Journal of Integrated System (JIS)
Vol. 7 No. 1 June 2024: 1-12
e-ISSN: 2621-7104

Received: 28 March 2024
Accepted: 8 June 2024
https://doi.org/10.28932/jis.v7i1.8602

# A Comparative Study of Reduced Parameter Versions of the Bees Algorithm for Traveling Salesman Problem

**Natalia Hartono[1*], Hamid Furkan Suluova[1], Fatih Mehmet Eker[1], Sultan Zeybek[2], Mario Caterino[3]**

[1]Mechanical Engineering Department, University of Birmingham, Birmingham, England
[2]Computer Engineering Department, Fatih Sultan Mehmet Vakif University, Istanbul, Turkey
[3]Industrial Engineering Department, University of Salerno, Fisciano, Italy
*Corresponding author, email: nxh886@alumni.bham.ac.uk

***Abstract***

*Metaheuristics have shown dominance over exact methods with their capability to find near-optimal solutions to complex problems in a shorter time. Among these metaheuristics, the Bees Algorithm (BA) has proven its performance in various applications. However, fine-tuning the parameters of the BA is challenging due to its numerous parameters. There have been few studies aiming to reduce the number of parameters while maintaining or improving performance, such as the ternary BA, two-parameter BA, and Fibonacci BA. This paper reviews these variants for combinatorial problems using 13 datasets from the Travelling Salesman Problem TSPLIB. The results were compared using an independent t-test in conjunction with descriptive statistics. The findings show that the Fibonacci BA outperforms other variants, and potential suggestions for improvements in the future were proposed.*

*Keywords: Bees Algorithm, Fibonacci, Metaheuristic, Travelling Salesman Problem*

## 1. Introduction

Metaheuristics have garnered significant attention over the last three decades due to their ability to find near-optimal solutions faster than exact approaches (Hussain *et al.*, 2019; Alorf, 2023). Many complex problems fall into the category of NP-hard or NP-complete, where the number of possible solutions grows exponentially with the problem size. Therefore, finding the best solution using an exact approach is often not feasible due to the time required (Juan *et al.*, 2023). Many real-world problems require a fast solution, even if it is not the best, as a near-optimal solution is sufficient given the time constraints.

Sörensen and Glover defined metaheuristics as a high-level problem-independent algorithm framework that provides a set of guidelines to solve optimisation problems, with the first term coined by Glover in 1986 (Sörensen and Glover, 2013). Metaheuristics are known to be able to effectively solve complex problems in engineering and many other problems (Sarhani, Voß and Jovanovic, 2023).

One such metaheuristic is the Bees Algorithm (BA), which has been employed in various fields such as scheduling, engineering, function optimisation, production, and manufacturing (Castellani and Pham, 2023). Introduced in 2005 by Pham et al. (Pham *et al.*, 2005), the Bees Algorithm was inspired by honeybee foraging behaviour and is particularly effective in finding nectar sources. This version is later referred to as the basic Bees Algorithm. BA is widely recognised for its versatility and superior performance compared to other algorithms such as Genetic Algorithm, Particle Swarm Optimisation, Simulated Annealing, and more (Pham and Castellani, 2009; Liu *et al.*, 2018; Laili *et al.*, 2019; Ismail, 2021). Previous studies have consistently demonstrated BA's robustness and effectiveness, making it a compelling choice for our research focus.

Since its inception, the BA has evolved and been combined with other algorithms, resulting in several improved versions. The most renowned version, known as the standard BA, was introduced in 2009 (Pham and Castellani, 2009) and includes a site abandonment strategy. The basic BA had five to six

**How to Cite:**
Hartono, N. *et al.* (2024) 'A comparative study of reduced parameter versions of the Bees Algorithm for Traveling Salesman Problem', *Journal of Integrated System*, 7(1), pp. 1–12. Available at: https://doi.org/10.28932/jis.v7i1.8602.

parameters to set, while the standard BA has seven to eight (Hartono, 2023; Hartono and Pham, 2024). The most recent versions strive to decrease the number of parameters to enhance usability, as setting numerous parameters necessitates more time before the algorithm can utilise its optimal settings. The reduced-parameter versions include the ternary BA ($BA_{Ternary}$) introduced in 2019 (Laili *et al.*, 2019), which has only one parameter; the two-parameter BA ($BA_2$) (Ismail, 2021); and the Fibonacci Bees Algorithm ($BA_F$) (Hartono, 2023), which has four parameters. Among these reduced-parameter versions, the $BA_F$ has demonstrated superior accuracy compared to basic BA (Hartono, 2023; Hartono and Pham, 2024), while $BA_2$ exhibits similar performance to basic BA (Ismail, 2021). The $BA_{Ternary}$ is the fastest version among these, designed specifically for real-time disassembly sequence re-planning problems. These versions have not been directly compared, and their applications vary. The $BA_{Ternary}$ is used in the two-point strategy for robotic disassembly re-planning, $BA_2$ in vehicle routing problems (VRP), and $BA_F$ in robotic disassembly sequence planning.

The objective of this study is to conduct a comprehensive analysis of the strengths of these versions in the same application, making it the first study to compare them in the combinatorial domain. The selected application is the classic combinatorial problem, the Travelling Salesman Problem (TSP). The corresponding programmes are provided via the GitHub link in the appendix. This work contributes to the development of the BA by exploring and comparing its basic and reduced parameter versions with the basic BA. This exploration can provide insights into the performance of each version under varying circumstances, potentially guiding future improvements and applications of the algorithm.

## 2. Literature Review

The Travelling Salesman Problem (TSP), first introduced in 1930, continues to be a topic of current research in the optimisation field (Pop *et al.*, 2024). It has been widely used as a standard for evaluating the effectiveness of optimisation algorithms (Pop *et al.*, 2024). TSP is classified as an NP-hard problem (Thong-ia and Champrasert, 2023; Toaza and Esztergár-Kiss, 2023), making it very challenging due to its search space growing exponentially with more cities, making exact solutions impractical for large instances. Therefore, metaheuristics have become a useful tool for addressing this problem (Ismail *et al.*, 2020; Toaza and Esztergár-Kiss, 2023; Pop *et al.*, 2024). The TSP's lasting popularity stems from its diverse practical applications in logistics, scheduling, manufacturing, DNA sequencing, and other fields (Pop *et al.*, 2024).

BA is a nature-inspired metaheuristic created to address continuous and discrete problems (Castellani and Pham, 2023). In the basic version of the algorithm (Pham *et al.*, 2005), parameters include the number of scout bees (*n*), number of selected sites (*m*), number of elite sites (*e*), recruited bees for elite sites (*nep*), recruited bees for other selected sites (*nsp*), and neighbourhood range (*ngh*). Figure 1 depicts the flowchart of the basic Bees Algorithm (BA). The standard version of BA introduced two additional parameters: neighbourhood shrinking and site abandonment (Pham and Castellani, 2009). The site abandonment procedure occurs when a specified number (*stlim*) of consecutive stagnation cycles is observed. At this point, it is assumed that the local search procedure has reached its peak of fitness, and further improvement is unlikely. Consequently, the examination of the patch is stopped, and a new random solution is generated.

The application of BA in the combinatorial domain was first presented in 2007 to tackle scheduling difficulties in manufacturing (Pham, Afify and Koç, 2007; Pham *et al.*, 2007). Similar to other metaheuristics, the main difference between the continuous and combinatorial versions of BA lies in their local operators and neighbourhood shrinking strategies. In the combinatorial versions, most BA variants utilise the entire neighbourhood size and do not employ neighbourhood shrinking. In the combinatorial domain, the basic BA utilises swap, insert, and reverse operators in its local search strategy for addressing the TSP (Ismail *et al.*, 2020). In the context of robotic disassembly, this version employs swap, insert, and mutate operators (Liu *et al.*, 2018). BA has been enhanced and integrated with other algorithms to improve its performance over time. The enhancements involve improving the local operators and, in certain versions, utilising fuzzy logic to remove the necessity for manual parameter adjustments. Refer to the study conducted by Hussein, Sahran, and Sheikh Abdullah for a survey of the various versions of the BA up to 2017 (Hussein, Sahran and Sheikh Abdullah, 2017).

As previously mentioned, BA is known for its numerous parameters (six to eight parameters), a topic that has received limited attention since its introduction in 2005. Concerns about the time-consuming nature of finding optimal parameter settings for metaheuristics have been raised. While the efficacy of parameter reduction as a definitive solution remains uncertain (Castellani and Pham, 2023), it does simplify the process for users to find optimal settings (Hartono, 2023; Hartono and Pham, 2024). This study aims to compare three parameter-reducing variants of the Bees Algorithm—$BA_{Ternary}$, $BA_2$, and $BA_F$—against the basic BA, each characterised by 1, 2, and 4 parameters, respectively.

$BA_{Ternary}$ is designed for re-planning in robotic disassembly sequences where there are only three bees employed. The best ones to search the elite sites, the second best to search the non-selected sites, and the last one for global search. $BA_{Ternary}$ closely resembles the SBA with the addition of a site abandonment strategy. However, as it is designed for the combinatorial domain, it does not include neighbourhood shrinking. Thus, the $BA_{Ternary}$ relies solely on a single parameter setting, *stlim*, which determines the number of stagnation cycles after which the site is abandoned, and a new random solution is generated. The $BA_{Ternary}$ was compared with other algorithms and shows it outperforms other algorithms and is suitable for real-time planning for small and medium-sized problems up to 200 disassembly parts (Laili *et al.*, 2019). Figure 2 depicts the flowchart of the $BA_{Ternary}$.

The $BA_2$ is designed for continuous problems and combinatorial problems, with applications in continuous function evaluation in the continuous domain and TSP and VRP in the combinatorial domain. Notably, the application of $BA_2$ in the combinatorial domain represents an improved version with local operators such as the Bees Routing Optimizer (BRO), and as such, no results have been published for the basic version of $BA_2$. However, the code for the basic $BA_2$ in the TSP can be accessed through GitHub (Ismail, 2022). Additionally, the local operator random selection for $BA_2$ is designed such that the reverse operator has a higher probability of selection compared to the swap and insert operators. Conversely, basic BA, $BA_{Ternary}$, and $BA_F$ employ equal probabilities for the local operators (swap, inverse, and reverse) when randomly selected. Figure 3 illustrates the flowchart of $BA_2$.

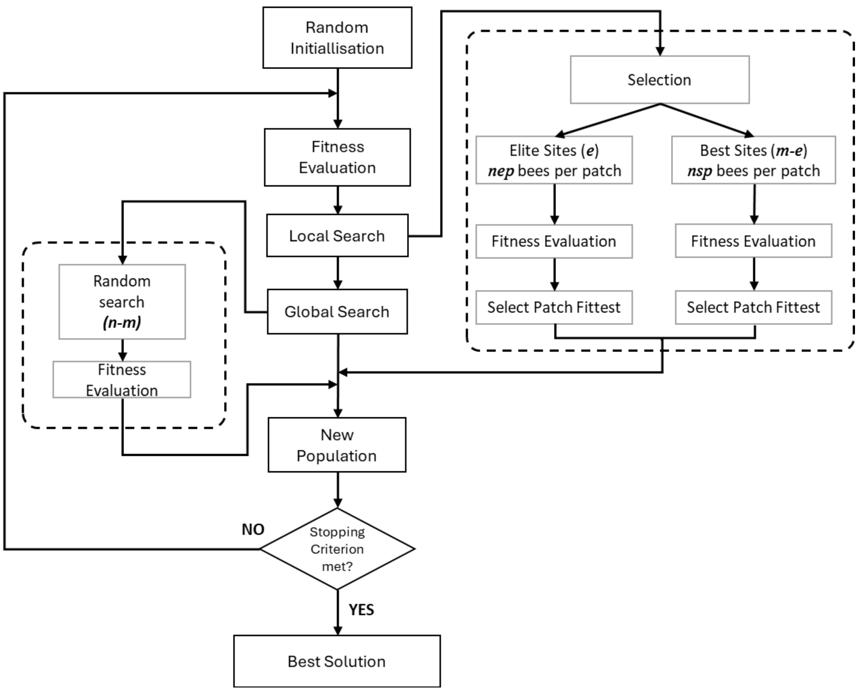


Figure 1. Bees Algorithm (BA)

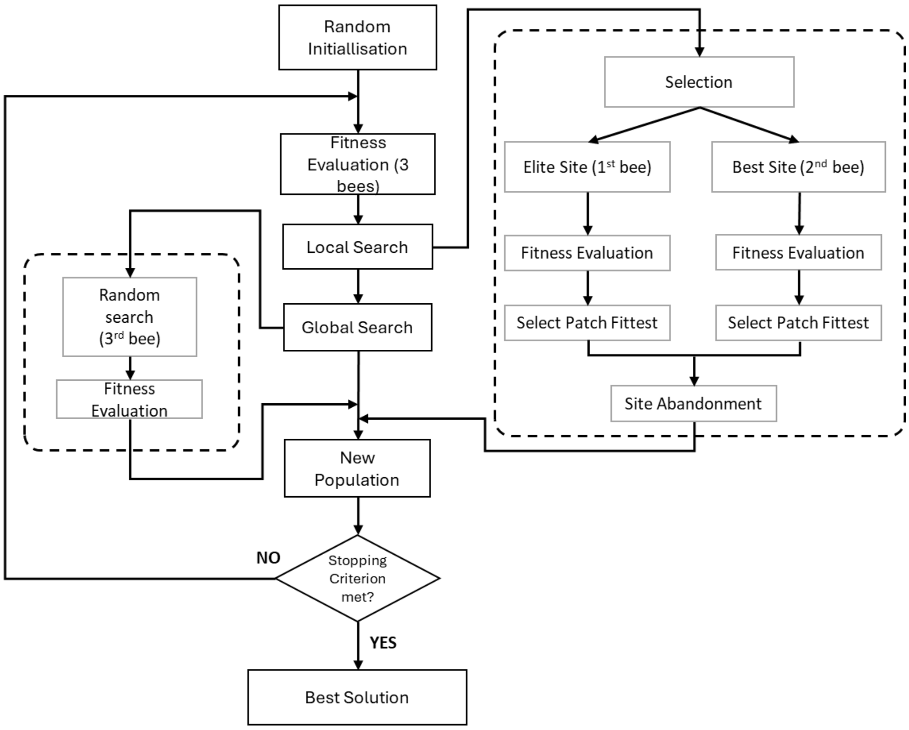


Figure 2. Ternary Bees Algorithm ($BA_{Ternary}$)

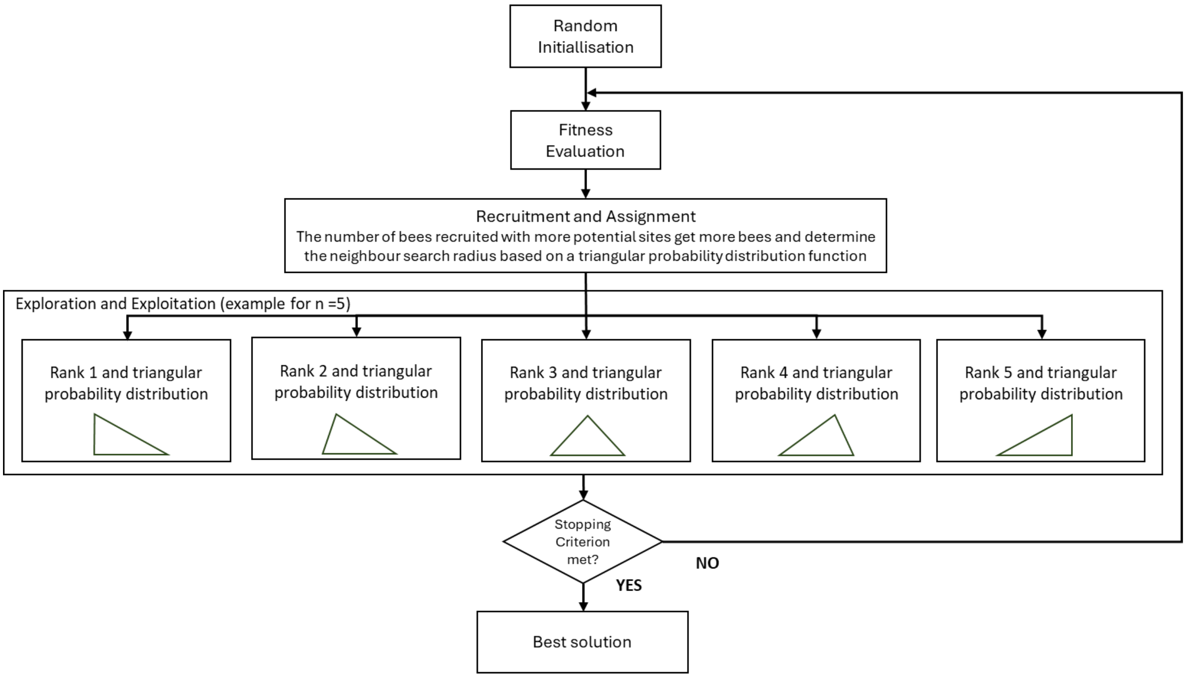


Figure 3. Two-parameter Bees Algorithm ($BA_2$)

$BA_F$ is designed by Hartono to operate in both combinatorial and continuous domains, with its initial implementation focused on addressing robotic disassembly sequence problems (Hartono *et al.*, 2023). Subsequently, the continuous version of $BA_F$ was tested on benchmark functions and various engineering problems (Suluova, Hartono and Pham, 2023). Notably, $BA_F$ has demonstrated superior performance over basic BA in addressing robotic disassembly sequence problems and has shown its ability to identify optimal solutions within the continuous domain. In addition, enhanced $BA_F$ for bus routing problems also shows better performance compared to enhanced BA (Zhao *et al.*, 2023).

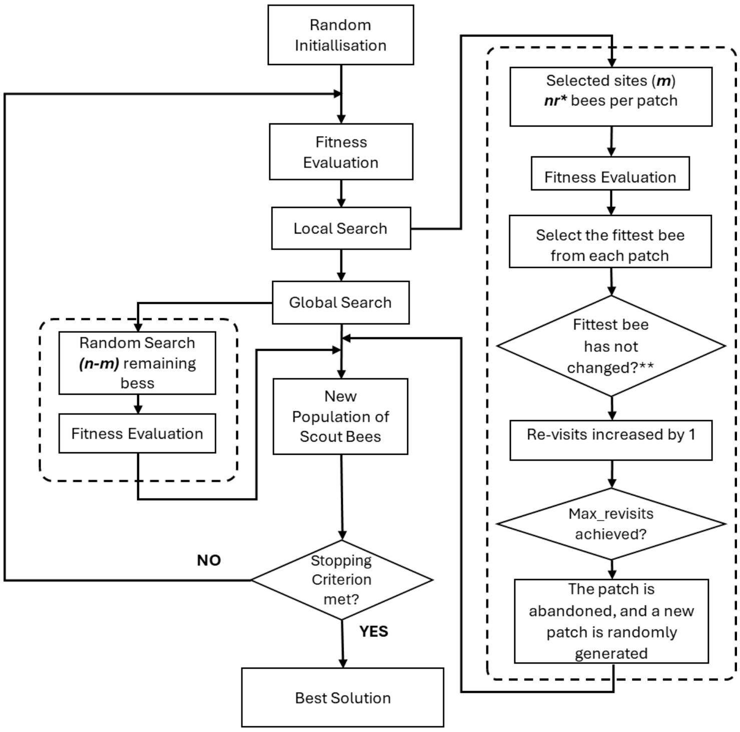


Figure 4. Fibonacci Bees Algorithm ($BA_F$)

$BA_F$ was inspired by the Fibonacci sequence-derived family tree structure in drones (male bees) to calculate the quantity of bees dispatched to flower patches. This approach based on ranking seeks to optimise foraging efficiency by effectively utilising the most promising patches. $BA_F$ concentrates on selected sites, removing the necessity to differentiate between 'elite' and 'other selected' sites. The highest-ranked bee is assigned the highest number from the specified (maximum) Fibonacci sequence, while lower-ranked bees receive decreasing numbers from the Fibonacci sequence. More bees are assigned to flower patches of higher quality, resulting in differential allocation. The number of bees recruited for selected sites using ranking-based recruitment (*nr*) is set to zero, and a new set of patches is initialised if the fittest bee in a given patch remains unchanged after reaching the maximum number of re-visits (*max_rv*).

## 3. Results and Discussion

The experiment comprises four steps, which are illustrated in Figure 5. To ensure a fair comparison, the parameter setting for each version of BA are as follows. The optimal parameter settings for basic BA in TSP were reported in Hartono et al., where $n$ = 40, $m$ = 20, $e$ =8, $nsp$ = 10, and $nep$ =40 (Hartono *et al.*, 2023). The $BA_{Ternary}$ has only 1 parameter to set, which is the site abandonment, and since it has never been used in TSP applications, the experiment was conducted to find the best parameters. The stagnated size before the site is abandoned started at 10 and increased by 10 until 2000, with optimal settings found in 1500. $BA_2$ has only 2 parameters to set, and these parameters follow the settings from Ismail: $n$ = 30 and $nep$ = 30.

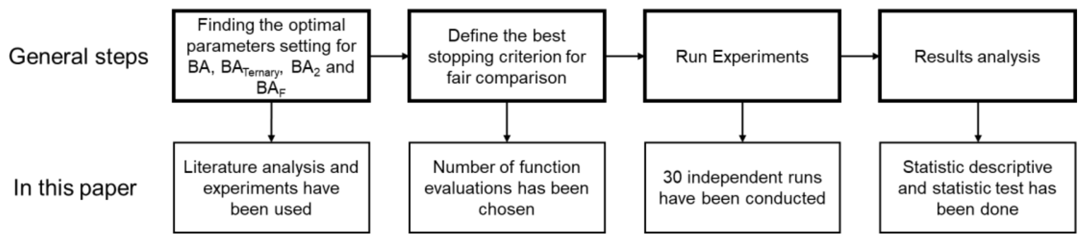


Figure 5. Experiment outline

The $BA_F$ follows the design of experiment steps conducted by Hartono (Hartono, 2023; Hartono and Pham, 2024). The best parameter settings of *n* scoutbees, *m* selected sites, maximum Fibonacci number for ranking-based recruitment (*maxnr*), and maximum number of re-visits before the *nr* is set to zero (*max_rv*) are 96, 20, 144, and 100, respectively. The population size of BA and $BA_{Ternary}$ is calculated using Equation 1, $BA_2$ using Equation 2, and $BA_F$ using Equation 3. The population sizes are as follows: BA and $BA_F$ = 460, $BA_2$ = 465, and $BA_{Ternary}$ = 3.

$$BA_{population} = (e * nep) + ((m - e) * nsp) + (n - m) \quad (1)$$

$$\sum_{k=1}^{n} \left| nep - (k-1)\left(\frac{nep-1}{n-1}\right) \right| \quad (2)$$

where $k$ = rank of the worker bees in $BA_2$

$$BA_F = (n - m) + \sum_{i=1}^{m}(nr_i) \quad (3)$$

Researchers often use two primary stopping criteria: the maximum number of iterations or the number of function evaluations (NFE). If the population sizes are equal, employing the maximum iteration as a stopping criterion would be appropriate. Considering the limitation of $BA_{Ternary}$, which only allows for three bees, the maximum iteration is not suitable as a stopping criterion. For this research, the chosen stopping criterion is NFE, set at 2 million. The value was discovered through experimentation, beginning with 1 million NFE and gradually increasing by 200,000 until reaching 3 million NFE. When the algorithms stabilised and approached the optimum value, the nearest number was selected. The algorithms were then performed for 30 independent runs.

Table 1 presents the best-found value (BFV) and the best-known solution (BKS) from the experiments, where a smaller value indicates better performance as TSP aims to minimise the route. The deviation of BFV from BKS ($D_{best}$) is calculated using Equation 4, which measures the percentage difference between BFV and BKS. A smaller deviation indicates better performance, demonstrating the accuracy of the result. Similarly, the deviation of the average value ($D_{avg}$) demonstrates the average accuracy of the results (see Equation 5).

$$D_{best} = \frac{(BFV - BKS)}{BKS} x100\% \quad (4)$$

$$D_{avg} = \frac{(Average\ value - BKS)}{BKS} x100\% \quad (5)$$

$BA_F$ demonstrates similar performance to BA in finding the best-found value (near-optimal solutions) for datasets with up to 150 cities. However, $BA_F$ consistently outperforms BA in most datasets, except for KroA100 and KroA150, where BA achieves better results. For datasets with fewer than 100 cities, $BA_F$ shows a deviation from the BKS below 0.86%, compared to 1.12% for BBA, 9% for $BA_{Ternary}$, and 6.28% for $BA_2$. In datasets with cities ranging from 150 to 200, $BA_F$'s deviation ranges from 1.77% to 5.11%, while BA's ranges from 1.31% to 7.96%. The other versions of BA exceed 14.02% in this range. Overall, $BA_F$ outperforms BBA, $BA_{Ternary}$, and $BA_2$ in terms of finding the best-found value and the average over 30 independent runs, particularly for datasets with more than 150 cities.

Table 2 indicates that the average performance of BA and $BA_F$ in finding near-optimal solutions for datasets with fewer than 150 cities is comparable, with deviations ranging from 1.29% to 5.10%. However, for datasets with more than 200 cities, $BA_F$ outperforms other versions of BA. Furthermore, the average results for $BA_F$ are lower than those for other versions of BA across all datasets, as shown in Table 2.

The descriptive statistics clearly indicate that $BA_F$ outperforms the other versions for larger datasets. Hence, conducting statistical tests to determine significant differences is unnecessary for datasets with more than 200 cities. Since it is evident from the descriptive statistics that BA and $BA_F$ are similar, a statistical test is conducted to compare these two versions of BA. Therefore, the final step is to demonstrate the statistical significance of the differences between BA and $BA_F$. The following steps, as shown in Figure 6 and suggested by Hartono (Hartono, Ramírez, & Pham, 2022), are implemented with an additional step. As the independent *t*-test compares only two groups, a post-hoc test is unnecessary.

The first step involves testing for normality, with the results presented in Table 3. The p-value, which is greater than 0.05, provides strong evidence for accepting the null hypothesis and indicates that the variable follows a normal distribution.

Table 1. Experiment results (best-found value)

| Dataset | BKS | BA | | $BA_{Ternary}$ | | $BA_2$ | | $BA_F$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | BFV | $D_{best}$ | BFV | $D_{best}$ | BFV | $D_{best}$ | BFV | $D_{best}$ |
| Eil76 | 538 | 544 | 1.12% | 552 | 2.60% | 548 | 1.86% | **541** | **0.56%** |
| Rat99 | 1211 | 1224 | 1.07% | 1319 | 8.92% | 1287 | 6.28% | **1217** | **0.50%** |
| KroA100 | 21282 | **21292** | **0.05%** | 22894 | 7.57% | 21606 | 1.52% | 21379 | 0.46% |
| KroB100 | 22141 | 22300 | 0.72% | 22408 | 1.21% | 22771 | 2.85% | **22179** | **0.17%** |
| KroC100 | 20749 | 20855 | 0.51% | 22278 | 7.37% | 21364 | 2.96% | **20769** | **0.10%** |
| KroD100 | 21294 | 21459 | 0.77% | 23211 | 9.00% | 22189 | 4.20% | **21478** | **0.86%** |
| KroE100 | 22068 | 22116 | 0.22% | 23563 | 6.77% | 23062 | 4.50% | **22112** | **0.20%** |
| KroA150 | 26524 | **27313** | **2.97%** | 30308 | 14.27% | 31877 | 20.18% | 27364 | 3.17% |
| KroB150 | 26130 | 26473 | 1.31% | 29793 | 14.02% | 30258 | 15.80% | **26593** | **1.77%** |
| KroA200 | 29368 | 31321 | 6.65% | 35269 | 20.09% | 42943 | 46.22% | **30569** | **4.09%** |
| KroB200 | 29437 | 31780 | 7.96% | 33872 | 15.07% | 42000 | 42.68% | **30941** | **5.11%** |
| Lin318 | 42029 | 52149 | 24.08% | 59105 | 40.63% | 98915 | 135.35% | **45578** | **8.44%** |
| Pcb442 | 50778 | 76308 | 50.28% | 79708 | 56.97% | 155957 | 207.13% | **57158** | **12.56%** |

Notes: BKS = Best Known Solution; BFV = Best Found Value; $D_{best}$ = deviation of BFV from BKS, bold indicates the best-found value.

Tabl**e** 2. Experiment results (average)

| Dataset | BKS | BA | | $BA_{Ternary}$ | | $BA_2$ | | $BA_F$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Average | $D_{avg}$ | Average | $D_{avg}$ | Average | $D_{avg}$ | Average | $D_{avg}$ |
| Eil76 | 538 | 551.7 | 2.54% | 595.1 | 10.61% | 562.3 | 4.52% | **551.0** | **2.42%** |
| Rat99 | 1211 | **1254.3** | **3.58%** | 1421.2 | 17.36% | 1323.5 | 9.29% | 1256.2 | 3.73% |
| KroA100 | 21282 | **21556.2** | **1.29%** | 24719.4 | 16.15% | 22653.9 | 6.45% | 21671.2 | 1.83% |
| KroB100 | 22141 | **22659.5** | **2.34%** | 25001.7 | 12.92% | 23664.7 | 6.88% | 22748.2 | 2.74% |
| KroC100 | 20749 | 21189.5 | 2.12% | 24649.5 | 18.80% | 22378.6 | 7.85% | **21181.5** | **2.08%** |
| KroD100 | 21294 | **21771.0** | **2.24%** | 24787.1 | 16.40% | 22744.8 | 6.81% | 21858.9 | 2.65% |
| KroE100 | 22068 | **22458.8** | **1.77%** | 25218.7 | 14.28% | 23487.3 | 6.43% | 22471.0 | 1.83% |
| KroA150 | 26524 | **27829.6** | **4.92%** | 32241.4 | 21.56% | 33311.0 | 25.59% | 27877.4 | 5.10% |
| KroB150 | 26130 | 27397.0 | 4.85% | 32158.5 | 23.07% | 32676.9 | 25.06% | **27255.9** | **4.31%** |
| KroA200 | 29368 | 31974.5 | 8.88% | 39085.5 | 33.09% | 45725.7 | 55.70% | **31325.6** | **6.67%** |
| KroB200 | 29437 | 32323.0 | 9.80% | 39362.9 | 33.72% | 45952.5 | 56.10% | **31588.2** | **7.31%** |
| Lin318 | 42029 | 53965.8 | 28.40% | 65184.3 | 55.09% | 104385.9 | 148.37% | **46877.4** | **11.54%** |
| Pcb442 | 50778 | 79428.8 | 56.42% | 87145.0 | 71.62% | 164189.7 | 223.35% | **59400.4** | **16.98%** |

Notes: BKS = Best Known Solution; $D_{avg}$ = deviation of average result from BKS, bold indicates the best average result.

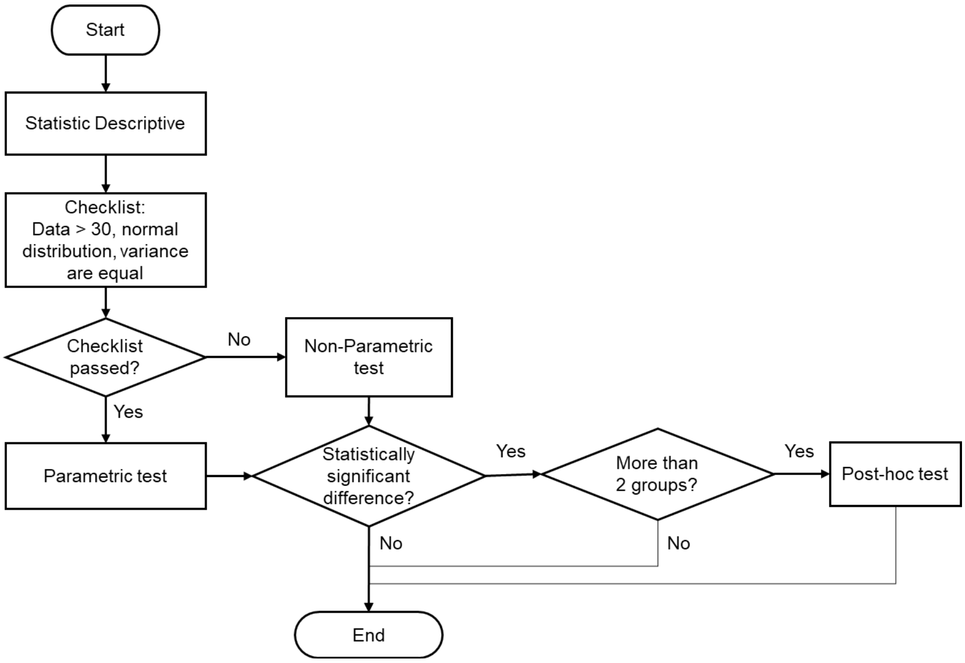


Figure 6. Statistic test decision in this research (modified from Hartono, Ramírez and Pham (2022))

Table 3. SPSS output of Test of Normality

| Tests of Normality | | | | |
|---|---|---|---|---|
| | Algorithm | Kolmogorov-Smirnov[a] | | |
| | | Statistic | df | Sig. |
| Eil76 | BAF | 0.107 | 30 | .200* |
| | BA | 0.147 | 30 | 0.095 |
| Rat99 | BAF | 0.104 | 30 | .200* |
| | BA | 0.125 | 30 | .200* |
| KroA100 | BAF | 0.078 | 30 | .200* |
| | BA | 0.111 | 30 | .200* |
| KroB100 | BAF | 0.128 | 30 | .200* |
| | BA | 0.122 | 30 | .200* |
| KroC100 | BAF | 0.114 | 30 | .200* |
| | BA | 0.144 | 30 | 0.115 |
| KroD100 | BAF | 0.124 | 30 | .200* |
| | BA | 0.093 | 30 | .200* |
| KroE100 | BAF | 0.07 | 30 | .200* |
| | BA | 0.135 | 30 | 0.171 |
| KroA150 | BAF | 0.111 | 30 | .200* |
| | BA | 0.115 | 30 | .200* |
| KroB150 | BAF | 0.127 | 30 | .200* |
| | BA | 0.135 | 30 | 0.175 |
| KroA200 | BAF | 0.13 | 30 | .200* |
| | BA | 0.136 | 30 | 0.168 |
| KroB200 | BAF | 0.073 | 30 | .200* |
| | BA | 0.088 | 30 | .200* |
| Lin318 | BAF | 0.138 | 30 | 0.153 |
| | BA | 0.098 | 30 | .200* |
| Pcb442 | BAF | 0.114 | 30 | .200* |
| | BA | 0.126 | 30 | .200* |

* This is a lower bound of the true significance.
[a] Lilliefors Significance Correction

Therefore, a parametric test, an independent t-test, is conducted to compare the means of BA and $BA_F$, with the null hypothesis ($H_0$) stating that there is no significant difference between the two means and the alternative hypothesis ($H_1$) suggesting that there is a significant difference. Figure 7 shows that the Levene's test for homogeneity of variances test results indicate that the population variances of BA and $BA_F$ are equal. Therefore, the row of equal variances assumed for the *t*-test is checked. The results indicate that for Eil76, Rat99, KroB100, KroC100, KroD100, KroE100, KroA150, and KroB150, there is no significant difference in the means of BA and $BA_F$, as indicated by $p > 0.05$. However, significant differences are observed for KroA100, KroA200, KroB200, Lin318, and Pcb442 ($p < 0.001$).

The statistical results indicate that for datasets with less than 150 cities, there is no statistical difference between BA and $BA_F$, except for KroA100, where BA shows better performance. However, for cities with more than 200, $BA_F$ shows a statistically significant difference from BA. The descriptive statistics show that the deviation (average) from the best-known solution ($D_{avg}$) for $BA_F$ is 6.67-7.31% for 200 cities, compared to 8.88-9.80% for BA. For 318 cities, the $D_{avg}$ for $BA_F$ is 11.54%, while for BA it is 28.40%. Similarly, for 442 cities, $D_{avg}$ for $BA_F$ is 16.98%, while for BA it is 56.42%. These findings support the conclusion that $BA_F$ outperforms BA on larger datasets.

The comparison of BA, $BA_{Ternary}$, $BA_2$, and $BA_F$ reveals that $BA_F$ outperformed all versions of BA in TSP. This is consistent with previous findings by Hartono, where the $BA_F$ outperformed the BA in disassembly problems (Hartono, 2023; Hartono and Pham, 2024). It is evident that smaller parameter settings did not guarantee the highest performance; rather, they simply made parameter adjustment easier.

$BA_{Ternary}$'s key strength lies in its efficiency, being the fastest version of BA with only one parameter setting, making it particularly useful for real-time planning or fast decision-making. As a result, $BA_{Ternary}$ has been employed in training hyperparameters in deep learning (Zeybek, 2023; Kumar *et al.*, 2024). The main benefit of $BA_2$ is its simplicity, as it only requires the configuration of two parameters. Despite the lower performance of the basic version of $BA_2$ in comparison to other versions, the incorporation of bee traplining operators, such as the bees routing optimizer (BRO), resulted in $BA_2$ outperforming BA in terms of performance (Ismail, 2021).

**Independent Samples Test**

| | | Levene's Test for Equality of Variances | | t-test for Equality of Means | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Significance | | | | 95% Confidence Interval of the Difference | |
| | | F | Sig. | t | df | One-Sided p | Two-Sided p | Mean Difference | Std. Error Difference | Lower | Upper |
| Eil76 | Equal variances assumed | 2.522 | .118 | 1.152 | 58 | .127 | .254 | 1.00000 | .86790 | -.73730 | 2.73730 |
| | Equal variances not assumed | | | 1.152 | 54.015 | .127 | .254 | 1.00000 | .86790 | -.74003 | 2.74003 |
| Rat99 | Equal variances assumed | 1.451 | .233 | -.540 | 58 | .296 | .592 | -1.86667 | 3.45953 | -8.79167 | 5.05834 |
| | Equal variances not assumed | | | -.540 | 56.199 | .296 | .592 | -1.86667 | 3.45953 | -8.79640 | 5.06306 |
| KroA100 | Equal variances assumed | .972 | .328 | -2.404 | 58 | .010 | .019 | -114.96667 | 47.83241 | -210.71363 | -19.21970 |
| | Equal variances not assumed | | | -2.404 | 55.893 | .010 | .020 | -114.96667 | 47.83241 | -210.79055 | -19.14278 |
| KroB100 | Equal variances assumed | .985 | .325 | .553 | 58 | .291 | .582 | 26.13333 | 47.21721 | -68.38217 | 120.64884 |
| | Equal variances not assumed | | | .553 | 55.738 | .291 | .582 | 26.13333 | 47.21721 | -68.46391 | 120.73058 |
| KroC100 | Equal variances assumed | .365 | .548 | .162 | 58 | .436 | .872 | 8.00000 | 49.28930 | -90.66326 | 106.66326 |
| | Equal variances not assumed | | | .162 | 57.993 | .436 | .872 | 8.00000 | 49.28930 | -90.66353 | 106.66353 |
| KroD100 | Equal variances assumed | .001 | .978 | -.542 | 58 | .295 | .590 | -28.86667 | 53.29054 | -135.53927 | 77.80594 |
| | Equal variances not assumed | | | -.542 | 57.936 | .295 | .590 | -28.86667 | 53.29054 | -135.54178 | 77.80845 |
| KroE100 | Equal variances assumed | .008 | .931 | -.931 | 58 | .178 | .356 | -39.60000 | 42.52753 | -124.72811 | 45.52811 |
| | Equal variances not assumed | | | -.931 | 57.500 | .178 | .356 | -39.60000 | 42.52753 | -124.74387 | 45.54387 |
| KroA150 | Equal variances assumed | .440 | .510 | -.678 | 58 | .250 | .501 | -47.73333 | 70.43809 | -188.73048 | 93.26382 |
| | Equal variances not assumed | | | -.678 | 57.287 | .250 | .501 | -47.73333 | 70.43809 | -188.76785 | 93.30118 |
| KroB150 | Equal variances assumed | 2.354 | .130 | 1.682 | 58 | .049 | .098 | 142.56667 | 84.75587 | -27.09064 | 312.22398 |
| | Equal variances not assumed | | | 1.682 | 53.722 | .049 | .098 | 142.56667 | 84.75587 | -27.37877 | 312.51210 |
| KroA200 | Equal variances assumed | 4.847 | .032 | 7.269 | 58 | <.001 | <.001 | 648.96667 | 89.27275 | 470.26784 | 827.66549 |
| | Equal variances not assumed | | | 7.269 | 53.151 | <.001 | <.001 | 648.96667 | 89.27275 | 469.92011 | 828.01322 |
| KroB200 | Equal variances assumed | .015 | .903 | 9.853 | 58 | <.001 | <.001 | 734.76667 | 74.56921 | 585.50017 | 884.03316 |
| | Equal variances not assumed | | | 9.853 | 57.995 | <.001 | <.001 | 734.76667 | 74.56921 | 585.49990 | 884.03343 |
| Lin318 | Equal variances assumed | 1.843 | .180 | 39.220 | 58 | <.001 | <.001 | 7088.43333 | 180.73373 | 6726.65547 | 7450.21119 |
| | Equal variances not assumed | | | 39.220 | 54.479 | <.001 | <.001 | 7088.43333 | 180.73373 | 6726.15694 | 7450.70973 |
| Pcb442 | Equal variances assumed | 2.470 | .122 | 78.933 | 58 | <.001 | <.001 | 20028.43333 | 253.73913 | 19520.51929 | 20536.34738 |
| | Equal variances not assumed | | | 78.933 | 56.276 | <.001 | <.001 | 20028.43333 | 253.73913 | 19520.18771 | 20536.67895 |

Figure 7. SPSS output of independent *t*-test

Compared to $BA_{Ternary}$ and $BA_2$, $BA_F$ requires time for parameter adjustment due to its four parameters; however, it is the best-performing version of BA for TSP. The choice of which version of BA to use depends on the researcher's objectives, whether to prioritise the decision's speed, minimise the number of parameter settings, or achieve the best results. There is still ample opportunity for further improvement in the $BA_F$ version.

## 4. Conclusion

The results demonstrate that $BA_F$, with four parameter settings, outperformed BA, $BA_{Ternary}$, and $BA_2$, which require five, one, and two parameter settings, respectively. These findings suggest a trade-off between performance and the choice of algorithm version. Furthermore, the $BA_F$ outperformed other versions in its basic form, suggesting potential for further improvement by other researchers. The $BA_F$ is still in its basic version, and various improvements could be made to this newest version of BA. The $BA_F$ code for TSP is shared on GitHub, and other researchers are encouraged to use it with proper attribution to this paper and the code. In addition, $BA_2$, with its improvements using numerous strategies such as BRO, shows good results for both continuous and combinatorial versions. The integration of these enhancements into the $BA_F$ version for comparative analysis may yield valuable insights. The application of $BA_F$ in robotic disassembly sequences, continuous benchmark functions, engineering problems, bus routing problems, and TSP highlights its versatility and potential for further exploration. Future research will focus on expanding the application of $BA_F$ to other applications and further improving the algorithm.

## Acknowledgement

The authors wish to express their gratitude to Professor D.T. Pham, University of Birmingham, for his support with this study.

## Appendix

The following GitHub link https://github.com/NataliaHartonoFung/Fibonacci-Bees-Algorithm-TSP is provided to access the $BA_F$ code.

## References

Alorf, A. (2023) 'A survey of recently developed metaheuristics and their comparative analysis', *Engineering Applications of Artificial Intelligence*, 117, p. 105622. Available at: https://doi.org/10.1016/j.engappai.2022.105622.

Castellani, M. and Pham, D.T. (2023) 'The Bees Algorithm—a gentle introduction', in D.T. Pham and N. Hartono (eds) *Intelligent production and manufacturing optimisation—the Bees Algorithm approach*. 1st edn. Cham: Springer Nature, pp. 3–21. Available at: https://doi.org/10.1007/978-3-031-14537-7_1.

Hartono, N. (2023) *Intelligent robotic disassembly optimisation for sustainability using the Bees Algorithm*. Doctoral Thesis. University of Birmingham. Available at: https://etheses.bham.ac.uk/id/eprint/14384/ (Accessed: 12 December 2023).

Hartono, N. *et al.* (2023) 'Parameter tuning for combinatorial Bees Algorithm in Travelling Salesman Problems', in *13th International Seminar on Industrial Engineering and Management*. Bandung: AIP Conference Proceedings, p. 090005. Available at: https://doi.org/10.1063/5.0106177.

Hartono, N. and Pham, D.T. (2024) 'A novel Fibonacci-inspired enhancement of the Bees Algorithm: application to robotic disassembly sequence planning', *Cogent Engineering*, 11(1). Available at: https://doi.org/10.1080/23311916.2023.2298764.

Hartono, N., Ramírez, F.J. and Pham, D.T. (2022) 'Optimisation of robotic disassembly plans using the Bees Algorithm', *Robotics and Computer-Integrated Manufacturing*, 78, p. 102411. Available at: https://doi.org/10.1016/j.rcim.2022.102411.

Hussain, K. *et al.* (2019) 'Metaheuristic research: a comprehensive survey', *Artificial Intelligence Review*, 52(4), pp. 2191–2233. Available at: https://doi.org/10.1007/s10462-017-9605-z.

Hussein, W.A., Sahran, S. and Sheikh Abdullah, S.N.H. (2017) 'The variants of the Bees Algorithm (BA): a survey', *Artificial Intelligence Review*, 47(1), pp. 67–121. Available at: https://doi.org/10.1007/s10462-016-9476-8.

Ismail, A.H. *et al.* (2020) 'Using the Bees Algorithm to solve combinatorial optimisation problems for TSPLIB', *IOP Conference Series: Materials Science and Engineering*, 847(1), p. 012027. Available at: https://doi.org/10.1088/1757-899X/847/1/012027.

Ismail, A.H. (2021) *Enhancing the Bees Algorithm using the traplining metaphor*. Doctoral Thesis. University of Birmingham. Available at: https://etheses.bham.ac.uk/id/eprint/12128/ (Accessed: 29 December 2023).

Ismail, A.H. (2022) *BA_C2*, *GitHub*. Available at: https://github.com/asrulharunismail/BA_C2 (Accessed: 4 October 2023).

Juan, A.A. *et al.* (2023) 'A review of the role of heuristics in stochastic optimisation: from metaheuristics to learnheuristics', *Annals of Operations Research*, 320(2), pp. 831–861. Available at: https://doi.org/10.1007/s10479-021-04142-9.

Kumar, A. *et al.* (2024) 'Bees Algorithm for hyperparameter search with deep learning to estimate the remaining useful life of ball bearings', in D.T. Pham and N. Hartono (eds) *Intelligent engineering optimisation with the Bees Algorithm*. Cham: Springer.

Laili, Y. *et al.* (2019) 'Robotic disassembly re-planning using a two-pointer detection strategy and a super-fast Bees Algorithm', *Robotics and Computer-Integrated Manufacturing*, 59, pp. 130–142. Available at: https://doi.org/10.1016/j.rcim.2019.04.003.

Liu, J. *et al.* (2018) 'Robotic disassembly sequence planning using enhanced discrete Bees Algorithm in remanufacturing', *International Journal of Production Research*, 56(9), pp. 3134–3151. Available at: https://doi.org/10.1080/00207543.2017.1412527.

Pham, D.T. *et al.* (2005) *Bee Algorithm - a novel approach to function optimisation*. Cardiff.

Pham, D.T. *et al.* (2007) 'Using the Bees Algorithm to schedule jobs for a machine', in *Proc Eighth International Conference on Laser Metrology*. Cardiff: CMM and Machine Tool Performance.

Pham, D.T., Afify, A. and Koç, E. (2007) 'Manufacturing cell formation using the Bees Algorithm', in *Innovative Production Machines and Systems Virtual Conference*. Cardiff.

Pham, D.T. and Castellani, M. (2009) 'The Bees Algorithm: Modelling foraging behaviour to solve continuous optimization problems', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(12), pp. 2919–2938. Available at: https://doi.org/10.1243/09544062JMES1494.

Pop, P.C. *et al.* (2024) 'A comprehensive survey on the generalized traveling salesman problem', *European Journal of Operational Research*, 314(3), pp. 819–835. Available at: https://doi.org/10.1016/j.ejor.2023.07.022.

Sarhani, M., Voß, S. and Jovanovic, R. (2023) 'Initialization of metaheuristics: comprehensive review, critical analysis, and research directions', *International Transactions in Operational Research*, 30(6), pp. 3361–3397. Available at: https://doi.org/10.1111/itor.13237.

Sörensen, K. and Glover, F. (2013) 'Metaheuristics', *Encyclopedia of Operations Research And Management Science*, 62, pp. 960–970.

Suluova, H.F., Hartono, N. and Pham, D.T. (2023) 'The Fibonacci Bees Algorithm for continuous optimisation problems – some engineering applications', *The International Workshop of the Bees Algorithm and its Applications (BAA) 2023* [Preprint]. Birmingham.

Thong-ia, S. and Champrasert, P. (2023) 'Gene-Ants: Ant Colony Optimization with Genetic Algorithm for Traveling Salesman Problem Solving', in *2023 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)*. IEEE, pp. 1–5. Available at: https://doi.org/10.1109/ITC-CSCC58803.2023.10212945.

Toaza, B. and Esztergár-Kiss, D. (2023) 'A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems', *Applied Soft Computing*, 148, p. 110908. Available at: https://doi.org/10.1016/j.asoc.2023.110908.

Zeybek, S. (2023) 'Prediction of the remaining useful life of engines for remanufacturing using a semi-supervised deep learning model trained by the Bees Algorithm', in D.T. Pham and N. Hartono (eds)

*Intelligent production and manufacturing optimisation—the Bees Algorithm approach*. Cham: Springer, pp. 383–397. Available at: https://doi.org/10.1007/978-3-031-14537-7_21.

Zhao, F. *et al.* (2023) ‘Enhanced Bees Algorithm for customised bus routing problem considering multi-level services’, *The International Workshop of the Bees Algorithm and its Applications (BAA) 2023* [Preprint]. Birmingham.