with the best fitness	Any krill with the best fitness	Any bat with the best fitness
Fitness/objective	Nectar amount of the food source	Distance between krill individual and food and the densest location in the herd	Distance between bat individual and target
Size of candidates	Colony	Herd	Population
Iterative search	Number of cycles	Number of generations	Number of iterations
Process for generating new solution	The employed bee becomes a scout.	Motion induced by krill herd, foraging activity and physical diffusion	Adjusting frequency, update velocity and position
Intensification	Neighbourhood search carried by employed and onlooker bees	Foraging motion	Random walk
Diversification	Random search of scout bees	Random diffusion	Flying randomly
Parameters	- Combination of the population size and the number of maximum cycles ( $nMCN$ ) - Limit factor which is a predefined value that limits the number of times that a food source can be moved without producing an improvement before it is abandoned. (percentage of the maximum number of cycles)	- Combination of population size and the number of max generations ( $nI_{max}$ ) - Inertia weight of motion induced ( $\omega_n$ ) - Inertia weight of the foraging motion ( $\omega_0$ ) - The maximum diffusion speed ( $D_{max}$ ) - The crossover operation ( $COP$ ) - The mutation rate ( $M_R$ )	- Combination of population size and the number of maximum iterations ( $nI_{max}$ ) - Pulse rate emission coefficient ( $\gamma$ ) - Loudness coefficient ( $\alpha$ )

Due to the different inspirations adopted within the metaheuristics, the unique mechanisms embedded in the metaheuristics have their own properties to avoid iterative search becoming trapped in local optima whilst performing the search in a more intelligently than random search. The advantages and disadvantages of the proposed of the classical algorithms, including ABC, KH, and BA are summarised in Table 3.

Table 3. Advantages and disadvantages of ABC, KH and BA.

Methods	Advantages	Disadvantages
ABC	<ul style="list-style-type: none"> <li>- Not sensitive to initial parameter values (Bansal, Sharma, and Jadon 2013).</li> <li>- Not affected by the number of dimensions of the problem (Bansal, Sharma, and Jadon 2013).</li> <li>- Can avoid local minimum (Karaboga and Basturk 2007).</li> <li>- Efficient for multivariable, multimodal function optimisation (Karaboga and Basturk 2007).</li> <li>- Good exploration (Khorsandi, Hosseinian, and Ghazanfari 2013; Gao and Liu 2012).</li> <li>- Quick convergence (Cui and Gu 2015).</li> <li>- Few control parameters (Cui and Gu 2015; Luo, Wang, and Xiao 2013).</li> </ul>	<ul style="list-style-type: none"> <li>- Premature convergence (Bansal, Sharma, and Jadon 2013).</li> <li>- Long execution times because of its stochastic nature (Kang, Li, and Li 2013).</li> <li>- Poor exploitation (Khorsandi, Hosseinian, and Ghazanfari 2013; Gao and Liu 2012).</li> <li>- Slow to converge (Luo, Wang, and Xiao 2013).</li> <li>- Can easily fall into the local optimum (Luo, Wang, and Xiao 2013).</li> <li>- Hard to find the best solution from all feasible solutions (Luo, Wang, and Xiao 2013).</li> </ul>
KH	<ul style="list-style-type: none"> <li>- Powerful exploration (Wang et al. 2013).</li> <li>- Unnecessary derivative information (Wang et al. 2013; Wang, Guo, et al. 2012).</li> <li>- Each agent can contribute to the search process according to its fitness.(Gandomi and Alavi 2012).</li> <li>- Each neighbour has an attractive/repulsive effect on the movement of the krill individual. (Gandomi and Alavi 2012).</li> <li>- Very few control variables (Mukherjee and Mukherjee 2016; Wang et al. 2013).</li> <li>- Good balance between global and local search (Agrawal, Pandit, and Dubey 2016).</li> <li>- Few parameters to regulate (Wang, Hossein Gandomi, and Hossein Alavi 2013).</li> <li>- Able to shrink the search region towards the promising area within a few generations (Wang et al. 2013).</li> </ul>	<ul style="list-style-type: none"> <li>- Easy to fall into the local optimum (Gandomi and Alavi 2012; Wang et al. 2013).</li> <li>- No guarantee of fast convergence (Wang et al. 2013).</li> <li>- Poor exploitation (Wang et al. 2013).</li> </ul>
BA	<ul style="list-style-type: none"> <li>- Powerful exploitation (Yilmaz and Kucuksille 2013; Dos Santos Coelho and Askarzadeh 2016).</li> <li>- Parameter control (automatically switching from exploration to exploitation) (Kaur and Chhabra 2016; Yang 2013).</li> <li>- Frequency tuning (Kaur and Chhabra 2016; Yang 2013).</li> <li>- Automatic zooming (Yang 2013).</li> <li>- Quick convergence at the initial stage by switching from exploration to exploitation (Yang 2013).</li> <li>- Balance between exploration and exploitation (Chua et al. 2015).</li> </ul>	<ul style="list-style-type: none"> <li>- Can easily to fall into the local optimum (Li and Zhou 2014; Pravesjit 2016).</li> <li>- Premature convergence (Ahmadi and Nikravesh 2016).</li> <li>- May be trapped in local optima (Dos Santos Coelho and Askarzadeh 2016).</li> <li>- May lead to stagnation after the initial stage (Yang 2013).</li> <li>- Obtains poor results when dealing with high-dimensional problems (Fister Jr, Fister, and Yang 2013).</li> </ul>

Table 4 summarises a comprehensive literature review of previous research that has used the proposed population-based metaheuristics (including ABC, KH and BA) for solving production scheduling problems. The hybridisation of the KH algorithm with other metaheuristics for solving the production scheduling problem is a gap in the literature.

Table 4. Applications of metaheuristics to solve production scheduling problems.

<b>Algorithms</b>	<b>Single metaheuristics</b>	<b>Hybridisation</b>
<b>BA</b>	<p><b>2012:</b> Musikapun and Pongcharoen (2012); Marichelvam and Prabaharam (2012)</p> <p><b>2013:</b> Chansombat et al. (2013); Marichelvam et al. (2013); Xie, Zhou, and Tang (2013)</p> <p><b>2014:</b> Luo et al. (2014)</p> <p><b>2016:</b> Kongkaew (2016)</p> <p><b>2017:</b> Xu, Bao, and Zhang (2017); Zaher, Ragaa, and Sayed (2017)</p> <p><b>2018:</b> Dao, Pan, and Pan (2018)</p>	<p><b>2016:</b> Tosun and Marichelvam (2016)</p> <p><b>2017:</b> Pei et al. (2017)</p>
<b>ABC</b>	<p><b>2010:</b> Pansuwan, Rukwong, and Pongcharoen (2010); Tasgetiren et al. (2010)</p> <p><b>2011:</b> Li, Pan, and Gao (2011); Pan et al. (2011); Tasgetiren et al. (2011)</p> <p><b>2012:</b> Deng, Xu, and Gu (2012); Pongcharoen et al. (2012); Banharnsakun, Sirinaovakul, and Achalakul (2012); Cui and Gu (2012); Sang, Gao, and Pan (2012); Li and Yin (2012); Sundar and Singh (2012); Wang, Zhou, Xu, and Liu (2012); Wang, Zhou, Xu, Wang, et al. (2012)</p> <p><b>2013:</b> Han et al. (2013); Lei (2013); Pan et al. (2013); Tasgetiren et al. (2013); Wang, Xie, and Cheng (2013)</p> <p><b>2014:</b> Vijaychakaravathy, Marimuthu, and Sait (2014); Kizilay et al. (2014); Li, Pan, and Tasgetiren (2014); Muthiah and Rajkumar (2014); Pan et al. (2014)</p> <p><b>2015:</b> Ribas, Companys, and Tort-Martorell (2015); Al-Salamah (2015); Caniyilmaz, Benli, and Ilkay (2015); Cui and Gu (2015); Gao et al. (2015)</p> <p><b>2016:</b> Asadzadeh (2016); Gao et al. (2016)</p> <p><b>2017:</b> Zhang et al. (2017); Li (2017); Pan et al. (2017)</p>	<p><b>2011:</b> Li et al. (2011)</p> <p><b>2013:</b> Liu and Liu (2013); Han et al. (2013); Lin, Ying, and Huang (2013); Thammano and Phu-Ang (2013); Zhang, Song, and Wu (2013)</p> <p><b>2014:</b> Lin and Ying (2014); Selvi, Uma Rani, and Sankar (2014)</p> <p><b>2015:</b> Li and Pan (2015); Nasiri (2015)</p> <p><b>2016:</b> Yue et al. (2016)</p> <p><b>2017:</b> Li et al. (2017); Muthulakshmi and Somasundaram (2017); Sundar et al. (2017); Wang et al. (2017)</p>
<b>KH</b>	<p><b>2014:</b> Puongyeam, Pongcharoen, and Vitayasak (2014)</p> <p><b>2015:</b> Wang, Deb, and Thampai (2015)</p>	No reported research

### 3 The development of Advanced Production and Scheduling (APS) tool

The APS tool was developed for solving production scheduling problems in the capital goods company using a Discrete Bat Algorithm (DBA), a Modified DBA (MDBA), and a Hybrid Discrete Bat Algorithm with Krill Herd (HDBK) algorithm. The objective was

to find an optimal schedule which minimised the total cost of earliness and tardiness penalties (equation 1). The tool was coded in a modular style using the C sharp programming language. The APS tool starts by obtaining input data. The input data comprises: (a) order information - due dates, the number of products for the penalty cost coefficients for earliness and tardiness; (b) product information - product structure (including all assemblies, subassemblies and components); (c) operational information - process routings, set-up, machining and transfer times (for all assemblies, subassemblies and components); (d) resource information - list of machines and their availability; (e) the DBA's parameters - the size of the population ( $n$ ), the number of iterations ( $I$ ), the pulse rate emission coefficient ( $\gamma$ ), the loudness coefficient ( $\alpha$ ) and the repositioning operation either using the swapping operator (Wang et al. 2003) or the adjustment operator (Wang et al. 2005).

A flowchart representing the proposed DBA, MDBA and HDBK used in the APS tool is shown in Figure 1, which includes:

- i) The main menu of the APS tool is displayed when the mouse is double clicked. The problem dataset can be selected and uploaded into the APS tool. All operations are encoded into alphanumeric strings that represent sequences of operations. These are analogous to the discrete vector position of a bat with the number of dimensions equal to the total number of operations in the schedule;
- ii) The graphical user interface (GUI) allows users to define parameters ( $pop$ ,  $iteration$ ,  $\gamma$ ,  $\alpha$ , and repositioning operator), scheduling characteristics ( $Pe$ ,  $Pt$ , and working hours per day), and the random seed number (if needed);

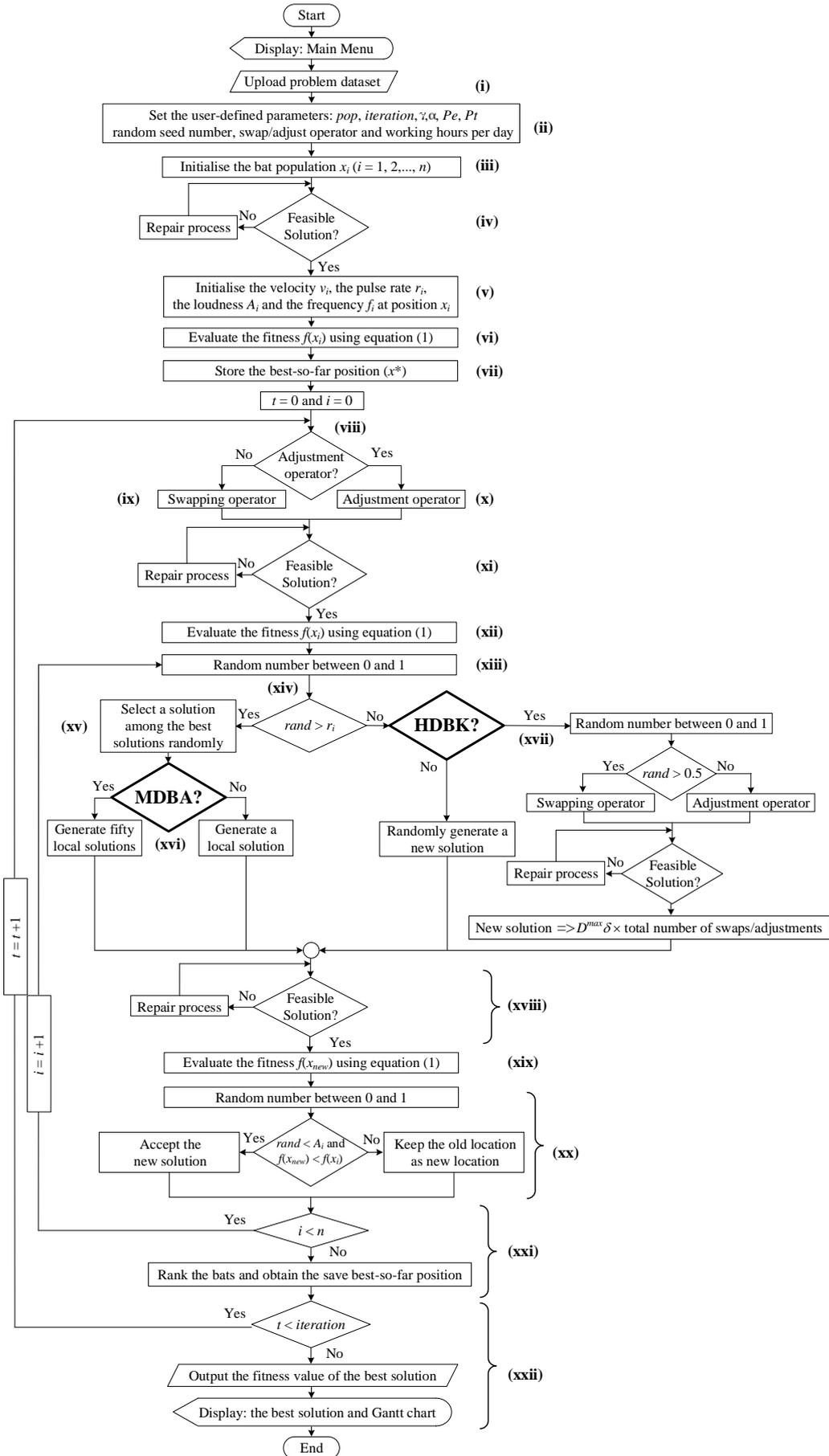
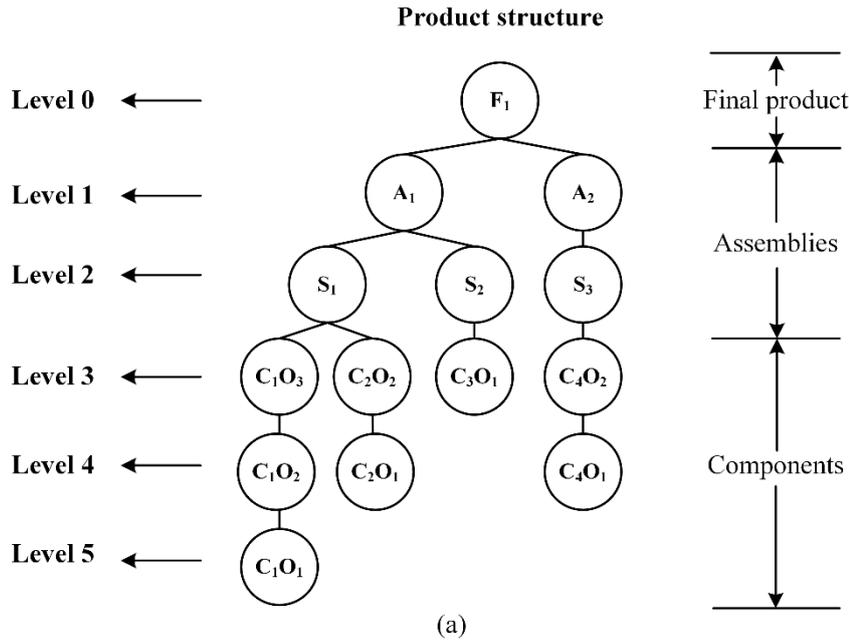


Figure 1. Flowchart of the DBA, MDBA and HDBK APS tool.

- iii) A swarm of bats is randomly generated. The product structure representation is illustrated in Figure 2 (a) using a simple example. The root node represents the final product ( $F_1$ ), which comprises assemblies ( $A_1$  and  $A_2$ ); subassemblies ( $S_1$ ,  $S_2$ , and  $S_3$ ); and components ( $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$ ) as the leaf nodes. All the nodes in the product structure will have a sequence of machining operations  $O_1, O_2 \dots O_n$ , which need to be completed sequentially. If the component  $C_1$  has three operations  $O_1, O_2$ , and  $O_3$ ,  $C_1$  can be represented as three intermediate items  $C_1O_1, C_1O_2$ , and  $C_1O_3$  where  $C_1O_3$  is the completed  $C_1$ , since it has three operations. Each bat represents a candidate solution (see Figure 2 (b));
- iv) Candidate solutions may be infeasible because they contravene assembly or operation precedence constraints. A repair process (Pongcharoen, Hicks, and Braiden 2004) was adopted to change routings and/or assembly sequences (the position vectors) to ensure that all precedence constraints are satisfied. The repair process also takes into account timing, finite capacity and deadlock. Figure 3 illustrates the adjustment of an infeasible schedule (repair process). In bat 2, the intermediate item  $C_1O_3$  is sequenced to take place before the intermediate item  $S_1$ . Therefore the algorithm swaps these operations so that they are in the correct sequence;
- v) Initially, each bat is randomly assigned the velocity  $v_i$ , the pulse rate  $r_i$  in the range  $[0,1]$ , the frequency  $f_i$  in the range  $[0,1]$ , and the loudness  $A_i$  in the range  $[1,2]$ , settings adopted (Chansombat et al. 2013);
- vi) The total penalty costs for all of the individuals within the initial population are calculated using equation (1);
- vii) The best-so-far position  $x^*$  leading to the lowest penalty cost is identified;



Bats	Position	Representation of operation sequences (Position of bat $x_i^t$ )													
1	$x_1^0$	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>3</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	S <sub>1</sub>	S <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	F <sub>1</sub>
2	$x_2^0$	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>2</sub>	S <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>3</sub> O <sub>1</sub>	S <sub>2</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	F <sub>1</sub>
⋮	⋮	⋮													
Pop	$x_{pop}^0$	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>1</sub> O <sub>3</sub>	S <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>3</sub> O <sub>1</sub>	S <sub>2</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	F <sub>1</sub>

(b)

Figure 2. Representation of a population of candidate solutions.

Bats	Position	Representation of operation sequences (Position of bat $x_i^t$ )														
1	$x_1^0$	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>3</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	S <sub>1</sub>	S <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	F <sub>1</sub>	✓
2	$x_2^0$	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>2</sub>	S <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>3</sub> O <sub>1</sub>	S <sub>2</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	F <sub>1</sub>	✗
		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
		C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>1</sub> O <sub>3</sub>	S <sub>1</sub>	C <sub>3</sub> O <sub>1</sub>	S <sub>2</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	F <sub>1</sub>	✓
⋮	⋮	⋮														
Pop	$x_{pop}^0$	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>1</sub> O <sub>3</sub>	S <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>3</sub> O <sub>1</sub>	S <sub>2</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	F <sub>1</sub>	✓

Figure 3. Check and reorder the part and operations precedence (repair process).

viii) All bats move from their current location  $x_i^t$  to a new location  $x_i^{t+1}$ . Instead of applying equation (3), which would apply for continuous optimisation, the discrete algorithm is based upon either the swapping operator (Wang et al. 2003) or the adjustment operator (Wang et al. 2005) which are described in steps ix) and x) below;

- ix) Swapping operator - the first stage is to calculate how many swaps would be required to map  $x_i^{t+1}$  to  $x^*$  (the best-so-far solution) using the six steps illustrated in Figure 4(a). The first step compares the elements of  $x^*$  from left to right with  $x_i^{t+1}$ . When a difference is detected (in this case  $C_2O_2 / C_3O_1$ ), the second step swaps the current element in  $x_i^{t+1}$  with the element containing the same value as  $x^*$  ( $C_2O_2$ ) and step 3 does the reverse swap to produce  $x_i^{t+1'}$  (where the number of dashes “'” indicates the number of swaps). This process then continues until the number of swaps required to map  $x_i^{t+1}$  to  $x^*$  is determined. In this case, steps 4,5 and 6 complete the process as  $x_i^{t+1''}$  is the same as  $x^*$  so the total number of swaps is 2; The second stage is to multiply this number of swaps by the random number  $f_i$  determined by step (v) above. The value is then rounded up to determine the actual number of swaps to be performed on  $x_i^{t+1}$  to determine its new position. This process is illustrated in Figure 4(b) that shows a situation where two swaps were required to transform,  $x_i^{t+1}$  to  $x^*$ , the  $f$  value was 0.1, giving 0.2 swaps, which would round up to one swap, so  $x_i^{t+1}$  is changed to  $x_i^{t+1'}$ ;
- x) Adjustment operator - the first step is to calculate how many adjustments would be required to map  $x_i^{t+1}$  to  $x^*$ . This is illustrated in Figure 5. The procedure compares the elements of  $x^*$  with  $x_i^{t+1}$ . When a difference is detected the current element in  $x^*$  is inserted into  $x_i^{t+1}$ , the duplicate value in  $x_i^{t+1}$  is deleted as shown. The remaining values to the right of the insertion point are moved one position to the right. This process is then continued until the number of adjustments required to map  $x_i^{t+1}$  to  $x^*$  is determined. Again, this value is multiplied by  $f$  and rounded up to determine the number of adjustments to be made on  $x_i^{t+1}$  to determine its new position;

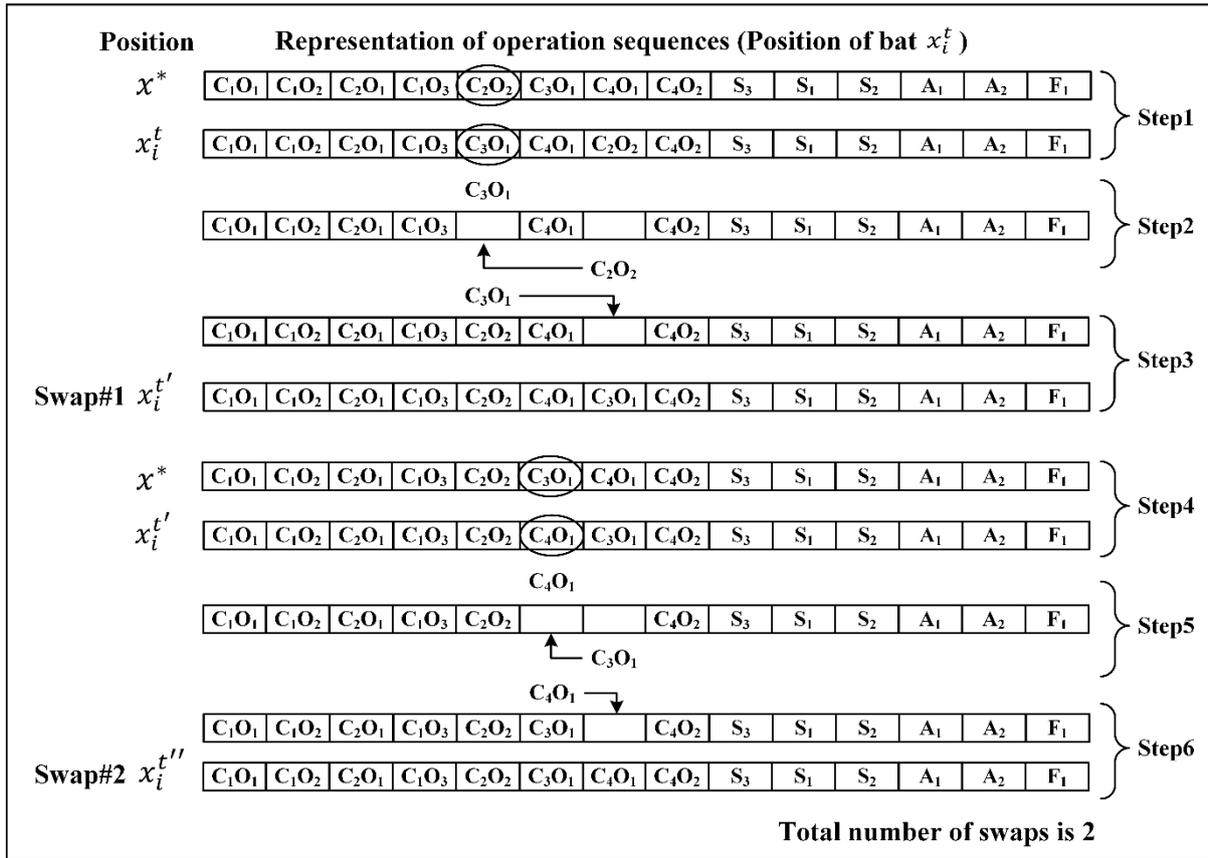


Figure 4(a). Swapping operator (Amara, Hamdani, and Alimi 2015).

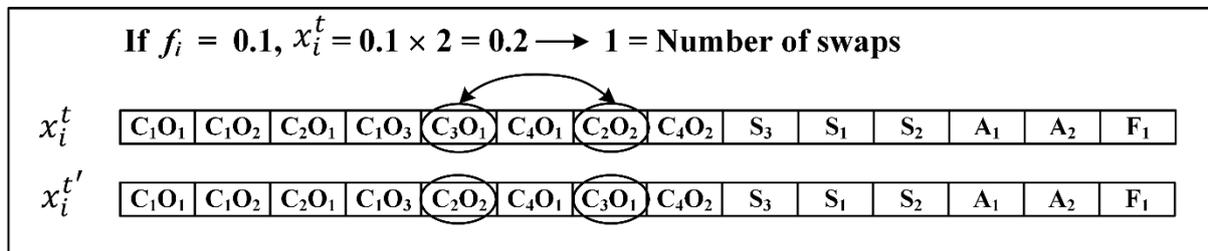


Figure 4(b). Swapping procedure.

- xi) The new positions of the bats are checked and repaired as necessary;
- xii) The total penalty costs for all of the individuals within the initial population is calculated and the best-so-far value is identified;
- xiii) A random number (*rand*) in the range 0-1 is generated;
- xiv) *rand* is compared with the pulse rate ( $r_i$ );
- xv) If  $rand > r_i$ , the best-so-far solution  $x^*$  is taken as the start point for a local search.

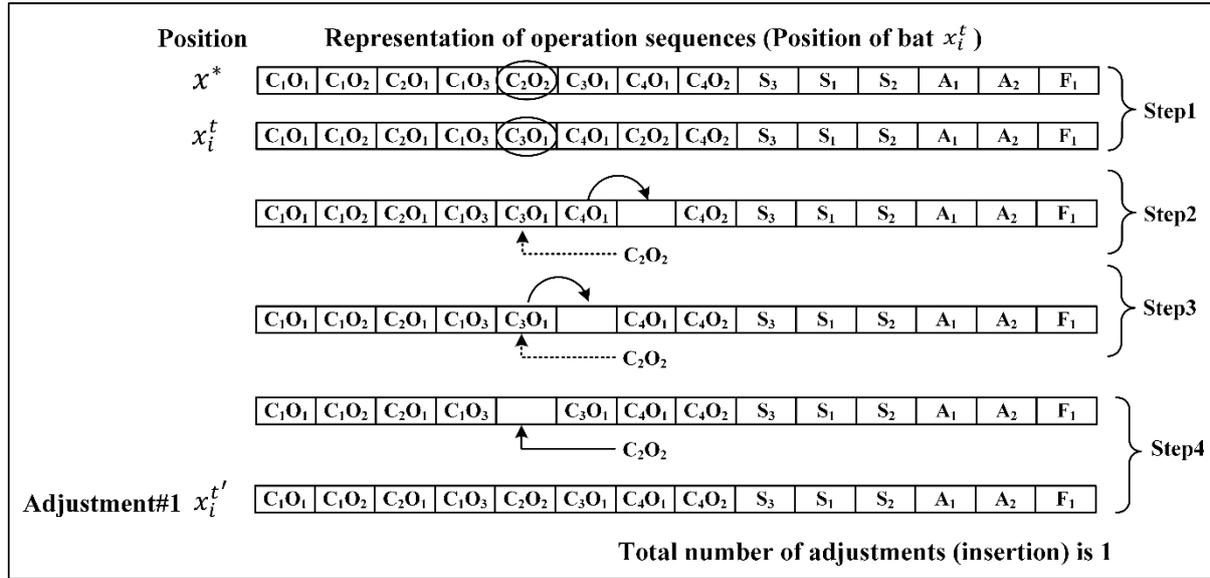


Figure 5. Adjustment operator.

- xvi) If the MDBA is selected, this step is repeated 50 times to improve the exploitation capability, otherwise just once;
- xvii) To improve the exploration capability of the BA, the random diffusion of the KH was incorporated into the conventional BA. This is illustrated in Figure 6.
- xviii) A new solution is repaired if necessary;
- xix) The fitness of new solution is evaluated;
- xx) A new *rand* in the range 0-1 is generated. If  $rand < A_i$  and if the penalty cost of  $x_i < x^*$  then  $x_i$  becomes  $x^*$ ,  $A_i$  will decrease using equation (6) and  $r_i$  will increase following equation (7). Otherwise,  $x^*$  remains unchanged;
- xxi) If the specified number of bats has not been completed, the procedure returns to step xiii). Otherwise, all bats are ranked and the best-so-far position is saved;
- xxii) If the required number of iterations has not been completed the procedure returns to step (viii), otherwise the program terminates and reports the best-so-far solution and displays it graphically as a Gantt chart.

If  $rand = 0.67 \rightarrow$  Swapping operator

Position	Representation of operation sequences (Position of bat $x_i^t$ )													
$x^*$	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>3</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	S <sub>1</sub>	S <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	F <sub>1</sub>
$x_i^t$	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>3</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	S <sub>1</sub>	S <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	F <sub>1</sub>
Swap#1 $x_i^{t'}$	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>3</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	S <sub>1</sub>	S <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	F <sub>1</sub>
Swap#2 $x_i^{t''}$	C <sub>1</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>2</sub>	C <sub>2</sub> O <sub>1</sub>	C <sub>1</sub> O <sub>3</sub>	C <sub>2</sub> O <sub>2</sub>	C <sub>3</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>1</sub>	C <sub>4</sub> O <sub>2</sub>	S <sub>3</sub>	S <sub>1</sub>	S <sub>2</sub>	A <sub>1</sub>	A <sub>2</sub>	F <sub>1</sub>

**Total number of swaps is 2**

From New solution  $\Rightarrow D^{max} \delta \times$  Total number of swaps/adjustments

If  $D^{max} = 0.1$ ,  $\delta = 0.75$ ,  $x_i^t = 0.1 \times 0.75 \times 2 = 0.15 \rightarrow 1 =$  Number of swaps

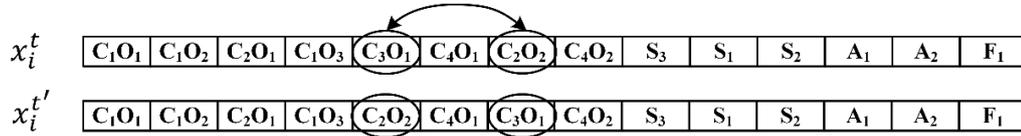


Figure 6. HDBK procedure.

## 4 Computational experiments

The computational experiments used data representing an 18 months schedule from a collaborating capital goods company. The first experiment identified the best DBA parameter settings. The second experiment evaluated the performance of the proposed HDBK and compared the performance with the MDBA, DBA, ABC, KH and MKH algorithms. Both experiments used the same datasets. The APS tool was experimented on a personal computer with a Core I7, 3.50 GHz CPU and 6 GB RAM.

### 4.1 Datasets

Pongcharoen et al. (2002) developed Genetic Algorithms for scheduling the production of capital goods and considered three problems (small, medium and large). Chainual, Lutuksin, and Pongcharoen (2007) developed an Ant Colony scheduling tool using the same problems. Xie, Hicks, and Pongcharoen (2010) additionally considered an extra-

large dataset that represented a complete schedule for a major product. These four datasets were used to test: the Artificial Bee Colony (Pansuwan, Rukwong, and Pongcharoen 2010; Pongcharoen et al. 2012) and Krill Herd (Puongyeam, Pongcharoen, and Vitayasak 2014). These algorithms can be directly compared because they were applied to common datasets with the same objective function outlined in equation (1). The characteristics of the problems considered are shown in Table 5.

Table 5. The characteristics of the four problems.

Problem sizes	No. of products (part number)	No. of items	Machining/assembly operations	No. of machines	Levels of product structure
Small	2 (245, 451)	15	25/9	8	11
Medium	2 (229, 451)	18	57/10	7	17
Large	2 (4, 228)	29	118/17	17	19
Extra-large	1 (227)	85	229/39	25	20

## 4.2 Identifying appropriate parameter settings

It is important to select metaheuristic parameters that obtain optimal results. A design of experiments strategy is much more effective and efficient than a trial-and-error approach. Factorial designs may be necessary to avoid misleading conclusions when interactions are present. They allow the effects of a factor to be estimated at several levels of the other factors producing results that are valid over a wide range (Montgomery 2012). Previous research on production scheduling in the capital goods has used this approach to identify appropriate parameter settings (Pongcharoen 2001; Pansuwan, Rukwong, and Pongcharoen 2010; Puongyeam, Pongcharoen, and Vitayasak 2014).

This experiment used the full factorial design to identify the appropriate parameter settings for the DBA. The factors included: (i) the combination of population size and the number of iterations ( $nI$ ), which determines the amount of search. In the experiments the value was fixed at 2,500 to ensure comparability with previous research; (ii) the pulse rate

emission coefficient ( $\gamma$ ); (iii) the loudness coefficient ( $\alpha$ ); and (iv) the repositioning operator (the swapping operation (SO) (Wang et al. 2003) or the adjustment operator (AO) (Wang et al. 2005)). The experiment was replicated ten times with different random number seeds. The number of runs for each replicate was  $3^3 \times 2 = 54$ , giving a total of 540 runs. For each run the best-so-far penalty cost was the dependent variable. The results were analysed using a general linear model form of analysis of variance (ANOVA). The main effects and first level interactions were considered in accordance with the sparsity of effects principle that states that a system is usually dominated by main effects and low level interactions (Montgomery 2012). Table 6 shows the ANOVA table, which shows the source of variation (*Source*), degrees of freedom (*DF*), sum of squares (*SS*), mean square (*MS*), *F* value, and *P* value. The factors with a *P* value of  $\leq 0.05$  were statistically significant with a 95% confidence interval. All the DBA parameters were considered as the main sources of variation as well as the interaction effects.

From Table 6, it can be seen that all of the DBA's parameters except  $\gamma$  were statistically significant. The only two-way interaction that was statistically significant was  $\alpha$ \**Repositioning operation*. The best parameter settings for the DBA were determined by considering the lowest mean best-so-far total cost obtained from main effect and interaction plots. Figure 7 shows the best combination for the interactions which were: (a)  $\alpha = 0.9$  with AO; (b)  $nI = 100*25$  and  $\alpha = 0.9$ ; (c)  $nI = 100*25$  and AO; and (d)  $\gamma = 0.1$  and AO.

### **4.3 Performance comparison of the proposed algorithms with other approaches**

The performance of the proposed algorithms (DBA, MDBA and HDBK) were compared against the ABC algorithm (Pansuwan, Rukwong, and Pongcharoen 2010), the KH and MKH algorithms (Puongyeam, Pongcharoen, and Vitayasak 2014). In each case the appropriate parameter settings had been identified through a design of experiments approach.

Each experiment was replicated 30 times to be consistent with Pansuwan, Rukwong, and Pongcharoen (2010) and Puongyeam, Pongcharoen, and Vitayasak (2014).

Table 6. ANOVA analysis of DBA parameters.

Source	DF	SS	MS	F value	P value
$nI$	2	5,637,225,926	2,818,612,963	42.160	<b>0.000</b>
$\gamma$	2	160,270,370	80,135,185	1.200	0.302
$\alpha$	2	29,966,267,593	14,983,133,796	224.110	<b>0.000</b>
Repositioning operation	1	4,547,201,852	4,547,201,852	68.010	<b>0.000</b>
$nI*\gamma$	4	142,679,630	35,669,907	0.530	0.711
$nI*\alpha$	4	301,440,741	75,360,185	1.130	0.343
$nI*$ Repositioning operation	2	68,137,037	34,068,519	0.510	0.601
$\gamma*\alpha$	4	53,362,963	13,340,741	0.200	0.939
$\gamma*$ Repositioning operation	2	17,403,704	8,701,852	0.130	0.878
$\alpha*$ Repositioning operation	2	1,290,334,259	645,167,130	9.650	<b>0.000</b>
Error	514	34,364,118,519	66,856,262		
Total	539				

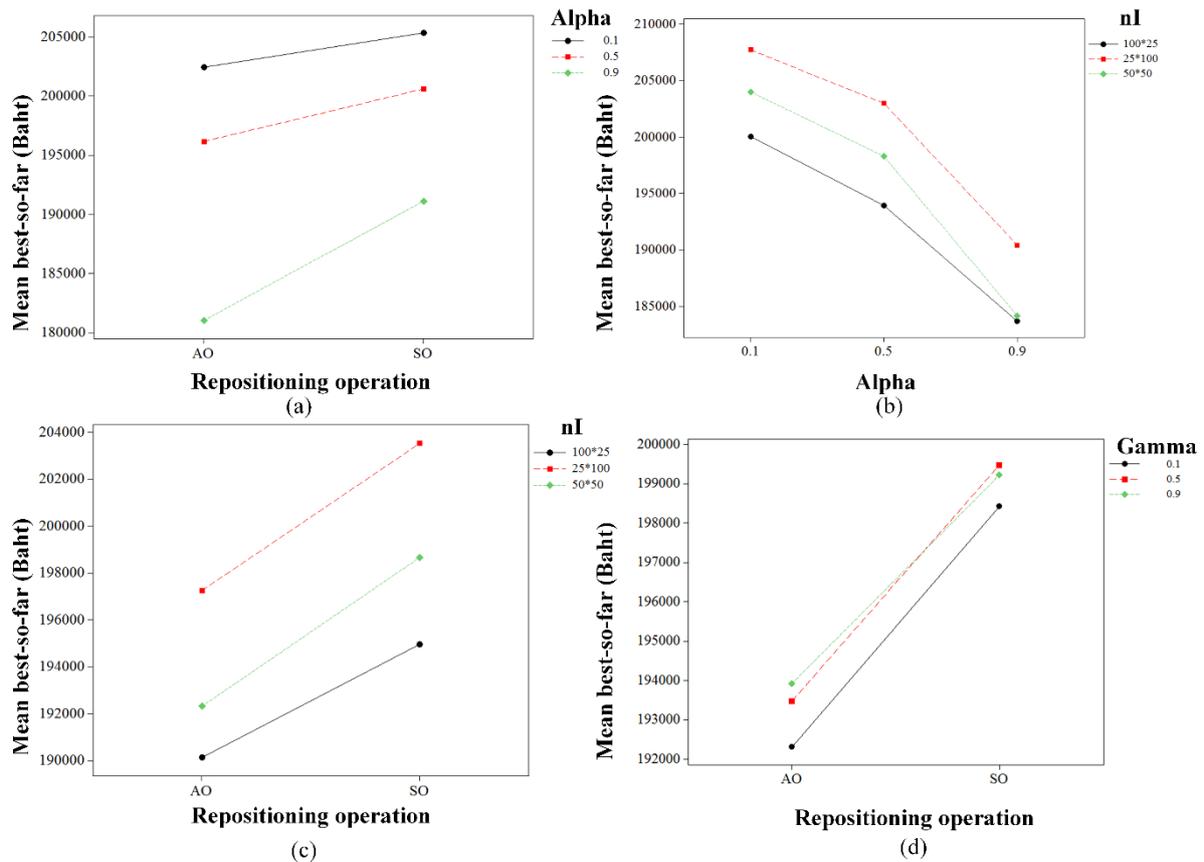


Figure 7. Interaction plots of (a)  $\alpha$  \* Repositioning operation, (b)  $nI*\alpha$ , (c)  $nI*$ Repositioning operation and (d)  $\gamma*$ Repositioning operation.

Table 7 shows that the performance of the HDBK, MDBA, DBA, ABC, KH and MKH in term of minimum (*Min*), maximum (*Max*), and arithmetic mean best-so-far penalty value (*Mean*) and standard deviation (*SD*). The dependent variable in this analysis was the best-so-far result achieved by each replicate. In terms of minimum total penalty cost, the HDBK outperformed the MDBA, DBA, ABC, KH and MKH for all problem sizes except the small problem.

Table 7. Performance comparison (penalty cost in currency units).

Problems	Total Penalty Cost	Methods					
		HDBK	MDBA	DBA	ABC	KH	MKH
Small	<i>Min</i>	15,000	15,000	15,000	15,000	16,500	15,000
	<i>Max</i>	15,000	15,000	19,000	15,500	19,500	15,500
	<i>Mean</i>	15,000	15,000	15,133	15,033	18,750	15,217
	<i>SD</i>	0	0	730	127	728	252
Medium	<i>Min</i>	31,000	31,500	32,500	52,500	58,500	55,500
	<i>Max</i>	57,000	54,500	57,000	57,500	60,500	59,000
	<i>Mean</i>	36,933	36,283	39,117	55,133	59,800	57,600
	<i>SD</i>	6,276	5,474	6,867	1,293	581	1,029
Large	<i>Min</i>	163,000	165,000	165,000	234,500	282,000	244,000
	<i>Max</i>	204,000	189,500	201,000	277,000	321,000	307,000
	<i>Mean</i>	181,767	178,033	182,733	258,417	304,367	291,883
	<i>SD</i>	9,081	6,608	7,681	11,503	9,872	12,663
Extra-large	<i>Min</i>	5,251,500	5,664,000	5,664,000	9,089,500	11,822,000	9,906,500
	<i>Max</i>	7,434,500	7,361,500	8,031,500	10,608,500	12,337,000	12,192,500
	<i>Mean</i>	6,350,617	6,572,750	6,890,500	9,860,517	12,174,600	11,412,867
	<i>SD</i>	621,625	426,103	585,670	417,689	162,237	631,384

A student *t* test established whether the mean differences were statistically significant. Table 8 shows the *T* value obtained by the *t*-test method, the *P* value, and the percentage improvement (*%Imp*) achieved by the algorithms. Almost all of the comparisons between the results obtained from the HDBK and the other approaches for extra-large problem were statistically significant with a 95% confidence interval (*P* value  $\leq 0.05$ ) except the modified DBA. For medium and large problems, the statistical comparisons indicated that the results obtained from the HDBK were significantly better than the results obtained from the ABC, KH and MKH. For small problems, the results obtained from the HDBK were significantly better than the results obtained from the KH and MKH. The performance of HDBK achieved

the highest percentage improvement (%*Imp*) of 47.837% when compared with KH followed by 44.356% when compared with MKH and 35.595% when compared with ABC.

Table 8. Statistical analysis using a *t*-test.

Methods	Statistical analysis	Problems			
		Small	Medium	Large	Extra-large
HDBK versus MDBA	<i>T value</i>	*	0.97	2.87	-1.64
	<i>P value</i>	*	0.338	0.008	0.111
	<i>% Imp</i>	0	-1.791	-2.097	<b>3.380</b>
HDBK versus DBA	<i>T value</i>	-1.00	-2.21	-0.96	-3.69
	<i>P value</i>	0.326	0.035	0.344	0.001
	<i>% Imp</i>	0.879	5.582	0.529	<b>7.835</b>
HDBK versus ABC	<i>T value</i>	-1.44	-15.87	-30.48	-30.78
	<i>P value</i>	0.161	0.000	0.000	0.000
	<i>% Imp</i>	0.220	33.011	29.661	<b>35.595</b>
HDBK versus KH	<i>T value</i>	-28.21	-19.59	-46.26	-47.98
	<i>P value</i>	0.000	0.000	0.000	0.000
	<i>% Imp</i>	20.00	38.239	40.280	<b>47.837</b>
HDBK versus MKH	<i>T value</i>	-4.71	-18.29	-38.75	-30.31
	<i>P value</i>	0.000	0.000	0.000	0.000
	<i>% Imp</i>	1.426	35.880	37.726	<b>44.356</b>
MDBA versus DBA	<i>T value</i>	-1.00	-4.74	-5.11	-4.23
	<i>P value</i>	0.326	0.000	0.000	0.000
	<i>% Imp</i>	0.879	<b>7.243</b>	2.572	4.611
MDBA versus ABC	<i>T value</i>	-1.44	-19.10	-34.11	-30.25
	<i>P value</i>	0.161	0.000	0.000	0.000
	<i>% Imp</i>	0.220	<b>34.190</b>	31.106	33.343
MDBA versus KH	<i>T value</i>	-28.21	-22.97	-53.54	-63.97
	<i>P value</i>	0.000	0.000	0.000	0.000
	<i>% Imp</i>	20.000	39.326	41.507	<b>46.013</b>
MDBA versus MKH	<i>T value</i>	-4.71	-21.65	-41.35	-38.64
	<i>P value</i>	0.000	0.000	0.000	0.000
	<i>% Imp</i>	1.426	37.008	39.005	<b>42.409</b>
DBA versus ABC	<i>T value</i>	0.73	-13.06	-30.91	-22.33
	<i>P value</i>	0.470	0.000	0.000	0.000
	<i>% Imp</i>	-0.665	29.051	29.287	<b>30.120</b>
DBA versus KH	<i>T value</i>	-19.86	-16.27	-50.65	-44.26
	<i>P value</i>	0.000	0.000	0.000	0.000
	<i>% Imp</i>	19.291	34.588	39.963	<b>43.403</b>
DBA versus MKH	<i>T value</i>	-0.63	-14.78	-38.54	-35.45
	<i>P value</i>	0.531	0.000	0.000	0.000
	<i>% Imp</i>	0.552	32.089	37.395	<b>39.625</b>

Remark \* mean that all minimum values (total penalty cost) are identical.

## 5 Conclusions and future work

This research has developed a novel APS tool that effectively solves production scheduling problems for capital goods with many levels of product structure and multiple

resource constraints. It was the first research to adopt a Discrete Bat Algorithm (DBA), a Modified Discrete Bat Algorithm (MDBA) with additional local search and a Hybrid Discrete Bat Algorithm with Krill Herd (HDBK) for solving this problem. This required a novel representation to be developed that included product structure relationships and operations to be included. A repair process was included to ensure that operation and assembly precedence relationships were satisfied as well as taking into account finite capacity and avoiding deadlock. The tool was tested using four datasets obtained from a collaborating capital goods company. These had been used by previous researchers investigating ABC, KH and MKH optimisation. The first experiment identified appropriate parameter settings for the DBA. All of the DBA's parameters except  $\gamma$  were statistically significant. The only two-way interaction that was statistically significant was  $\alpha$ \**repositioning operation*. The appropriate parameter settings for the DBA were determined by considering the lowest mean best-so-far total cost obtained from main effect and interaction plots. The best settings were  $nI = 100*25$ ,  $\gamma = 0.1$ ,  $\alpha = 0.9$  with the adjustment operator (AO).

The second experiment was aimed to evaluate and compare the performance of the proposed HDBK with MDBA, DBA, ABC, KH, and MKH by using the Student *t*-test. The minimum total penalty costs indicated that the HDBK outperformed the other approaches (MDBA, DBA, ABC, KH, and MKH) for all problem sizes except small problem. Almost all of the comparisons between the results obtained from the HDBK and the other approaches for extra-large problem were statistically significant with a 95% confidence interval ( $P$  value  $\leq 0.05$ ) except the MDBA. The HDBK achieved the highest %*Imp* 47.837 when compared with the KH. These results demonstrate that the HDBK is a promising approach for advanced planning and scheduling systems for complex scheduling situations such as those encountered in the capital goods industry.

Future research may focus on the application of mathematical analysis and/or metaheuristics to solve production scheduling problem in capital goods industry. The integration of production and preventive maintenance scheduling problem in the capital goods industry or other integrations (e.g. lot sizing, or uncertainty issues in manufacturing environment) can also be another research direction in the future.

## Acknowledgement

The first author would like to acknowledge funding from the Thailand Research Fund Royal Golden Jubilee Ph.D. Scholarships Program - Grant Number PHD/0135/2552.

## References

- Acha, V., A. Davies, M. Hobday, and A. Salter. 2004. "Exploring the capital goods economy: complex product systems in the UK." *Industrial and Corporate Change* 13 (3):505-529.
- Agrawal, P.K., M. Pandit, and H.M. Dubey. 2016. "Improved Krill Herd Algorithm with neighborhood distance concept for optimization." *International Journal of Intelligent Systems and Applications* 8 (11):34-50. doi: 10.5815/ijisa.2016.11.05.
- Ahmadi, A.H., and S.K.Y. Nikraves. 2016. A novel instantaneous exploitation based Bat Algorithm. Paper presented at the 24th Iranian Conference on Electrical Engineering, ICEE 2016, Shiraz, Iran, 10th -12th May 2016.
- Al-Salamah, M. 2015. "Constrained binary Artificial Bee Colony to minimize the makespan for single machine batch processing with non-identical job sizes." *Applied Soft Computing Journal* 29:379-385. doi: 10.1016/j.asoc.2015.01.013.
- Amara, J., T.M. Hamdani, and A.M. Alimi. 2015. A new Hybrid Discrete Bat Algorithm for Traveling Salesman Problem using ordered crossover and 3-Opt operators for Bat's local search. Paper presented at the 15th International Conference on Intelligent Systems Design and Applications, ISDA 2015, Marrakesh, Morocco, 14th - 16th December 2015.
- Asadzadeh, L. 2016. "A parallel Artificial Bee Colony Algorithm for the job shop scheduling problem with a dynamic migration strategy." *Computers & Industrial Engineering* 102:359-367. doi: <https://doi.org/10.1016/j.cie.2016.06.025>.
- Banharnsakun, A., B. Sirinaovakul, and T. Achalakul. 2012. "Job shop scheduling with the best-so-far ABC." *Engineering Applications of Artificial Intelligence* 25 (3):583-593. doi: 10.1016/j.engappai.2011.08.003.
- Bansal, J.C., H. Sharma, and S.S. Jadon. 2013. "Artificial Bee Colony Algorithm: A survey." *International Journal of Advanced Intelligence Paradigms* 5 (1-2):123-159. doi: 10.1504/IJAIP.2013.054681.
- Bean, J.C. 1994. "Genetic Algorithms and random keys for sequencing and optimization." *ORSA Journal on Computing* 6 (2):154-160.
- Blum, C., and A. Roli. 2003. "Metaheuristics in combinatorial optimization: overview and conceptual comparison." *Acm Computing Surveys* 35 (3):268-308.

- Caniyilmaz, E., B. Benli, and M.S. Ilkay. 2015. "An Artificial Bee Colony Algorithm approach for unrelated parallel machine scheduling with processing set restrictions, job sequence-dependent setup times, and due date." *International Journal of Advanced Manufacturing Technology* 77 (9-12):2105-2115. doi: 10.1007/s00170-014-6614-9.
- Chainual, A., T. Lutuksin, and P. Pongcharoen. 2007. "Computer based scheduling tool for multi-product scheduling problems." *International Journal of the Computer, the Internet and Management* 15 (4):26-31.
- Chansombat, S., P. Musikapun, P. Pongcharoen, and C. Hicks. 2013. A modified Bat Algorithm for production scheduling in the capital goods industry. Paper presented at the 22nd International Conference on Production Research, Iguassu Falls, Brazil, 28th July – 2nd August 2013.
- Chaudhry, S.S., and W. Luo. 2005. "Application of Genetic Algorithms in production and operations management: A review." *International Journal of Production Research* 43 (19):4083-4101.
- Chen, K., P. Ji, and Q. Wang. 2011. "A case study for advanced planning and scheduling (APS)." *Journal of Systems Science and Systems Engineering* 20 (4):460-474.
- Chua, P.S., A.H.M. Salleh, M.S. Mohamad, S. Deris, S. Omatu, and M. Yoshioka. 2015. "Identifying a gene knockout strategy using a hybrid of the Bat Algorithm and flux balance analysis to enhance the production of succinate and lactate in *Escherichia coli*." *Biotechnology and Bioprocess Engineering* 20 (2):349-357. doi: 10.1007/s12257-014-0466-x.
- Civicioglu, P. 2013a. "Artificial cooperative search algorithm for numerical optimization problems." *Information Sciences* 229:58-76. doi: 10.1016/j.ins.2012.11.013.
- Civicioglu, P. 2013b. "Backtracking Search Optimization Algorithm for numerical optimization problems." *Applied Mathematics and Computation* 219:8121-8144.
- Cui, Z., and X. Gu. 2012. An improved discrete Artificial Bee Colony Algorithm for hybrid flow shop problems. Paper presented at the International Conference on Intelligent Computing for Sustainable Energy and Environment, Shanghai, China, 12th-13th September 2012.
- Cui, Z., and X. Gu. 2015. "An improved Discrete Artificial Bee Colony Algorithm to minimize the makespan on hybrid flow shop problems." *Neurocomputing* 148:248-259. doi: 10.1016/j.neucom.2013.07.056.
- Dao, T.-K., T.-S. Pan, and J.-S. Pan. 2018. "Parallel Bat Algorithm for optimizing makespan in job shop scheduling problems." *Journal of Intelligent Manufacturing* 29 (2):451-462.
- Dayou, L., Y. Pu, and Y. Ji. 2009. "Development of a multiobjective GA for advanced planning and scheduling problem." *International Journal of Advanced Manufacturing Technology* 42 (9-10):974-992. doi: 10.1007/s00170-008-1653-8.
- De Oliveira, I.M.S., and R. Schirru. 2011. "Swarm intelligence of artificial bees applied to in-core fuel management optimization." *Annals of Nuclear Energy* 38 (5):1039-1045. doi: 10.1016/j.anucene.2011.01.009.
- Deng, G., Z. Xu, and X. Gu. 2012. "A discrete Artificial Bee Colony Algorithm for minimizing the total flow time in the blocking flow shop scheduling." *Chinese Journal of Chemical Engineering* 20 (6):1067-1073. doi: 10.1016/S1004-9541(12)60588-6.
- Dorigo, M. 1992. "Optimization, Learning and Natural Algorithms." PhD, Politecnico di Milano, Italie.
- Dos Santos Coelho, L., and A. Askarzadeh. 2016. "An enhanced Bat Algorithm approach for reducing electrical power consumption of air conditioning systems based on differential operator." *Applied Thermal Engineering* 99:834-840. doi: 10.1016/j.applthermaleng.2016.01.155.

- Eusuff, M., K. Lansey, and F. Pasha. 2006. "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization." *Engineering Optimization* 38 (2):129-154. doi: 10.1080/03052150500384759.
- Fauceglia, D. 2014. "Credit constraints and firm imports of capital goods: Evidence from middle- and low-income countries." *International Economics* 140:1-18. doi: <https://doi.org/10.1016/j.inteco.2014.07.002>.
- Fister Jr, I., D. Fister, and X.S. Yang. 2013. "A hybrid Bat Algorithm." *Elektrotehnicki Vestnik/Electrotechnical Review* 80 (1-2):1-7.
- Fister Jr, X.S. Yang, I. Fister, J. Brest, and D. Fister. 2013. "A Brief Review of Nature-Inspired Algorithms for Optimization." *Elektrotehnicki Vestnik/Electrotechnical Review* 80 (3):116-122.
- Fogel, L., A.J. Owens, and M.J. Walsh. 1966. *Artificial Intelligence through Simulated Evolution*: Wiley.
- Fuchigami, H.Y., and S. Rangel. 2017. "A survey of case studies in production scheduling: Analysis and perspectives." *Journal of Computational Science*. doi: <http://dx.doi.org/10.1016/j.jocs.2017.06.004>.
- Gandomi, A.H., and A.H. Alavi. 2012. "Krill Herd: A new bio-inspired optimization algorithm." *Communications in Nonlinear Science and Numerical Simulation* 17 (12):4831-4845. doi: <http://dx.doi.org/10.1016/j.cnsns.2012.05.010>.
- Gao, K.Z., P.N. Suganthan, T.J. Chua, C.S. Chong, T.X. Cai, and Q.K. Pan. 2015. "A two-stage Artificial Bee Colony Algorithm scheduling flexible job-shop scheduling problem with new job insertion." *Expert Systems with Applications* 42 (21):7652-7663. doi: <https://doi.org/10.1016/j.eswa.2015.06.004>.
- Gao, K.Z., P.N. Suganthan, Q.K. Pan, T.J. Chua, C.S. Chong, and T.X. Cai. 2016. "An improved Artificial Bee Colony Algorithm for flexible job-shop scheduling problem with fuzzy processing time." *Expert Systems with Applications* 65:52-67. doi: <https://doi.org/10.1016/j.eswa.2016.07.046>.
- Gao, W.F., and S.Y. Liu. 2012. "A modified Artificial Bee Colony Algorithm." *Computers and Operations Research* 39 (3):687-697. doi: 10.1016/j.cor.2011.06.007.
- Geem, Z.W., J.H. Kim, and G.V. Loganathan. 2001. "A new heuristic optimization algorithm: harmony search." *Simulation* 76 (2):60-68.
- Gen, M., and R. Cheng. 1997. *Genetic Algorithms and Engineering Design*. New York: John Wiley and Sons.
- Gen, M., and L. Lin. 2014. "Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey." *Journal of Intelligent Manufacturing* 25 (5):849-866. doi: 10.1007/s10845-013-0804-4.
- Glover, F. 1990. "Tabu Search - a Tutorial." *Interfaces* 20 (4):74-94.
- Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Reading, MA: Addison-Wesley.
- Gupta, N., and S. Sharma. 2016. "Nature-inspired techniques for optimization: A brief review." *International Journal of Advance Research in Science and Engineering* 5 (5):36-44.
- Han, Y.Y., J.J. Liang, Q.K. Pan, J.Q. Li, H.Y. Sang, and N.N. Cao. 2013. "Effective hybrid discrete Artificial Bee Colony algorithms for the total flowtime minimization in the blocking flowshop problem." *International Journal of Advanced Manufacturing Technology* 67 (1-4):397-414. doi: 10.1007/s00170-012-4493-5.
- Hernández, H., and C. Blum. 2012. "Distributed graph coloring: An approach based on the calling behavior of Japanese tree frogs." *Swarm Intelligence* 6 (2):117-150. doi: 10.1007/s11721-012-0067-2.

- Hicks, C. 1998. "Computer Aided Production Management (CAPM) Systems in Make-to-order / Engineer-to-order Heavy Engineering Companies." PhD, University of Newcastle.
- Hicks, C., and P.M. Braiden. 2000. "Computer Aided Production Management issues in the engineer-to-order production of complex capital goods explored using a simulation approach." *International Journal of Production Research* 38 (18):4783-4810.
- Holland, J. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. 1st ed. U.S.A: University of Michigan Press.
- Hvolby, H.-H., and K. Steger-Jensen. 2010. "Technical and industrial issues of Advanced Planning and Scheduling (APS) systems." *Computers in Industry* 61 (9):845-851.
- Kang, F., J. Li, and H. Li. 2013. "Artificial Bee Colony Algorithm and pattern search hybridized for global optimization." *Applied Soft Computing Journal* 13 (4):1781-1791. doi: 10.1016/j.asoc.2012.12.025.
- Karaboga, D. 2005. "An Idea Based On Honey Bee Swarm For Numerical Optimisation." Erciyes University, Erciyes, Turkey, Report No. report-tr06.
- Karaboga, D., and B. Akay. 2009. "A comparative study of Artificial Bee Colony Algorithm." *Applied Mathematics and Computation* 214 (1):108-132. doi: <http://dx.doi.org/10.1016/j.amc.2009.03.090>.
- Karaboga, D., and B. Basturk. 2007. "A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) Algorithm." *Journal of Global Optimization* 39 (3):459-471. doi: 10.1007/s10898-007-9149-x.
- Karaboga, D., B. Gorkemli, C. Ozturk, and N. Karaboga. 2014. "A comprehensive survey: Artificial Bee Colony (ABC) Algorithm and applications." *Artificial Intelligence Review* 42 (1):21-57. doi: 10.1007/s10462-012-9328-0.
- Kashan, A.H. 2009. League Championship Algorithm: A new algorithm for numerical function optimization. Paper presented at the International Conference of Soft Computing and Pattern Recognition, Malacca, Malaysia, 4th - 7th December 2009.
- Kaur, N., and A. Chhabra. 2016. Analytical review of three latest nature inspired algorithms for scheduling in clouds. Paper presented at the Electrical, Electronics, and Optimisation Techniques (ICEEOT), Chennai, India, 3rd-5th March 2016.
- Kennedy, J., and R. Eberhart. 1995. Particle Swarm Optimization. Paper presented at the IEEE International Conference on Neural Networks, Perth, Australia, 27th November - 1st December 1995.
- Khorsandi, A., S.H. Hosseinian, and A. Ghazanfari. 2013. "Modified Artificial Bee Colony Algorithm based on fuzzy multi-objective technique for optimal power flow problem." *Electric Power Systems Research* 95:206-213. doi: 10.1016/j.epsr.2012.09.002.
- Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598):671-680.
- Kizilay, D., M.F. Tasgetiren, O. Bulut, and B. Bostan. 2014. A discrete Artificial Bee Colony Algorithm for the assignment and parallel machine scheduling problem in DY0 paint company. Paper presented at the Evolutionary Computation (CEC), Beijing, China, 6th - 11th July 2014.
- Kongkaew, W. 2016. Solving the single machine total weighted tardiness problem using bat-inspired algorithm. Paper presented at the IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2015, Singapore, 6th - 9th December 2015.
- Koza, J.R. 1990. *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*. Vol. 34: Stanford University, Department of Computer Science Stanford, CA.

- Lei, D. 2009. "Multi-objective production scheduling: a survey." *International Journal of Advanced Manufacturing Technology* 43 (9-10):925-938. doi: 10.1007/s00170-008-1770-4.
- Lei, D. 2013. "Multi-objective Artificial Bee Colony for interval job shop scheduling with flexible maintenance." *International Journal of Advanced Manufacturing Technology* 66 (9-12):1835-1843.
- Lenin, K., B.R. Reddy, and M.S. Kalavathi. 2014. "Dolphin Echolocation Algorithm for Solving Optimal Reactive Power Dispatch Problem." *International Journal of Computer (IJC)* 12 (1):1-15.
- Li, J.-Q., Q.-K. Pan, and M.F. Tasgetiren. 2014. "A discrete Artificial Bee Colony Algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities." *Applied Mathematical Modelling* 38 (3):1111-1132.
- Li, J.Q., and Q.K. Pan. 2015. "Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid Artificial Bee Colony Algorithm." *Information Sciences* 316:487-502. doi: 10.1016/j.ins.2014.10.009.
- Li, J.Q., Q.K. Pan, and K.Z. Gao. 2011. "Pareto-based discrete Artificial Bee Colony Algorithm for multi-objective flexible job shop scheduling problems." *International Journal of Advanced Manufacturing Technology* 55 (9-12):1159-1169. doi: 10.1007/s00170-010-3140-2.
- Li, J.Q., S.X. Xie, Q.K. Pan, and S. Wang. 2011. "A hybrid Artificial Bee Colony Algorithm for flexible job shop scheduling problems." *International Journal of Computers, Communications and Control* 6 (2):286-296.
- Li, L., and Y. Zhou. 2014. "A novel complex-valued Bat Algorithm." *Neural Computing and Applications* 25 (6):1369-1381. doi: 10.1007/s00521-014-1624-y.
- Li, S. 2017. "Parallel batch scheduling with inclusive processing set restrictions and non-identical capacities to minimize makespan." *European Journal of Operational Research* 260 (1):12-20. doi: 10.1016/j.ejor.2016.11.044.
- Li, X., Z. Peng, B. Du, J. Guo, W. Xu, and K. Zhuang. 2017. "Hybrid Artificial Bee Colony Algorithm with a rescheduling strategy for solving flexible job shop scheduling problems." *Computers and Industrial Engineering* 113:10-26. doi: 10.1016/j.cie.2017.09.005.
- Li, X., and M. Yin. 2012. "A discrete Artificial Bee Colony Algorithm with composite mutation strategies for permutation flow shop scheduling problem." *Scientia Iranica* 19 (6):1921-1935. doi: 10.1016/j.scient.2012.10.034.
- Lin, S.W., and K.C. Ying. 2014. "ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times." *Computers and Operations Research* 51:172-181. doi: 10.1016/j.cor.2014.05.013.
- Lin, S.W., K.C. Ying, and C.Y. Huang. 2013. "Multiprocessor task scheduling in multistage hybrid flowshops: A hybrid artificial bee colony algorithm with bi-directional planning." *Computers and Operations Research* 40 (5):1186-1195. doi: 10.1016/j.cor.2012.12.014.
- Liu, Y.F., and S.Y. Liu. 2013. "A hybrid discrete Artificial Bee Colony Algorithm for permutation flowshop scheduling problem." *Applied Soft Computing Journal* 13 (3):1459-1463. doi: 10.1016/j.asoc.2011.10.024.
- Lourenco, H.R., O.C. Martin, and T. Stützle. 2003. "Iterated Local Search." In *Handbook of Metaheuristics*, edited by Fred Glover and Gary A Kochenberger, 320-353. U.S.: Springer.
- Luo, J., Q. Wang, and X. Xiao. 2013. "A modified Artificial Bee Colony Algorithm based on converge-onlookers approach for global optimization." *Applied Mathematics and Computation* 219 (20):10253-10262. doi: 10.1016/j.amc.2013.04.001.

- Luo, Q., Y. Zhou, J. Xie, M. Ma, and L. Li. 2014. "Discrete Bat Algorithm for optimal problem of permutation flow shop scheduling." In *Scientific World Journal*, 1-16. Hindawi Publishing Corporation.
- Manda, K., S.C. Satapathy, and B. Poornasatyanarayana. 2012. "Population based meta-heuristic techniques for solving optimization problems: A selective survey." *International Journal of Emerging Technology and Advanced Engineering* 2 (11):206-211.
- Marichelvam, M., and T. Prabaharam. 2012. "A Bat Algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time." *ICTACT Journal on Soft Computing* 3 (1):428-433.
- Marichelvam, M., T. Prabaharan, X.-S. Yang, and M. Geetha. 2013. "Solving hybrid flow shop scheduling problems using Bat Algorithm." *International Journal of Logistics Economics and Globalisation* 5 (1):15-29.
- Montgomery, D.C. 2012. *Design and Analysis of Experiments*. 8th ed. New York: John Wiley & Sons.
- Moscato, P., and M.G. Norman. 1992. "A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems." *Parallel computing and transputer applications* 1:177-186.
- Mukherjee, A., and V. Mukherjee. 2016. "Chaos embedded Krill Herd Algorithm for optimal VAR dispatch problem of power system." *International Journal of Electrical Power and Energy Systems* 82:37-48. doi: 10.1016/j.ijepes.2016.02.039.
- Musikapun, P., and P. Pongcharoen. 2012. Solving Multi-Stage Multi-Machine Multi-Product Scheduling Problem Using Bat Algorithm. Paper presented at the International Conference on Management and Artificial Intelligence, Singapore, 7th April 2012.
- Muthiah, A., and R. Rajkumar. 2014. "A comparison of Artificial Bee Colony Algorithm and Genetic Algorithm to minimize the makespan for job shop scheduling." *Procedia Engineering* 97:1745-1754. doi: 10.1016/j.proeng.2014.12.326.
- Muthulakshmi, B., and K. Somasundaram. 2017. "A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment." *Cluster Computing*:1-9. doi: 10.1007/s10586-017-1174-z.
- Na, H., and J. Park. 2014. "Multi-level job scheduling in a flexible job shop environment." *International Journal of Production Research* 52 (13):3877-3887. doi: 10.1080/00207543.2013.848487.
- Nagar, A., J. Haddock, and S. Heragu. 1995. "Multiple and bicriteria scheduling: a literature survey." *European Journal of Operational Research* 81 (1):88-104.
- Nanda, S.J., and G. Panda. 2014. "A survey on nature inspired metaheuristic algorithms for partitioned clustering." *Swarm and Evolutionary Computation* 16:1-18. doi: 10.1016/j.swevo.2013.11.003.
- Nasiri, M.M. 2015. "A modified ABC algorithm for the stage shop scheduling problem." *Applied Soft Computing Journal* 28:81-89. doi: 10.1016/j.asoc.2014.12.001.
- Nawaz, M., E.E. Enscore Jr, and I. Ham. 1983. "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem." *Omega* 11 (1):91-95. doi: 10.1016/0305-0483(83)90088-9.
- Pan, Q.-K., L. Gao, X.-Y. Li, and K.-Z. Gao. 2017. "Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times." *Applied Mathematics and Computation* 303:89-112. doi: <https://doi.org/10.1016/j.amc.2017.01.004>.
- Pan, Q.K., M. Fatih Tasgetiren, P.N. Suganthan, and T.J. Chua. 2011. "A discrete Artificial Bee Colony Algorithm for the lot-streaming flow shop scheduling problem." *Information Sciences* 181 (12):2455-2468. doi: 10.1016/j.ins.2009.12.025.

- Pan, Q.K., L. Wang, J.Q. Li, and J.H. Duan. 2014. "A novel discrete Artificial Bee Colony Algorithm for the hybrid flowshop scheduling problem with makespan minimisation." *Omega (United Kingdom)* 45:42-56. doi: 10.1016/j.omega.2013.12.004.
- Pan, Q.K., L. Wang, K. Mao, J.H. Zhao, and M. Zhang. 2013. "An effective Artificial Bee Colony Algorithm for a real-world hybrid flowshop problem in steelmaking process." *IEEE Transactions on Automation Science and Engineering* 10 (2):307-322. doi: 10.1109/TASE.2012.2204874.
- Pansuwan, P., N. Rukwong, and P. Pongcharoen. 2010. Identifying optimum Artificial Bee Colony (ABC) algorithm's parameters for scheduling the manufacture and assembly of complex products. Paper presented at the Second International Conference on Computer and Network Technology, Bangkok, Thailand, 23rd - 25th April 2010.
- Pei, J., X. Liu, W. Fan, P.M. Pardalos, and S. Lu. 2017. "A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers." *Omega (United Kingdom)*. doi: 10.1016/j.omega.2017.12.003.
- Pinedo, M., and X. Chao. 1999. *Operations scheduling with applications in manufacturing and services*. Boston: McGraw-Hill.
- Pongcharoen, P. 2001. "Genetic algorithms for production scheduling in capital goods industries." PhD, University of Newcastle upon Tyne.
- Pongcharoen, P., C. Hicks, and P.M. Braiden. 2004. "The development of Genetic Algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure." *European Journal of Operational Research* 152 (1):215-225. doi: 10.1016/S0377-2217(02)00645-8.
- Pongcharoen, P., C. Hicks, P.M. Braiden, and D.J. Stewardson. 2002. "Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products." *International Journal of Production Economics* 78 (3):311-322. doi: 10.1016/S0925-5273(02)00104-4.
- Pongcharoen, P., H. Puangyeam, K. Pawinand, S. Vitayasak, and A. Khadwilard. 2012. Artificial Bee Colony with Random Key Technique for Production Scheduling in Capital Goods Industry. Paper presented at the 2012 2nd International Conference on Industrial Technology and Management (ICITM 2012), Phuket, Thailand, 1st September 2012.
- Pravesjit, S. 2016. "A hybrid Bat Algorithm with natural-inspired algorithms for continuous optimization problem." *Artificial Life and Robotics* 21 (1):112-119. doi: 10.1007/s10015-015-0248-3.
- Prugel-Bennett, A. 2010. "Benefits of a population: Five mechanisms that advantage population-based algorithms." *IEEE Transactions on Evolutionary Computation* 14 (4):500-517.
- Pu, Y., L. Dayou, Y. Donghui, and Y. Ji. 2007. Genetic Algorithm with local search for advanced planning and scheduling. Paper presented at the 3rd International Conference on Natural Computation, ICNC 2007, Haikou, Hainan, 24th - 27th August 2007.
- Puangyeam, H., P. Pongcharoen, and S. Vitayasak. 2014. Application of Krill Herd (KH) Algorithm for Production Scheduling in Capital Goods Industries. Paper presented at the International Conference on challenges in IT, Engineering and Technology, Phuket, Thailand, 17th -18th July 2014.
- Rechenberg, I. 1965. "Cybernetic Solution Path of an Experimental Problem." Royal Aircraft Establishment, Farnborough, UK, Report No. 1122.
- Ribas, I., R. Companys, and X. Tort-Martorell. 2015. "An efficient Discrete Artificial Bee Colony algorithm for the blocking flow shop problem with total flowtime minimization." *Expert Systems with Applications* 42 (15-16):6155-6167. doi: 10.1016/j.eswa.2015.03.026.

- Sang, H., L. Gao, and Q. Pan. 2012. "Discrete Artificial Bee Colony Algorithm for lot-streaming flowshop with total flowtime minimization." *Chinese Journal of Mechanical Engineering (English Edition)* 25 (5):990-1000. doi: 10.3901/CJME.2012.05.990.
- Selvi, V., R. Uma Rani, and R. Sankar. 2014. "Hybridization of evolutionary and swarm intelligence techniques for job scheduling problem." *International Journal of Engineering and Technology* 6 (3):1519-1529.
- Storn, R., and K. Price. 1997. "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces." *Journal of Global Optimization* 11 (341-359).
- Sundar, S., and A. Singh. 2012. "A swarm intelligence approach to the early/tardy scheduling problem." *Swarm and Evolutionary Computation* 4:25-32. doi: 10.1016/j.swevo.2011.12.002.
- Sundar, S., P.N. Suganthan, C.T. Jin, C.T. Xiang, and C.C. Soon. 2017. "A hybrid Artificial Bee Colony Algorithm for the job-shop scheduling problem with no-wait constraint." *Soft Computing* 21 (5):1193-1202. doi: 10.1007/s00500-015-1852-9.
- Talbi, E.-G. 2009. *Metaheuristics: from design to implementation*. Vol. 74: John Wiley & Sons.
- Tasgetiren, M.F., Y.-C. Liang, M. Sevkli, and G. Gencyilmaz. 2007. "A Particle Swarm Optimization Algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem." *European Journal of Operational Research* 177 (3):1930-1947.
- Tasgetiren, M.F., Q.-K. Pan, P.N. Suganthan, and A. Oner. 2013. "A discrete Artificial Bee Colony Algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion." *Applied Mathematical Modelling* 37 (10–11):6758-6779. doi: <http://dx.doi.org/10.1016/j.apm.2013.02.011>.
- Tasgetiren, M.F., Q.K. Pan, P.N. Suganthan, and A.H.L. Chen. 2010. A discrete artificial bee colony algorithm for the permutation flow shop scheduling problem with total flowtime criterion. Paper presented at the 2010 IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, 18th-23th July 2010.
- Tasgetiren, M.F., Q.K. Pan, P.N. Suganthan, and A.H.L. Chen. 2011. "A discrete Artificial Bee Colony Algorithm for the total flowtime minimization in permutation flow shops." *Information Sciences* 181 (16):3459-3475. doi: <http://dx.doi.org/10.1016/j.ins.2011.04.018>.
- Thammano, A., and A. Phu-Ang. 2013. "A hybrid Artificial Bee Colony Algorithm with local search for flexible job-shop scheduling problem." *Procedia Computer Science* 20:96-101.
- Tosun, Ö., and M.K. Marichelvam. 2016. "Hybrid Bat Algorithm for flow shop scheduling problems." *International Journal of Mathematics in Operational Research* 9 (1):125-138. doi: 10.1504/IJMOR.2016.077560.
- Vijaychakaravarthy, G., S. Marimuthu, and A.N. Sait. 2014. "Comparison of Improved Sheep Flock Heredity Algorithm and Artificial Bee Colony Algorithm for lot Streaming in m-Machine Flow Shop Scheduling." *Arabian Journal for Science and Engineering* 39 (5):4285-4300.
- Wang, B., Z. Guan, S. Ullah, X. Xu, and Z. He. 2017. "Simultaneous order scheduling and mixed-model sequencing in assemble-to-order production environment: a multi-objective hybrid Artificial Bee Colony Algorithm." *Journal of Intelligent Manufacturing* 28 (2):419-436. doi: 10.1007/s10845-014-0988-2.
- Wang, C., J. Zhang, J. Yang, C. Hu, and J. Liu. 2005. A modified Particle Swarm Optimization Algorithm and its application for solving traveling salesman problem. Paper presented at the International Conference on Neural Networks and Brain, Beijing, China, 13-15 Oct. 2005.

- Wang, G.-G., S. Deb, and L. Coelho. 2016. "Earthworm Optimization Algorithm: a bio-inspired metaheuristic algorithm for global optimization problems." *International Journal of Bio-Inspired Computation* 8 (6):394-409.
- Wang, G.-G., S. Deb, X.-Z. Gao, and L.D.S. Coelho. 2016. "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour." *International Journal of Bio-Inspired Computation* 8 (6):394-409.
- Wang, G.-G., A. Hossein Gandomi, and A. Hossein Alavi. 2013. "A Chaotic Particle-Swarm Krill Herd Algorithm for global numerical optimization." *Kybernetes* 42 (6):962-978.
- Wang, G., L. Guo, A.H. Gandomi, L. Cao, A.H. Alavi, H. Duan, and J. Li. 2013. "Lévy-flight Krill Herd Algorithm." *Mathematical Problems in Engineering* 2013. doi: 10.1155/2013/682073.
- Wang, G., L. Guo, H. Wang, H. Duan, L. Liu, and J. Li. 2012. "Incorporating mutation scheme into Krill Herd Algorithm for global numerical optimization." *Neural Computing and Applications*:1-19. doi: 10.1007/s00521-012-1304-8.
- Wang, G.G. 2016. "Moth Search Algorithm: a bio-inspired metaheuristic algorithm for global optimization problems." *Memetic Computing*:1-14. doi: 10.1007/s12293-016-0212-3.
- Wang, G.G., S. Deb, and Z. Cui. 2015. "Monarch Butterfly Optimization." *Neural Computing and Applications*:1-20. doi: 10.1007/s00521-015-1923-y.
- Wang, G.G., S. Deb, A.H. Gandomi, Z. Zhang, and A.H. Alavi. 2016. "Chaotic Cuckoo Search." *Soft Computing* 20 (9):3349-3362. doi: 10.1007/s00500-015-1726-1.
- Wang, G.G., S. Deb, and S.M. Thampi. 2015. A Discrete Krill Herd Method with multilayer coding strategy for flexible job-shop scheduling problem. Paper presented at the Intelligent Systems Technologies and Applications, Kochi, India, 10th-13th August 2015.
- Wang, K.P., L. Huang, C.G. Zhou, and W. Pang. 2003. Particle Swarm Optimization for Traveling Salesman Problem. Paper presented at the Machine Learning and Cybernetics, Xi'an, China, 5th November 2003.
- Wang, L., G. Zhou, Y. Xu, and M. Liu. 2012. "An enhanced Pareto-based Artificial Bee Colony Algorithm for the multi-objective flexible job-shop scheduling." *International Journal of Advanced Manufacturing Technology* 60 (9-12):1111-1123. doi: 10.1007/s00170-011-3665-z.
- Wang, L., G. Zhou, Y. Xu, S. Wang, and M. Liu. 2012. "An effective Artificial Bee Colony Algorithm for the flexible job-shop scheduling problem." *International Journal of Advanced Manufacturing Technology* 60 (1-4):303-315. doi: 10.1007/s00170-011-3610-1.
- Wang, X., X. Xie, and T.C.E. Cheng. 2013. "A modified Artificial Bee Colony Algorithm for order acceptance in two-machine flow shops." *International Journal of Production Economics* 141 (1):14-23. doi: <http://dx.doi.org/10.1016/j.ijpe.2012.06.003>.
- Xie, J., Y. Zhou, and Z. Tang. 2013. Differential Lévy-flights Bat Algorithm for minimization makespan in permutation flow shops. Paper presented at the 9th International Conference on Intelligent Computing, ICIC 2013, Nanning, China, 28th - 31st July 2013.
- Xie, W., C. Hicks, and P. Pongcharoen. 2010. An enhanced single-objective Genetic Algorithm scheduling tool for solving very large scheduling problems in capital goods industry. Paper presented at the 16th International Working Seminar on Production Economics, Innsbruck, Austria, 1st - 5th March 2010.
- Xu, H., Z.R. Bao, and T. Zhang. 2017. "Solving dual flexible job-shop scheduling problem using a Bat Algorithm." *Advances in Production Engineering And Management* 12 (1):5-16. doi: 10.14743/apem2017.1.235.

- Xu, Y., Z. Cui, and J. Zeng. 2010. Social Emotional Optimization Algorithm for nonlinear constrained optimization problems. Paper presented at the International Conference on Swarm, Evolutionary, and Memetic Computing, Chennai, India 16th-18th December 2010.
- Yan, G.W., Z.J. Hao, and J. Xie. 2013. "A novel atmosphere clouds model optimization algorithm." *Journal of Computers (Taiwan)* 24 (3):26-39.
- Yang, X.-S. 2010a. *Nature-Inspired Metaheuristic Algorithm*. Second ed: Luniver Press.
- Yang, X.S. 2010b. "A new metaheuristic bat-inspired algorithm." *Studies in Computational Intelligence* 284:65-74.
- Yang, X.S. 2012. Flower Pollination Algorithm for global optimization. Paper presented at the International Conference on Unconventional Computing and Natural Computation, Orléans, France, 3rd to 7th September 2012.
- Yang, X.S. 2013. "Bat Algorithm: Literature review and applications." *International Journal of Bio-Inspired Computation* 5 (3):141-149. doi: 10.1504/IJBIC.2013.055093.
- Yang, X.S., and S. Deb. 2009. Cuckoo search via Levy flights. Paper presented at the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009).. IEEE Publication, Coimbatore, India, 9th - 11th December 2009.
- Yilmaz, S., and E.U. Kucuksille. 2013. "Improved Bat Algorithm (IBA) on continuous optimization problems." *Lecture Notes on Software Engineering* 1 (3):279-283.
- Yue, L., Z. Guan, U. Saif, F. Zhang, and H. Wang. 2016. "Hybrid Pareto Artificial Bee Colony Algorithm for multi-objective single machine group scheduling problem with sequence-dependent setup times and learning effects." *SpringerPlus* 5 (1). doi: 10.1186/s40064-016-3265-3.
- Zaher, H., N. Ragaa, and H. Sayed. 2017. "A novel Improved Bat Algorithm for Job Shop Scheduling Problem." *International Journal of Computer Applications* 164 (5):24-30.
- Zambonelli, F., and M. Viroli. 2011. "A survey on nature-inspired metaphors for pervasive service ecosystems." *International Journal of Pervasive Computing and Communications* 7 (3):186-204. doi: 10.1108/17427371111172997.
- Zhang, R., P.-C. Chang, S. Song, and C. Wu. 2017. "A multi-objective Artificial Bee Colony Algorithm for parallel batch-processing machine scheduling in fabric dyeing processes." *Knowledge-Based Systems* 116:114-129.
- Zhang, R., S. Song, and C. Wu. 2013. "A hybrid Artificial Bee Colony Algorithm for the job shop scheduling problem." *International Journal of Production Economics* 141 (1):167-178. doi: <http://dx.doi.org/10.1016/j.ijpe.2012.03.035>.