International Journal of Advances in Production Research

The Bees Algorithm and Its Applications in Production and Manufacturing

D T Pham[1,a*], *Yanjie Song*[2,b], *Yuanjun Laili*[3,c], *Asrul Ismail*[4,d], *Natalia Hartono*[1,e], *Michael Packianather*[5,f] and *Marco Castellani*[1,g]

The University of Birmingham[1], United Kingdom

School of Reliability and System Engineering, Beihang University[2], China

Beihang University[3], China

University of Portsmouth[4], United Kingdom

Cardiff University[5], United Kingdom

d.t.pham@bham.ac.uk[a], yanjiesong96@gmail.com[b], lailiyuanjun@buaa.edu.cn[c], asrul.ismail@port.ac.uk[d], n.hartono@bham.ac.uk[e], packianatherms@cf.ac.uk[f], m.castellani@bham.ac.uk[g]

***Abstract*— With the advent of the Fourth Industrial Revolution, production and manufacturing processes and systems have become more complex. Obtaining the best performance from them requires efficient and effective optimisation techniques that do not depend on the availability of process or system models. Such models are usually either not obtainable or mathematically intractable due to the high degrees of nonlinearities and uncertainties in the processes and systems to be represented. The Bees Algorithm is a powerful swarm-based intelligent optimisation metaheuristic inspired by the foraging behaviour of honeybees. The algorithm is conceptually elegant and extremely easy to apply. It has attracted users from virtually all fields of engineering and natural, physical, medical and social sciences. This article reviews the original Bees Algorithm and some of its recent enhancements and gives examples of its applications to optimisation problems in production and manufacturing. The aim is to demonstrate the simplicity, effectiveness and versatility of the algorithm and encourage its further adoption by engineers and researchers across the world to realise smart and sustainable manufacturing and production in the age of Industry 4.0 and beyond.**

***Keywords*— Bees Algorithm; swarm intelligence; intelligent optimisation; metaheuristics; production optimisation; manufacturing optimisation; smart manufacturing; Industry 4.0**

I. INTRODUCTION

Today's smart production systems are often distributed and highly interconnected. Many newly developed manufacturing processes are complex, nonlinear and not fully characterised or understood. Obtaining the best performance from those processes and systems requires a different approach from those based on mathematical modelling and optimisation. This is because it is difficult and perhaps even impossible to obtain accurate models for them due to inherent uncertainties. To optimise modern production and manufacturing systems and processes, efficient techniques that do not involve mathematical modelling may be necessary.

The Bees Algorithm, a metaheuristic inspired by the foraging behaviour of honey bees, is eminently suitable for handling complex optimisation problems. All it needs to solve an optimisation problem is a means to evaluate the quality of potential solutions. This is true even if the problem is non-deterministic polynomial complete (NP-complete), i.e., the toughest decision problem in the NP category of problems whose solutions cannot be guaranteed to be found, but can be verified, in polynomial time. NP-complete problems are at least as hard as any other NP problem, meaning finding an efficient algorithm for one solves them all. This is the reason why, since the algorithm was first published by Pham et al. in 2005 [1], it has attracted users from virtually all fields of engineering and natural, physical, medical and social sciences.

This article gives an overview of the Bees Algorithm and its applications in production and manufacturing. Following an introduction to the main ideas underpinning the algorithm, including its global search (exploration) and local search (exploitation) techniques, the article will present recent results and developments relating to the algorithm and its application to production and manufacturing optimisation problems. The article will demonstrate the simplicity, effectiveness and versatility of the algorithm. It is hoped that this will encourage further adoption of this tool to realise smart and sustainable manufacturing and production in the era of the fourth industrial revolution and beyond.

II. BEE FORAGING BEHAVIOUR

*Corresponding Author

When searching for food, bees perform both exploration and exploitation. A honey bee colony can have thousands of bees. About 10% of the colony is sent up to 10 km around the hive to go looking for food – those are known as scout bees. Scout bees randomly explore the fields surrounding the hive in search of pollen and nectar. When they have found an interesting food source, they return to the hive to recruit more bees to follow them and help gather the food and bring it back to the colony. To recruit additional bees, the scouts perform a dance known as the waggle dance. The other bees in the hive observe the dance and then decide which scout to follow, i.e., which food source to visit and exploit. According to their fitness, food sources (flower patches) may be visited by more or fewer bees or may be abandoned.

Thus, we have seen the essential points of the Bees Algorithm. Scout bees explore the landscape. Their aim is to find the most abundant source of food to feed the colony. The scouts recruit additional forager bees to help them exploit the discovered food sources. The more abundant the food source, the more foragers will be recruited. When nectar and pollen in a flower patch run short, the bees stop foraging it and move elsewhere. The process continues until the best food source is located.

III. The Original Bees Algorithm

Figure 1 is the flow chart of the original Bees Algorithm proposed in 2005 [1, 2].

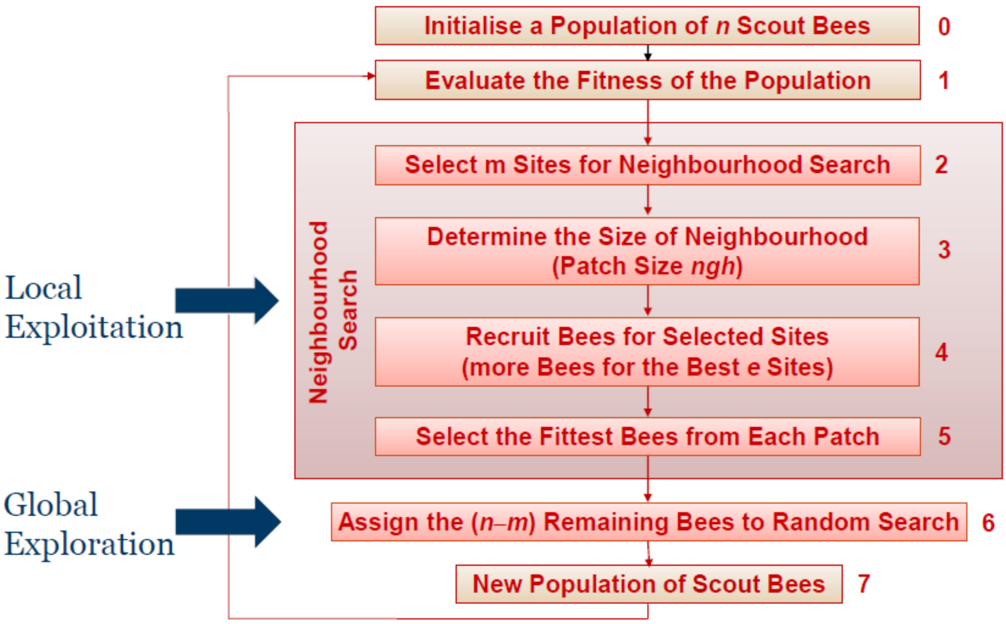

Figure 1. The original Bees Algorithm (Pham et al, 2005; Pham et al., 2006)

The algorithm begins with n scout bees moving randomly in space looking for food (Step 0).

The quality of the flower patches (or sites) found by the scout bees is evaluated (Step 1).

A certain number of high-ranking patches (m) are then selected for further exploitation (Step 2).

The size of the search area around those patches (ngh) is defined (Step 3).

The scout bees that found high-quality flower patches are allocated follower bees to help them exploit those patches and their surroundings. The e scout bees that found the very best quality patches are allocated the largest number of followers (Step 4).

In each search area, the bee that found the best flower patch is identified (Step 5). This results in m bees for the next iteration.

Those n- m scouts without followers are again assigned to explore the search space and look for other flower patches (Step 6).

A new population of scouts is thus formed comprising the m bees of Step 5 and the (n-m) bees of Step 6 (Step 7).

The cycle begins again in Step 1 until a stopping criterion is met. That could be when a sufficiently high-quality flower patch has been found or a preset number of iterations has been reached.

The example in Figure 2 illustrates how the algorithm finds the maximum point of a one-dimensional function.

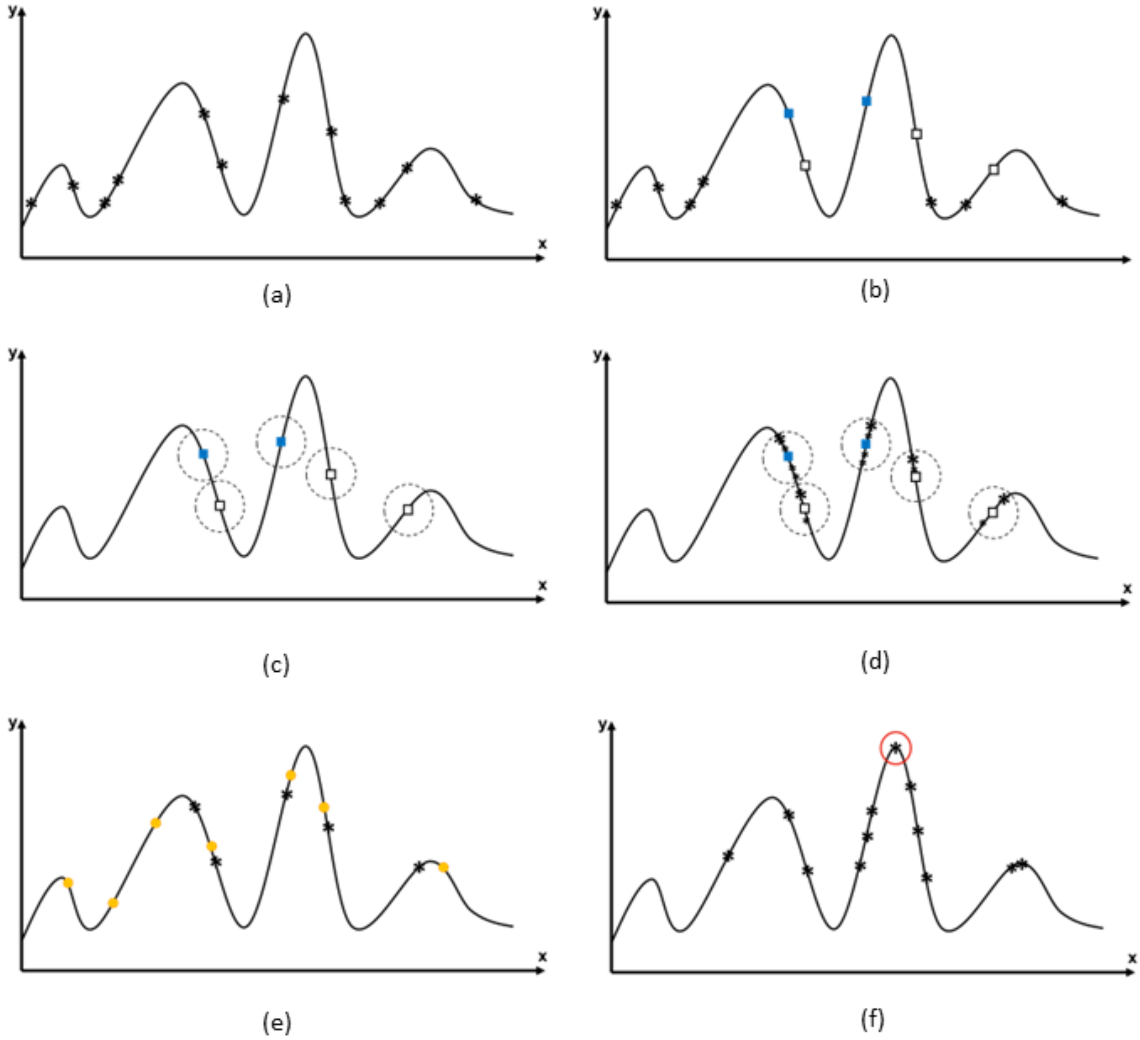

FIGURE 2. ONE-D OPTIMISATION EXAMPLE (A) TWELVE SCOUT BEES (N= 12) REPRESENTING TWELVE RANDOM SOLUTIONS (ASTERISKS); (B) THE TOP FIVE SOLUTIONS (M= 5; WHITE AND BLUE SQUARES); (C) NEIGHBOURHOODS OF THE TOP FIVE SOLUTIONS; (D) RECRUITED BEES ASSIGNED TO THE NEIGHBOURHOODS OF THE TOP SOLUTIONS (MORE RECRUITED BEES FOR THE NEIGHBOURHOODS OF THE TOP 2 SOLUTIONS (E = 2), I.E., THE BLUE SQUARES); (E) THE TWELVE SCOUT BEES FORMING THE POPULATION IN THE NEW ITERATION (THE ORANGE CIRCLES REPRESENT THE SOLUTIONS FOUND BY THE 7 SCOUT BEES (N-M = 7) NOT PREVIOUSLY USED FOR LOCAL EXPLOITATION; (F) CONVERGENCE TO THE GLOBAL OPTIMUM (ASTERISK IN THE RED CIRCLE)

IV. FEATURES OF THE ORIGINAL BEES ALGORITHM AND SOME (MAINLY) EARLY ENHANCEMENTS

The simplicity of the original Bees Algorithm arises from the fact that it does not require any special interaction mechanism for exchanging information between the bees. For instance, there is no pheromone deposition as with the Ant Algorithm or particle velocity computation like in the case of the Particle Swarm Optimisation algorithm. The Bees Algorithm also has built-in strategies for mobilising resources to exploit promising areas of the search space to locate potential solutions and avoid local optimum traps. The selective recruitment method explained above, whereby foragers focus on the best flower patches, is what makes the algorithm efficient at finding locally optimum solutions.

Although the Bees Algorithm's greedy exploitation method enables it quickly to discover potential solutions, there is a risk of premature convergence to false optima. In addition to random exploration by scout bees, which helps maintain diversity in the solution population to prevent premature convergence, the Bees Algorithm also employs neighbourhood shrinking and site abandonment to escape from local optimum traps. This strategy involves gradually reducing the size of the local search area when the solution does not improve and abandoning that area after a predetermined number of attempts.

In the early years of the development of the Bees Algorithm, it was realised that the new solutions discovered by the scout bees in Steps 7/8 (Figure 1) tended to be poor. Forcing them to compete immediately with those already in the population is counterproductive. However, as a new promising solution may be at the foot of a tall mountain and be worse than old solutions that have stalled at a lower peak, it should be protected and allowed to develop its potential to reach greater heights.

A new category of bees was thus introduced - the "young bees", that is, scout bees that were created in recent iterations of the optimisation process [3]. These new bees are shielded from the stronger adults in the general population. They are permitted to compete for survival only among themselves until they reach "adulthood" after having evolved a given number of iterations when they are ready to 'fight' against other more established bees.

The "young bees" idea was a rare excursion into the global search mechanism of the Bees Algorithm. Early work on enhancing the Bees Algorithm tended to focus on local search methods and produced a variety of operators to improve their effectiveness. In addition to mutation and crossover, there were operators such as interpolation, extrapolation and creep [3], as well as those based on TRIZ [4] and Kalman filtering [5]. The use of different operators gave rise to different versions of the Bees Algorithm with different features making them suitable for a variety of applications.

Over the years, several versions of the Bees Algorithm have been developed by hybridising the bee foraging metaphor with other metaheuristics. The Bees Algorithm has also been used in conjunction with computational intelligence tools such as neural networks and fuzzy logic. Table I is a non-exclusive list of hybrid algorithms and paired applications. Many more hybrid versions could be invented by combining the Bees Algorithm with new optimisation methods when available. Similarly, the Bees Algorithm could be synergistically used with other new intelligent tools to tune them, refine their outputs or improve its own performance.

For an extensive survey of other versions of the Bees Algorithm created before 2017, the reader is referred to the article by Hussein et al [6]. A more recent review can be found in the doctoral thesis by Ismail [7]. It will be seen that large amounts of effort were spent on making the algorithm faster. Research was also directed at reducing its parameters from the six or seven that need to be set (namely, n scout bees, m high-performing bees, e 'elite' bees, ne forager bees assigned to the 'elite' bees, np forager bees assigned to the m-e high-performing bees, ngh, the neighbourhood size, and r, the neighbourhood shrinking rate if neighbourhood shrinking is adopted).

TABLE I
HYBRIDISATION AND COMBINATION

'Partner' Algorithm	Method	Authors
Ant Colony Optimisation (ACO)	Using pheromone for increased social interaction.	[8] Packianather *et al*., 2009
Particle Swarm Optimisation (PSO)	'Elite' and other successful scouts acting together as a swarm or as leaders of their own swarms.	[9] Pham and Sholedolu, 2008
Simulated Annealing (SA) and Direct Annealing (DA)	Using SA or DA-based selection to update all elite and non-elite bees.	[10 Sadiq and Hamad, 2010; [11] Masmoudi *et al*., 2016
Genetic Algorithm (GA) and Differential Evolution (DE) Algorithm	Using mutation and crossover search operators and biased selection operator.	[12] Anh and Vi, 2023
Firefly Algorithm	Using firefly rules in global search.	[13] Gholami and Mohammadi, 2018
Crow Search Algorithm	Optimising in two stages.	[14] Deghbouch and Debbat, 2022
Fuzzy Logic	Transforming a multi-objective optimisation problem into a fuzzy inference system using fuzzy objective functions.	[15] Tolabi *et al.*, 2014
Artificial Neural Networks (ANN)	Using ANN to obtain a model and the BA to optimise its parameters.	[16] Ebrahimpoor *et al.*, 2019
Support Vector Machines (SVM)	Using the BA for feature selection before classification by SVM.	[17] Abdusalam *et al*., 2023
Convolutional Neural Networks (CNN)	Using the BA to optimise CNNs	[18] Alamri *et al*., 2022

V. Three Recent Versions of the Bees Algorithm

This section highlights three new versions of the Bees Algorithm, of which one was designed to speed up its operation, another to simplify its parameter setting and the third, to achieve both simplicity and fast convergence. The first version is the dual population Bees Algorithm by Song et al [19], the second version is the two-parameter Bees Algorithm by Ismail et al [20] and the third version is the super-fast Bees Algorithm by Laili et al [21].

A. Dual-Population Bees Algorithm

Figure 3 [19] is a flowchart of this version of the Bees Algorithm. As its name implies, the algorithm uses two populations of bees, one with which to perform greedy (local) search and the other for increasing population diversity. The individual sizes of the two populations vary dynamically. The total size of the two populations is kept constant. The 'search' population contains bees that perform above the average of the two populations. The 'supplementary' population comprises below-average bees. Bees from the search population are demoted to the supplementary population if they fall below the average. Bees in the supplementary population can be probabilistically selected to create the population for the next iteration. Bees from the search population are demoted to the supplementary population if they fall below the average. Bees in the supplementary population can be probabilistically selected to create the population for the next iteration.

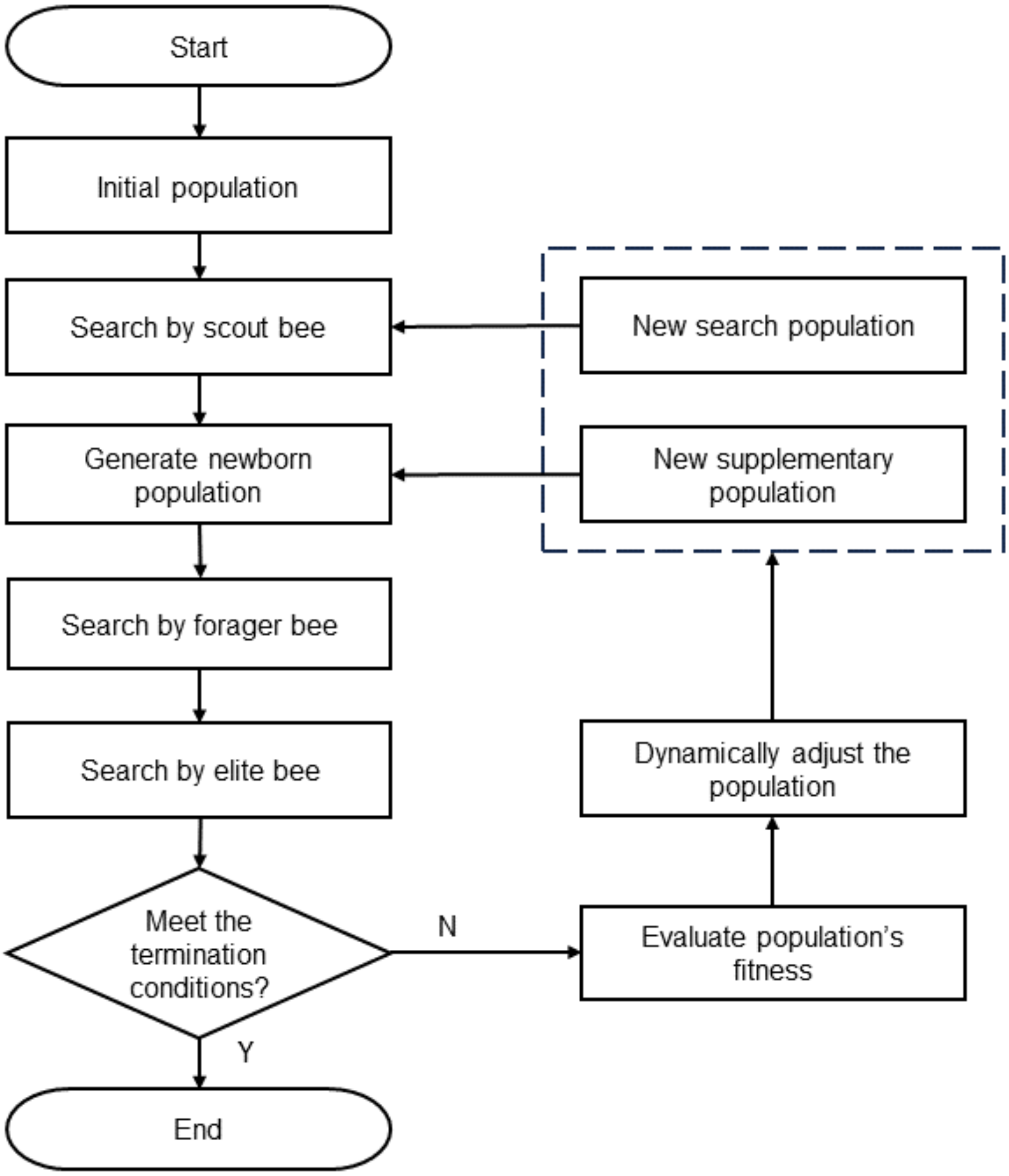

Figure 3. Flowchart of Dual-Population Bees Algorithm [19]

It should be noted that this version of the Bees Algorithm is significantly different from the basic version described previously. This shows that Bees Algorithm research has virtually unlimited possibilities.

B. Two-parameter Bees Algorithm (BA2)

This algorithm follows the operation of the original Bees Algorithm but differs from it in two respects related to recruitment and exploitation. The algorithm is summarised in Figure 4 which also shows its parameters and those of the original algorithm [20].

0. Initialisation
1. Population ranking
2. **Forager recruitment**
3. **Local exploitation** and **global exploration**
4. Population updating
5. Returning to Step 1 until a stopping criterion is met

The steps of the two-parameter BA

Original BA	**2-parameter BA**
n	***n***
m	-
e	1
n_e	n_e
n_p	-
ngh	-

FIGURE 4. THE TWO-PARAMETER BA AND ITS USER-SELECTABLE PARAMETERS [20]

As with the original Bees Algorithm, the user specifies the number n of scout bees.

For the recruitment of foragers, the user sets the number ne of bees that follow the top scout bee (the single 'elite' bee that found the best flower patch or solution). A decreasing function is used to allocate foragers to follow the other scout bees. For example, the second best-performing bee could recruit half of ne, the third best-performing bee could have half of that number of foragers again, etc. Thus, there is no need for the user to choose m, e and np.

Another difference with the original Bees Algorithm is that in BA2, bees perform exploration and exploitation at the same time. Some followers focus on exploiting the area surrounding the solution discovered by the scout. Others wander around exploring other parts of the solution space. The higher the ranking of the scout (i.e., the better the flower patch that it found), the larger the number of followers that focus on exploiting the patch and the smaller the number of wanderers.

In BA2, the concept of neighbourhood is generalised to cover the whole solution space, and the user no longer has to choose ngh. For each scout, a probability function is used to determine the distribution of the foragers in the solution space, i.e., which followers focus on the discovered solution and which ones stray further away.

C. Super-fast Bees Algorithm

This is the simplest version of the Bees Algorithm developed to date. The algorithm uses only 3 bees. It conducts greedy local search with the best-performing bee and applies genetic operators (precedence preserving crossover and mutation) using the middle-ranking bee. The poorest-performing bee may be used for the crossover operation or may be discarded. If it is not crossed with the middle-ranking bee, the top-performing bee is used instead. In both cases, a new bee is randomly created to replace the poorest bee, thus maintaining population diversity.

Figure 5 [21] depicts the super-fast Bees Algorithm. The boxes labelled 1, 2 and 3 represent the actions of the best-performing, middle-ranking and worst bees, respectively. Local exploitation is conducted by the best bee, and global exploration, by the worst bee. The middle-ranking bee could be viewed as executing a hybrid of local and global search. As the population size is fixed and there is only one elite high-performing bee, the user does not have to specify n, m or e. No forager bees are employed for neighbourhood search and thus ne, np and ngh are not required either.

The super-fast Bees Algorithm, also called the Ternary Bees Algorithm because it uses three bees, is extremely efficient. This is a result of having a very small population. However, this could arguably be the algorithm's potential weakness causing it to converge quickly to false optima in some problems.

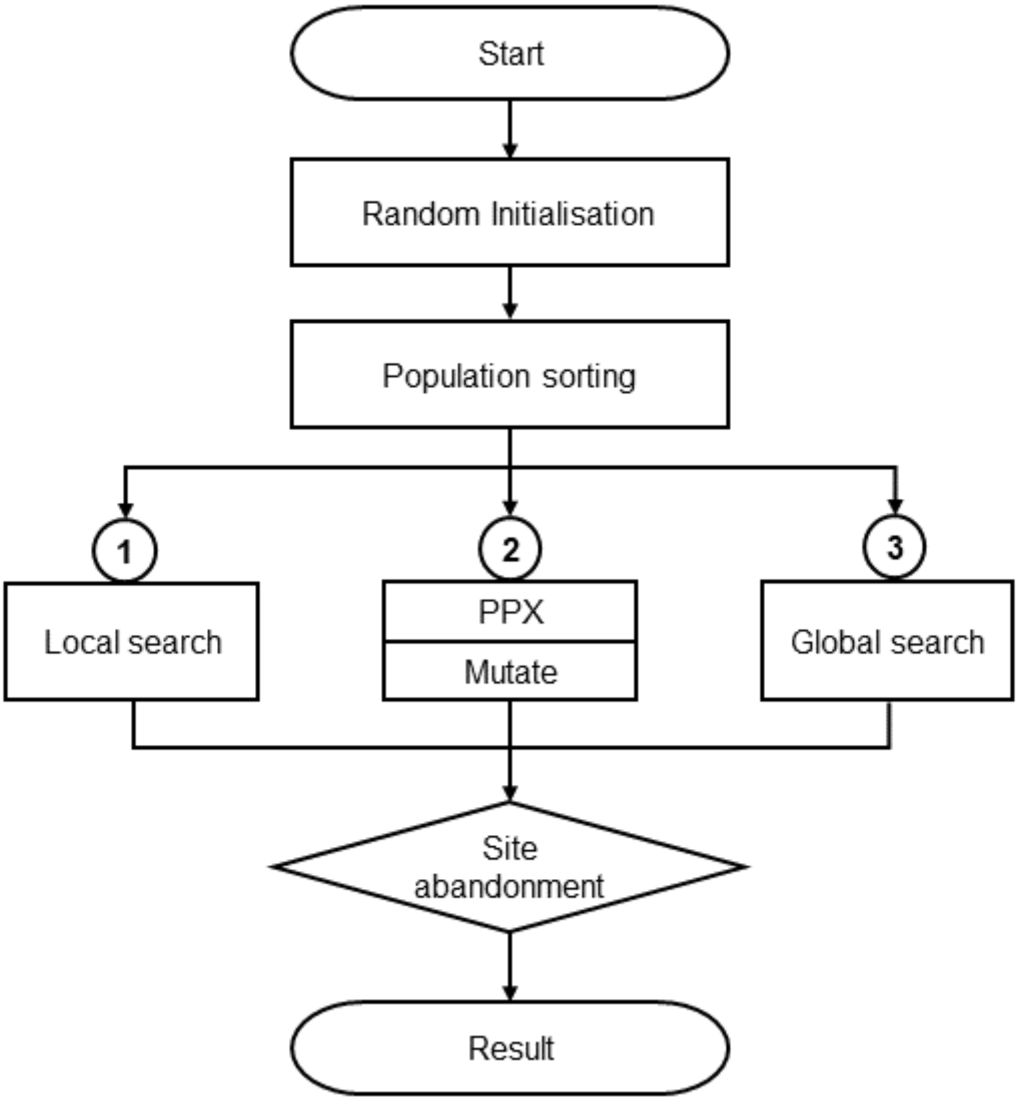

FIGURE 5. THE SUPER-FAST BEES ALGORITHM [21]

VI. SOME APPLICATIONS OF THE BEES ALGORITHM IN PRODUCTION AND MANUFACTURING

This section reviews some recent uses of the Bees Algorithm to solve optimisation problems in production and manufacturing. Both continuous and combinatorial optimisation problems have been addressed.

A. Continuous Optimisation

Packianather et al. [22] employed the Bees Algorithm to optimise process parameters for wire electrical discharge machining. The authors leveraged the algorithm to obtain the best combination of process parameters for the digital twin of the product being machined. Conte and d'Addona [23] applied the Bees Algorithm to tool wear identification and measurement during turning operations. The author's goal was to use the bees to define the contours of the wear area of a tool and locate the point of maximum wear.

Ay et al. [24] applied Bee-Miner, a cost-sensitive classification algorithm for data mining derived from the Bees Algorithm, to classify defects in manufacturing. Zeybek [25] trained a Long-Short-Term Memory (LSTM) deep learning network to predict the remaining useful life (RUL) of turbofan engines before they are available for remanufacturing with the help of a modified version of the super-fast Bees Algorithm.

A memory-based Bees Algorithm with a Lévy search facility was used by Shatnawi et al. [26] to perform multilevel thresholding of images, a basic operation required when processing images for automated visual inspection or general computer vision. Castellani et al. [27] have also adopted the Bees Algorithm in work relevant to vision systems. They used the Bees Algorithm to fit primitive shapes to point-cloud scenes for real-time 3D object recognition. Lan et al. [28] solved the problem of finding a spatial transformation that aligns two point clouds with an enhanced Bees Algorithm. They employed singular value decomposition to increase the search efficiency of the Bees Algorithm, achieving higher consistency, precision and robustness than the popular Iterative Closest Point method.

Öztürk et al. [29] used the Bees Algorithm to find the parameters (pulse frequency, amplitude and base) for a test to obtain the biaxial stress-strain curves required to control the pulsating hydroforming process to yield a uniform thickness distribution. Şahin and Çakıroğlu [30] employed a multi-objective Bees Algorithm to design PID control systems such as those for robotic equipment. They successfully found the PID parameters to minimise the settling time, rise time, overshoot, and system error all at once. Continuing with robotics applications, Kashkash et al. [31] modified the Bees Algorithm by adding a new initial population generation method and used the modified algorithm to find the shortest collision-free path for a mobile robot.

Mechanical design was performed by Acar et al. [32] using a version of the Bees Algorithm hybridised with Harmony Search. The hybrid algorithm which incorporated a harmony memory was employed to optimise the design of spherical four-link mechanisms for robot grippers.

B. Combinatorial Problems

Ang and Ng [33] optimised process plans for printed circuit board assembly. They combined the Bees Algorithm with TRIZ operators to minimise assembly time. Song et al. [19] used their dual-population Bees Algorithm to perform parallel machine scheduling with time windows. Production planning and scheduling for a sheet metal fabrication plant was accomplished by Wang et al. [34] using both the original Bees Algorithm and the version implementing the site abandonment strategy.

Li et al. [35] scheduled tasks and selected resources for a cloud-edge collaborative manufacturing environment using a dynamic Bees Algorithm. The operators in the algorithm were adjusted according to a set of indicators, and a parallel sorting scheme was adopted to accelerate the scheduling and selection of cloud-edge resources and collaboration modes.

Task allocation in a cloud remanufacturing environment was performed by Caterino et al. [36] with the help of a Bees Algorithm the parameters of which were selected following a full-factorial experiment. Their work confirmed the importance of increasing the number of scout bees and exploiting the best-performing and elite sites.

Remanufacturing applications of the Bees Algorithm also include the optimisation of product disassembly which is the first operation in any product remanufacturing process chain. For example, Liu et al. [37] employed a discrete Bees Algorithm simultaneously to optimise disassembly sequences and balance the disassembly line, with the analytic process network assigning weights to the different optimisation objectives. Hartono et al. [38] adopted a multi-objective Bees Algorithm to devise robotic disassembly sequence plans to achieve maximum profit while minimising energy consumption and greenhouse gas emissions.

The Bees Algorithm has been used for logistics and supply chain optimisation. Mastrocinque [39] implemented a tool based on the Bees Algorithm to optimise the supply chain for a bulldozer based on its Bill of Materials, finding that the Bees Algorithm performed better than Ant Colony Optimisation. Finally, Ismail and Pham [40] used the two-parameter Bees Algorithm to solve the capacitated vehicle routing problem. To speed up the solution, a decomposition approach was adopted whereby customers were first clustered before the optimal route was found for each cluster.

VII. Conclusion

This paper has presented the main features of the original Bees Algorithm and some enhancements to it. Three new versions have also been reviewed. The paper concludes by giving examples of the use of the Bees Algorithm in production and manufacturing. There are many more potential applications, especially with the advent of the recently developed simpler and more efficient versions.

It has been seen how the original Bees Algorithm mimics the foraging behaviour of bees in nature where three types of bees exist. Scout bees perform exploration then go back to the hive to recruit forager bees to follow them and help exploit the discovered food sources. Those foragers are initially observer bees in the hive. They spring into action after watching the waggle dance by the scout bees.

It is hoped that, like observer bees, some readers of the paper will spring into action and become forager bees. In the field of metaheuristic optimisation, there is so much more to explore and exploit. The new versions outlined in this paper have only been used for a limited number of problems in production engineering. They could readily be applied to many more problems.

For researchers who have used other optimisation algorithms, comparing their results with those obtained using the Bees Algorithm will confirm the effectiveness of this powerful metaheuristic technique. Further research avenues also beckon, such as hybridising the Bees Algorithm with other tools and performing deep theoretical analysis of the resulting hybrid algorithms to understand their true areas of strength.

1. D T Pham, A Ghanbarzadeh, E Koc, S Otri, S Rahim and M Zaidi (2005) The Bees Algorithm, Technical Note, Manufacturing Engineering Centre, Cardiff University, UK

2. D T Pham, A Ghanbarzadeh, E Koç, S Otri, S Rahim and M Zaidi (2006). The Bees Algorithm - a novel tool for complex optimisation problems, Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems, D T Pham, E E Eldukhri and A J Soroka (eds), Elsevier (Oxford) (2006), 454-460, ISBN 0-08-045157-8

3. Q T Pham, D T Pham and M Castellani (2012). A modified bees algorithm and a statistics-based method for tuning its parameters, Proceedings of Institution of Mechanical Engineers, Part I, Volume 226, 287-301, ISSN 0954-4054

4. M C Ang, D T Pham and K W Ng (2009). Application of the Bees Algorithm with TRIZ-inspired operators for PCB assembly planning, Proceedings of the 5th Virtual International Conference on Innovative Production Machines and Systems, D T Pham, E E Eldukhri and A J Soroka (eds), Whittles (Dunbeath, Caithness) and CRC Press (Boca Raton, FL), 405-410, ISBN 978-1-84995-007-7W

5. D T Pham and A Haj Darwish (2010). Using the Bees Algorithm with Kalman filtering to train an Artificial Neural Network for Pattern Classification, Proceedings of Institution of Mechanical Engineers, Part I, Volume 224, 885-892, ISSN 0959 6518

6. A Hussein, S Sahran and S N H Sheikh Abdullah (2017). The variants of the Bees Algorithm (BA): a survey. Artif Intell Rev, 47, 67–121.

7. A H Ismail (2021). Enhancing the Bees Algorithm using the Traplining Metaphor, PhD Thesis, University of Birmingham, UK.

8. M Packianather, M Landy and D T Pham (2009). Enhancing the speed of the Bees Algorithm using pheromone-based recruitment classifiers, Proc IEEE 7th International Conference on Industrial Informatics (INDIN 2009), Cardiff, UK, 789-794, ISBN: 978-142443760-3

9. D T Pham and M Sholedolu (2008). Using a hybrid PSO-Bees Algorithm to train neural networks for wood defect classification, Proceedings of the 4th Virtual International Conference on Innovative Production Machines and Systems, D T Pham, E E Eldukhri and A J Soroka (eds), Whittles (Dunbeath, Caithness) and CRC Press (Boca Raton, FL), 385-390, ISBN 978-1-4398-0117-8

10. A T Sadiq and A G Hamad (2010). BSA: A hybrid bees' simulated annealing algorithm to solve optimization and NP-complete problems. Eng. & Tech. Journal, 28(2), 271-281

11. M A Masmoudi, M Hosny, K Braekers and A Dammak (2016). Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem, Transportation Research Part E: Logistics and Transportation Review, 96, 60-80, ISSN 1366-5545, https://doi.org/10.1016/j.tre.2016.10.002

12. T T M Anh and T D Vi (2023). Hybrid Genetic-Bees Algorithm in Multi-layer Perceptron Optimization. In: Saraswat, M., Chowdhury, C., Kumar Mandal, C., Gandomi, A.H. (eds) Proceedings of International Conference on Data Science and Applications. Lecture Notes in Networks and Systems, 552, Springer, Singapore. https://doi.org/10.1007/978-981-19-6634-7_11

13. J Gholami and S Mohammadi (2018). A Novel Combination of Bees and Firefly Algorithm to Optimize Continuous Problems, Proceedings of 8th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 40-46, https://doi.org/10.1109/ICCKE.2018.8566263

14. H Deghbouch and F Debbat (2022). Hybrid algorithm for node deployment with the guarantee of connectivity in wireless sensor networks, Informatica, 46(8), 147-164, https://doi.org/10.31449/inf.v46i8.3370

15. H B Tolabi, M H Ali, S B M Ayob and M Rizwan (2014). Novel hybrid fuzzy-Bees algorithm for optimal feeder multi-objective reconfiguration by considering multiple-distributed generation, Energy, 71, 507-515, https://doi.org/10.1016/j.energy.2014.04.099

16. S Ebrahimpoor, V Kiarostami, M Khosravi, M Davallo and A Ghaedi (2019). Bees metaheuristic algorithm with the aid of artificial neural networks for optimization of acid red 27 dye adsorption onto novel polypyrrole/SrFe12O19/graphene oxide nanocomposite. Polym. Bull., 76, 6529–6553, https://doi.org/10.1007/s00289-019-02700-7

17. O Abdusalam, F Anayi and M Packianather (2023). Three-Phase Power Transformer Fault Diagnosis Based on Support Vector Machines and Bees Algorithm, Proceedings 2023 IEEE 3rd International Conference in Power Engineering Applications (ICPEA), Putrajaya, Malaysia, 145-150, https://doi.org/10.1109/ICPEA56918.2023.10093147

18. N M H Alamri, M Packianather and S Bigot (2022). Deep Learning: Parameter Optimization Using Proposed Novel Hybrid Bees Bayesian Convolutional Neural Network, Applied Artificial Intelligence, 36(1), https://doi.org/10.1080/08839514.2022.2031815

19. Y Song, L Xing and Y Chen (2023). Application of the dual-population Bees Algorithm in a parallel machine scheduling problem with a time window. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_14

20. A H Ismail, W Ruslan and D T Pham (2023). A user-friendly Bees Algorithm for continuous and combinatorial optimisation, Cogent Engineering, 10:2, https://doi.org/10.1080/23311916.2023.2278257

21. Y Laili, F Tao, D T Pham, Y Wang and L Zhang (2019). Robotic disassembly re-planning using a two-pointer detection strategy and a super-fast bees algorithm, Robotics and Computer-Integrated Manufacturing, 59, 130-142, https://doi.org/10.1016/j.rcim.2019.04.00

22. M Packianather, T Alexopoulos and S Squire, S. (2023). The application of the Bees Algorithm in a digital twin for optimising the wire electrical discharge machining (WEDM) process parameters. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_3

23. S Conte and D M d'Addona (2023). Bees Algorithm models for the identification and measurement of tool wear. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_7

24. M Ay, A Baykasoglu, L Ozbakir and S Kulluk (2023). A case study with the BEE-Miner Algorithm: defects on the production line. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_4

25. S Zeybek (2023). Prediction of the remaining useful life of engines for remanufacturing using a semi-supervised deep learning model trained by the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_21

26. N Shatnawi, S Sahran and M F Nasrudin (2023). Memory-based Bees Algorithm with Lévy Flights for multilevel image thresholding. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_11

27. M Castellani, L Baronti, S Zheng and F Lan (2023). Shape recognition for industrial robot manipulation with the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_6

28. F Lan, M Castellani, Y Wang and S Zheng (2023). Global optimisation for point cloud registration with the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_8

29. O Öztürk, M A Şen, M Kalyoncu and H S Halkacı (2023). An application of the Bees Algorithm to pulsating hydroforming. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_5

30. M Şahin and S Çakıroğlu (2023). Automatic PID tuning toolkit using the Multi-Objective Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_9

31. M Kashkash, A Haj Darwish and A Joukhadar (2023) . A new method to generate the initial population of the Bees Algorithm for robot path planning in a static environment. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_12

32. O Acar, H Sağlam and Z Şaka (2023). The effect of Harmony Memory integration into the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_10

33. M C Ang and K W Ng (2023). Minimising printed circuit board assembly time using the Bees Algorithm with TRIZ-inspired operators. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_2

34. C Wang, T Chen and Z Li (2023). Method for the production planning and scheduling of a flexible manufacturing plant based on the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_13

35. Y Li, C Peng, Y Laili and L Zhang (2023). A parallel multi-indicator-assisted dynamic Bees Algorithm for cloud-edge collaborative manufacturing task scheduling. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_15

36. M Caterino, M Fera, R Macchiaroli and D T Pham (2023). Task optimisation for a modern cloud remanufacturing system using the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_20

37. J Liu, Q Liu, Z Zhou, D T Pham, W Xu and Y Fang (2023). Collaborative optimisation of robotic disassembly planning problems using the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_18

38. N Hartono, F Javier Ramírez and D T Pham (2023). Optimisation of robotic disassembly sequence plans for sustainability using the multi-objective Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_19

39. E Mastrocinque (2023). Supply chain design and multi-objective optimisation with the Bees Algorithm. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_17

40. A H Ismail and D T Pham (2023). Bees traplining metaphors for the Vehicle Routing Problem using a decomposition approach. In: Pham, D.T., Hartono, N. (eds) Intelligent Production and Manufacturing Optimisation—The Bees Algorithm Approach. Springer Series in Advanced Manufacturing. Springer, Cham. https://doi.org/10.1007/978-3-031-14537-7_16