A critical analysis of the bat algorithm

Iannick Gagnon© | Alain April | Alain Abran

Software and Information Technology Engineering Department, École de technologie supérieure, Montreal, Canada

Correspondence
Iannick Gagnon, Software and Information Technology Engineering Department, École de technologie supérieure, 1100 Notre-Dame W., H3C 1K3, Canada.
Email: iannick.gagnon.1@ens.etsmtl.ca

Abstract
This article presents an analysis of the bat algorithm (BA) based on elementary mathematical analysis and statistical comparisons of the first hitting time performance metric distributions obtained on a test set comprising five carefully selected objective functions. The findings show that the BA is not an original contribution to the metaheuristics literature and that it is not generally superior to the Particle Swarm Optimization algorithm when fair comparisons are made. It is also shown that some components of the BA can be either replaced by simpler alternatives or be removed entirely to increase performance. Finally, the results suggest that the best version of the BA is in fact a simple hybrid between Particle Swarm Optimization and Simulated Annealing. To encourage more transparency in metaheuristics research, the entirety of the MATLAB code used in this article is available in a GitHub repository for suggestions and/or corrections.

KEYWORDS
bat algorithm, Metaheuristics, particle swarm optimization

1 | INTRODUCTION

Metaheuristics are universal stochastic optimization algorithms used to solve problems where objective functions are unknown or infeasible to solve with limited computational resources. Such problems include the Quadratic Assignment and Traveling Salesman problems which are well known in the Operations Research community. More recent examples include the training of convolutional neural networks and on-line load sharing management for web servers among others. This article compares and constrasts two specific metaheuristics, namely the bat algorithm (BA) and Particle Swarm Optimization (PSO).

The bat algorithm is proposed in Reference 3, as a novel metaheuristic inspired by the echolocation mechanism of bats. The original article makes two main claims. The first is that BA is a novel metaheuristic and the second is that BA is generally much superior to Particle Swarm Optimization and Genetic Algorithms (GA) based on experiments with 10 benchmark functions. There is no a priori reason to question the bat metaphor given that other metaheuristics like Ant Colony Optimization (ACO) really are inspired by biology. The scientific literature on bat echolocation does not discredit the metaphor, but it shows that it is shallow. BA parameterizes true bat echolocation based on frequency \( f \), pulse rate \( r \) and loudness \( A \) while real bats use the former plus duration, directionality and a complex motor control system to home in on prey. To be fair, the fact that BA is not a perfect model of true bat behaviour is not a problem per se, but if the research community accepts metaphorical descriptions, it should insist that other possible...
analogies be differentiated. Case in point, bats are not the only species that use echolocation for navigation and hunting. Some bird species employ a similar strategy and so do certain species of dolphins and whales. Any one of these animals could have been chosen to create the Whale Algorithm for echolocation-based optimization instead of BA. This creates unnecessary confusion since Whale Optimization Algorithm (WOA)\(^8\) does exist and it is not based on echolocation. To repeat one of Kenneth Sörensen’s points in Reference 9 the language of metaphors obscures what is truly novel about an algorithm other than its name and says very little if anything about why it works. The reliance on metaphors instead of mathematical descriptions is but one of the methodological flaws commonly observed in the metaheuristics literature.

The purpose of this article is to show that inadequate research methodology has led the research community to believe that BA is an original contribution to the metaheuristics literature. The research hypothesis is that BA is an underperforming variant of PSO instead of the other way around. The reason for singling out BA over other metaheuristics is that BA has obtained more than 10 000 citations since it was published in 2010 as is reported in Web of Science as of November 2019. This makes BA a good candidate for demonstrating that the growing concern that bad methodology is widespread in the metaheuristics research literature\(^9-14\) is in fact supported by evidence obtained by following good methodological practices.

The rest of this article is organized as follows: Sections 2 and 3 describe PSO and BA respectively, Section 4 contains a detailed mathematical analysis of BA, Section 5 contains a detailed description of the methodology used to generate and analyse the results of benchmark function minimization discussed in Section 6 and finally Section 7 contains a summary of the findings and proposes future research avenues.

2 THE PARTICLE SWARM OPTIMIZATION METAHEURISTIC

In historical terms, PSO is the third swarm intelligence metaheuristic after Stochastic Diffusion Search (SDS)\(^15\) in 1989 and ACO\(^5\) in 1992. It is relevant to our discussion to point out that it precedes BA by a decade and a half. PSO was designed by social psychologist James Kennedy and electrical engineer Russell Eberhart.4 The initial purpose was to simulate the social behavior of humans through the imagery of the movements of bird flocks and fish schools. It was quickly realized that the resulting paradigm was well-suited for mathematical optimization and thereafter was used to optimize highly non-convex functions and neural network training.

Kennedy and Eberhart did not use names like Bird Flock Optimization and Fish School Optimization precisely because they systematically simplified PSO by removing some of the original nature-inspired components of the algorithm and found that it worked just as well. As such, they decided to use the neutral imagery of particles as defined in Reference 16 and 17. Others like Reference 18 with Migrating Birds Optimization (MBO) and Reference 19 with Fish School Search (FSS) were apparently not as concerned with the impact of using metaphors.

The characteristic equations of PSO are given in Algorithm 1. The position \((\mathbf{x}_i^t)\) at time step \(t\) of the \(i\)-th particle are its coordinates in the search space and the velocities \((\mathbf{v}_i^t)\) are a weighed sum of the relative position vectors between the particles, their personal historical best positions \((\mathbf{p}_i^t)\) and the current global best position \((\mathbf{x}^*)\). The velocity term is weighed by what is called the inertia factor \((\omega)\). The effect of inertia is to accumulate velocity in a “good” direction over time. The primary difference with BA is the use of the \(i\)-th particle's personal best-known location. The parameters \(c_1\) and \(c_2\) are constants that weigh the relative importance between the global best and the personal best positions, respectively. The parameters \(r_1\) and \(r_2\) are random variables drawn from a uniform distribution generally with support \([0, 1]\) represented by \(U(0, 1)\). The randomness introduced by these parameters causes the particles to overshoot global best and personal best positions with proportions \(1/c_1\) and \(1/c_2\) respectively when \(t\) is large. The net effect is to prevent premature convergence by providing what Kennedy and Eberhart called a stochastic “kick” in Reference 4. The key idea behind PSO is that at every time step, the particle is pushed—accelerated in PSO terminology—in the direction of a weighted average of the particle's personal best-known location and the entire swarm's best-known position. The implied assumption is that regions of high fitness are close to other regions of high fitness.\(^20\)

There are hundreds of variants of PSO some of which were proposed by the original authors. The original PSO given in Reference 3 is the one used in this article despite there being a standard reference PSO algorithm proposed in Reference 21. The reason for this choice is that the authors believe that BA is only marginally different from the original PSO, therefore Original PSO is a more appropriate basis of comparison.
Algorithm 1. Pseudocode for minimization with PSO

1. initialize positions, velocities, global and personal best positions
2. while $t < t_{\text{max}}$ do
   3. for $i = 1$ to number of particles do
      4. $v_{i}^{t+1} \leftarrow \omega v_{i}^{t} + c_{1} r_{1}(x^{*} - x_{i}^{t}) + c_{2} r_{2}(p_{i}^{t} - x_{i}^{t})$
      5. $x_{i}^{t+1} \leftarrow x_{i}^{t} + v_{i}^{t+1}$
      6. if $x_{i}^{t+1} < x^{*}$ then
         7. $x^{*} \leftarrow x_{i}^{t+1}$
      end if
      8. if $x_{i}^{t+1} < p_{i}$ then
         9. $p_{i} \leftarrow x_{i}^{t+1}$
      end if
   10. end for
3. end while
4. return $x^{*}$

3. THE BAT ALGORITHM METAHEURISTIC

BA is also a population metaheuristic. It works on several solutions called bats at every time step $t$. Each of the $n$ bats is represented by a position in parameter space denoted by the $d$-dimensional vector $\vec{x}_{i}^{t} = (x_{i,1}^{t}, x_{i,2}^{t}, \ldots, x_{i,d}^{t})$ just like PSO. The characteristic equations of BA are the following:

$$f_{i} = f_{\text{min}} + \beta(f_{\text{max}} - f_{\text{min}})$$  \hspace{1cm} (1)

$$\vec{v}_{i}^{t+1} = \vec{v}_{i}^{t} + f(\vec{x}^{*} - \vec{x}_{i}^{t})$$  \hspace{1cm} (2)

$$\vec{x}_{i}^{t+1} = \vec{x}_{i}^{t} + \vec{v}_{i}^{t+1}$$  \hspace{1cm} (3)

The frequency of the sonar $f$ is bounded by $f_{\text{min}}$ and $f_{\text{max}}$ which are generally set to 0 and 1 respectively. The parameter $\beta$ is a random number drawn from a uniform distribution with support $[0,1]$. The position is updated at each iteration by moving it by an amount $\vec{v}^{t+1}$ called velocity. Thus far, there is no practical difference with PSO except that the personal best position is omitted. This is not insignificant, because the authors of PSO explicitly reported in Reference 4 that this component increased performance. It is therefore implicitly stated that the new mechanisms proposed in Algorithm 2 more than make up for this exclusion. Three other parameters are introduced in BA: pulse rate $r$, loudness $A$ and step size $\delta$. This last parameter controls the size of the random step taken around the global best position when the conditional statement $U(0,1) > r_{i}$ is true.

Pulse rate and loudness are used as parameters for the two probabilistic acceptance mechanisms of lines 8 and 11. This idea is borrowed from Simulated Annealing (SA) which introduced the notion that non-improving moves should be accepted following some probabilistic mechanism so that the algorithm can escape local minima and go on to search for other promising regions of the fitness landscape. While SA uses what is called a cooling schedule to gradually decrease the probability of accepting non-improving moves, BA uses pulse rate and loudness. Pulse rate (Equation 4) is a function of the initial value $r_{0}$ and $\gamma$ which controls speed of convergence as shown in Figure 1. A value of $r_{0} = 1$ was used for convenience since $r^{t+1} \rightarrow r_{0}$ as $t \rightarrow \infty$.

$$r^{t+1} = r_{0}[1 - e^{-\gamma t}]$$  \hspace{1cm} (4)
Algorithm 2. Pseudocode for minimization with BA

1. initialize positions, velocities, global best, pulse rate, loudness and $\gamma$
2. while $t < t_{\text{max}}$ do
3. \[ f = f_{\text{min}} + \beta(f_{\text{max}} - f_{\text{min}}) \]
4. \[ \vec{v}^{t+1} \leftarrow \vec{v}^t + f(\vec{x}^* - \vec{x}^t) \]
5. \[ \vec{x}_{\text{temp}} \leftarrow \vec{x}^t + \vec{v}^{t+1} \]
6. for $i = 1$: number_of_bats
7. \[ \text{if } U(0, 1) > r^t_i \text{ then} \]
8. \[ \vec{x}^{i+1}_t \leftarrow \vec{x}^* + \delta \cdot \text{Normal}(0, 1) \]
9. \[ \text{end if} \]
10. \[ \text{if } U(0, 1) < A^t_i \text{ and } f(\vec{x}_{\text{temp},i}) < f(\vec{x}^*) \text{ then} \]
11. \[ \vec{x}^* \leftarrow \vec{x}_{\text{temp},i} \]
12. \[ \vec{x}^{i+1} \leftarrow \vec{x}_{\text{temp}} \]
13. \[ r^{i+1}_t \leftarrow r_0[1 - e^{-\gamma t}] \]
14. \[ A^{i+1}_t \leftarrow aA^t_i \]
15. \[ \text{end if} \]
16. \[ \text{end for} \]
17. \[ \text{return } \vec{x}^* \]
18. \[ \text{return } \vec{x}^* \]

FIGURE 1 Effect of $\gamma$ on pulse rate

FIGURE 2 Effect of $\alpha$ on loudness
Loudness is calculated at each time step using the following equation:

\[ A^{t+1} = \alpha A^t \]  

(5)

Equation (5) introduces the parameter \( \alpha \) which controls the rate of exponential decrease as shown in Figure 2. Possible values of \( A \) are bounded by \([0, 1]\) and \( \alpha \) is bounded by \([0, 1)\). Since \( t \in \mathbb{Z}^+ \), the continuous curves in Figures 1 and 2 are shown for illustrative purposes.

4 | COMPARATIVE ANALYSIS

In Reference 3 PSO is said to be a particular case of BA when the effects of pulse rate and loudness are removed. This is incorrect since Original BA does not use PSO’s so-called autobiographical memory (i.e., \( c_2 \)) to track each particle’s personal best position and it also does not use inertia. To better understand the differences between BA and PSO, it is helpful to do a few elementary algebraic manipulations.

The first component is represented by lines 4 to 6 of Algorithm 2 as shown in Algorithm 3.

Algorithm 3. Component 1 of BA

4 \( f = f_{\min} + \beta(f_{\max} - f_{\min}) \)
5 \( \vec{v}^{t+1} \leftarrow \vec{v}^t + f(\vec{x}^t - \vec{x}^t) \)
6 \( \vec{x}_{\text{temp}} \leftarrow \vec{x}^t + \vec{v}^{t+1} \)

Equation (4) can be rewritten as:

\[ f_i = U(f_{\min}, f_{\max}) \]  

(6)

Combining Equations (1) and (2) gives:

\[ \vec{v}_{i}^{t+1} = \vec{v}_i^t + U(f_{\min}, f_{\max}) \cdot (\vec{x}^* - \vec{x}_i^t) \]  

(7)

Combining Equations (3) and (7) gives:

\[ \vec{x}_{i}^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} + U(0, c_1)(\vec{x}^* - \vec{x}_i^t) \]  

(8)

The same logic applied to lines 4 and 5 of PSO in Algorithm 1 gives:

\[ \vec{x}_{i}^{t+1} = \vec{x}_i^t + \omega \vec{v}_i^t + U(0, c_1)(\vec{x}^* - \vec{x}_i^t) + U(0, c_2)(\vec{p}_i - \vec{x}_i^t) \]  

(9)

From Equations (8) and (9), the first component of BA is shown to be a special case of its counterpart in PSO when \( \omega = 1, c_2 = 0 \) and \( c_1 \) is drawn from the following probability density function (PDF) derived from the ratio of two uniform distributions:

\[ f_{X}(z) = \begin{cases} 
\frac{1}{2} & 0 < z \leq 1 \\
\frac{1}{2}z^2 & z > 1 \end{cases} \quad \text{with } z = \frac{U(f_{\min}, f_{\max})}{U(0, 1)}. \]  

(10)

The resulting PDF is shown in Figure 3 for \( f_{\min} = 0 \) and \( f_{\max} = 1 \). The area in blue covers \( \int_0^1 1/2\,dz = 1/2 \) and the orange area covers \( \int_1^\infty 1/(2z^2)\,dz = 1/2 \) so that \( \int_0^\infty f_{X}(z)\,dx = 1 \). This shows that BA uses a random number generated from the same uniform distribution as PSO about half of the time and an exponentially decreasing function for the rest.
In proportional terms, BA takes larger random steps in the direction of the global best position than PSO about three quarters of the time.

The previous conclusion holds for \( f_{\text{min}} = 0 \) and \( f_{\text{max}} = 1 \), but it can be generalized. Given \( f_{\text{min}} = 0 \) and \( f_{\text{max}} \geq 1 \), BA will, on average, take larger steps than PSO in the direction of the global best position a fraction of the time equal to \( 1 - 1/(4f_{\text{max}}) \). This also means that BA overshoots the global best position by larger amounts than PSO. More precisely, the resulting long-tailed distribution causes the overshoot to sometimes be a large multiple of the distance between an agent’s current position and the global best position. For example, if the distance \( \| \vec{x}^* - \vec{p}_i \| \) is 1 unit, then the resulting push in the direction of the global best will be larger than 5 units 10% of the time, that is, \( \int_5^{\infty} \frac{1}{2z^2} \, dz = 1/10 \), causing an overshoot of 4 units. In practice, it is similar to increasing \( c_1 \) and \( \omega \) in PSO, so the difference is likely to be of no consequence due to the effects of parameter selection.

Component 2 covers lines 8 to 10 of Algorithm 2 as shown in in Algorithm 4.

**Algorithm 4.** Component 2 of BA.

8 if \( U(0, 1) > r_{l}^{t} \) then
9 \( \vec{x}_{i}^{t+1} \leftarrow \vec{x}^* + \delta \cdot \text{Normal}(0, 1) \)
10 end if

Equation (4) and Figure 1 show that the pulse rate is an exponentially increasing function whose rate is governed by the parameter \( \gamma \). Technically, since \( t \in \mathbb{Z}^+ \), pulse rate goes from 0 to \( r_0 \) geometrically fast instead of exponentially, but the distinction is abandoned hereafter. The main effect of the second component is that it separates BA into two phases. The first phase is when the algorithm starts. When \( t \) is small, the second component is almost surely always applied since \( r \) begins at zero and \( U(0, 1) > r_{l}^{t} \) is likely to be true for most agents. This means that the second component makes the swarm converge on the initial global best value \( x^*_{\text{init}} \) and makes every agent take random steps around it in the beginning. It was pointed out during the peer-review process that this behaviour was observed in Reference 23 with BA-controlled robots. This makes the values of almost every agent's personal best \( \vec{p}_i \) equal to the global best \( \vec{x}^* \) in the first few iterations. This is similar to stochastic hill climbing (see Appendix B) because the swarm is centred around a single agent and is effectively sampling its immediate neighbourhood for improving positions. As such, the next following position has to be in its immediate neighbourhood. The direct consequence of this finding is that BA is expected to perform relatively well on convex objective functions and less well on multimodal ones because “bad” initial positions will cause BA to converge to local extrema from the beginning.

The second phase of the BA optimization process is when \( t \) becomes larger. If \( r \) is considered to be equal to \( r_0 \) when it reaches \( \beta \) percent of its final value, then at \( t = \lceil -\ln(1 - \beta)/\gamma \rceil \), the probability that the second component is applied is simply \( 1 - r_0 \). For example, if \( \beta = 99\% \) and \( \gamma = 0.5 \), then it would be true starting at \( t = 10 \). Typical values of \( \gamma \) range from

![Figure 3](image-url)  
**FIGURE 3** PDF of \( c_1 \) when \( f_{\text{min}} = 0 \) and \( f_{\text{max}} = 1 \) to match BA
0.5 to 0.9, so the preceding result could be as low as $t = 6$. This is a small value considering that the number of epochs is commonly observed in the $10^3$ to $10^4$ range. Larger values of $r_0$ cause the second component to be applied less often.

The third component covers lines 11 to 16 of Algorithm 2 as shown in Algorithm 5.

**Algorithm 5.** Component 3 of BA

```plaintext
11 if $U(0, 1) < A_t^i$ and $f(\bar{x}_{temp,i}) < f(\bar{x}_t^*)$ then
12    $\bar{x}_t^* \leftarrow \bar{x}_{temp,i}$
13    $\bar{x}_t^{i+1} \leftarrow \bar{x}_{temp}$
14    $r_{t+1} \leftarrow r_0[1 - e^{-\gamma t}]$
15    $A_t^{i+1} \leftarrow \alpha A_t^i$
16 end if
```

Loudness is governed by the exponentially decreasing function given in Equation (5). The initial loudness for every agent is the same and is represented by $A^0$ without the $i$ subscript. The probability that the loudness condition is true is given by Equation (11).

$$P(U(0, 1) < A_t^i) = \begin{cases} 1, & 0 \leq t \leq t_s \\ \alpha^i A_t^i, & t > t_s \end{cases} \quad \text{with } t_s = \lceil -\ln(A^0)/\ln(\alpha) \rceil$$

The effect is that when an agent finds a better solution than the current global best, there is a 100% probability that it will be accepted until $t$ reaches a value of $t_s$ when it starts to decrease exponentially until it is practically zero. As pointed out in Reference 3 $\alpha$ plays a similar role as the cooling factor in Simulated Annealing (SA). The resemblance becomes clear when it is realized that $\alpha^i$ is equivalent to $b^e_i$ for values of $\alpha = e^{(\ln(b) - c)}/t$ which is a form of SA cooling schedule.

To summarize, it is clear from Equation (9) and the preceding analysis that BA does not meet the traditional criteria for being considered an original contribution. The so-called traditional criteria refer to the ones given by the Canadian Intellectual Property Office (CIPO), the European Patent Convention (EPC) and the United States Patent and Trademark Office (USPTO) for patentability. BA uses mechanisms that (a) have been discovered before, (b) can be simplified and (c) can be considered obvious by most individuals with an average understanding of stochastic optimization algorithms. It is believed that this is the fruit of inadequate methodological standards and that the use of metaphors obscures these findings which would have been obvious otherwise. To support these conclusions, the following items will be investigated experimentally:

1. It is hypothesized that BA is particularly sensitive to the initial positions of the swarm. It appears that performance would decrease significantly if the swarm was initialized in a region that did not contain the global minimum.
2. It appears likely that the pulse rate mechanism (line 8, Algorithm 4) is no better than roulette wheel selection since it is asymptotically equivalent.
3. It appears likely that that component 3 of BA (Algorithm 5) is no better than the probabilistic acceptance mechanism of simulated annealing.

## 5 METHODOLOGY

This section describes the methodological approach used in this research. The goal is to provide all of the essential information to make it reproducible and easy to understand. This is accomplished by clearly stating the research assumptions. In this subsection, the following elements will be discussed briefly:

1. The first hitting time performance metric.
2. The notion of $\epsilon$-neighbourhood.
First, the performance metric used in this research is the FHT which is defined in statistics as the first time a process reaches a specific subset of the state space. In this research, it is defined as the number of objective function evaluations (i.e., a measure of time) required to reach an $\varepsilon$-neighbourhood (i.e., a subset of the state space) of the known best solution of a benchmark function. Convergence is an interesting property, but it is often of no practical importance if the swarm converged to the global minimum or not provided the best-found solution is kept in memory throughout the optimization process.

Second, the notion of “hit” is clarified by defining what an $\varepsilon$-neighbourhood is. An algorithm is considered to have hit target when it finds a position whose fitness is within $\varepsilon = 10^{-2}$ or less of the global minimum for a given benchmark. It is assumed that there is no need to make a distinction between $10^{-2}$ and $10^{-3}$ or $10^{-30}$ because the basin of attraction of the global minimum has almost certainly been found with the former. Some judgement is required.

Finally, Section 5.1 contains details about the test set selection process and supports the use of a small number of test functions by emphasizing that selection should be based on problem characteristics rather than the size of the test set. The results obtained on the test set will be compared to the results on two more benchmarks to demonstrate the strength of this approach for future reference. Section 5.2 contains information about the parameter selection process by pointing to the appropriate literature and by describing the grid search procedure employed to find reasonable parameters for fair comparison. Section 5.3 explains how the research hypotheses stated in Section 4 will be explored and describes the nonparametric statistical tests used to compare FHT distributions. The data that support the findings of this study are openly available in https://github.com/iangagn/ENG-2019-11-084.

### 5.1 Test set

The standard practice for the investigation and comparison of metaheuristics is to compare their performances on a series of benchmark problems. This is partly due of the No Free Lunch Theorem (NFLT) \(^{26}\) which says that the ideal universal optimizer does not exist and that good performance on a set of problems comes at the detriment of others. The result is that researchers often use large sets of test functions in order to find what problems the studied algorithm is good for. This approach is incomplete and redundant for the following reasons:

1. It does not attempt to explain why the algorithms perform the way they do on specific problems or problem attributes, so it largely fails to provide generalizable knowledge.
2. Choosing benchmarks with no reason other than to reach a predetermined number almost guarantees that some functions will provide redundant information already provided by other functions of the set.

To avoid these problems, this research aims to characterize the performance of the studied algorithms when facing the common difficulties encountered in optimization problems and to define a performance profile for each. This is achieved by making use of the well-known De Jong test functions. \(^{27}\) This choice is motivated by the following items:

1. The small size of the test set (i.e., 5 functions) makes the results more tractable.
2. It was designed specifically to measure the strengths and weaknesses of optimization algorithms when confronted with specific difficulties such as ridges (Equation 13), discontinuities (Equation 14), noise (Equation 15) and multiple peaks (Equation 16).

This decision is also supported by the fact that the average empirical linear correlation coefficient ($\rho$) as defined in Reference 28 for the five test functions is only .258 (Figure 4). An instance of two variables that have a correlation coefficient of 0.25 is shown in Figure 5 for reference. Functions $f_1$ and $f_4$ are highly correlated ($\rho = 0.91$), but this is explained by the fact that they both have bowl-like shapes (note that Figure 9 is inverted) and that their global minima coordinates are the same (i.e., the origin). The empirical linear correlation coefficient matrix is an imperfect measure of test set diversity, but it confirms that the functions “look” different without having to manually inspect every pair of graphs. Figure 4 can be generated using ENG2019110845_Empirical_Correlation_Matrix.m from the referenced GitHub repository.
The first benchmark is the unimodal convex function called Sphere (Figure 6). It is the least challenging an optimizer can face and simply serves to measure its general efficiency. Trajectory methods (i.e., algorithms that only use one solution per time step) always perform well on this kind of function, because the gradient always points in the direction of the global extremum (i.e., there are no local extrema).

\[
f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2
\]

\[-5.12 \leq x_i \leq 5.12\]

\[\min(f_1) = f_1(0, \ldots, 0) = 0\]  \hspace{1cm} (12)

The second function is a generalization of the Rosenbrock function (Figure 7). The minimum is located inside a banana shaped valley where optimizers can get stuck due to the scarcity of improving positions (i.e., low gradient).

\[
f_2(\vec{x}) = \sum_{i=1}^{n-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]
\]

\[-5.12 \leq x_i \leq 5.12\]

\[\min(f_2) = f_2(1, \ldots, 1) = 0\]  \hspace{1cm} (13)
The third function is called Step (Figure 8) and the main difficulty it poses is that it is made up of a large number of plateaus (ie, where $\nabla f = 0$). Optimizers fail when their neighbourhood topologies are fixed and/or are too small to search outside the plateaus where they are trapped.

$$f_3(\mathbf{x}) = 6n + \sum_{i=1}^{n} \lfloor x_i \rfloor$$

$$-5.12 \leq x_i \leq 5.12$$

$$\min(f_3) = f_3([-5.12, -5], \ldots, [-5.12, -5]) = 0$$

**FIGURE 8** Step function

The fourth function is a quartic function with Gaussian noise (Figure 9) represented by $N(0, 1)$. Noisy functions have the property of being rugged. Optimizers can fail when local minima appear to exist where they do not. Note that Figure 9 shows an inverted (ie, multiplied by $-1$) version of the noisy quartic function for visualization purposes.

$$f_4(\mathbf{x}) = \sum_{i=1}^{n} (ix_i^4 + N(0, 1))$$

$$-1.28 \leq x_i \leq 1.28$$

$$\min(f_4) = f_4(0, \ldots, 0) = 0$$

**FIGURE 9** Noisy quartic function

The fifth function is called Shekel’s Foxholes (Figure 10). It is a plateau filled with steep basins where optimizers often cannot escape. Note that Figure 10 is inverted for visualization purposes.

$$f_5(\mathbf{x}) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$$

$$a = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & \ldots & 32 & 32 & 32 \\ -65.536 \leq x_i \leq 65.536 \end{bmatrix}$$

$$\min(f_5) = f_5(-32, \ldots, -32) = 0.9980$$

**FIGURE 10** Shekel’s Foxholes

It is important to note that the domains of Equations (12) to (16) do not have any special meaning and that they could just as well all be $[-100, 100]$. The values were taken directly from Reference 27. These functions are found in
the GitHub repository under the names `sphere.m`, `rosenbrock.m`, `linear_step.m`, `noisy_quartic.m` and `foxholes.m`.

5.2 Parameter selection

In the current metaheuristics literature, the parameter selection process is often left unmentioned which can leave the impression that the values were either chosen at random or carefully selected to produce a desired outcome. The purpose of this section is to explain how the parameters were chosen for BA and PSO. Since this research is mainly concerned with exploring the claim that BA is generally superior to PSO, it is believed that it is not necessary to compare both algorithms on the elusive best possible parameter settings. This is because if general superiority really existed, it should not require extensive parameter tuning to observe. In other words, the dominance pattern should be apparent. Despite this, efforts were made to find good parameter settings to make the comparisons as fair as possible. The parameters for BA were chosen by measuring median FHT for 100 runs on a multidimensional grid of parameter combinations and selecting the combination whose average ranks is the best on the test set. The ranges for the BA parameters are based on a meta-analysis\(^\text{24}\) of historical parameter settings. The studied parameter settings are given in Table 1.

| Parameter | Symbol | Values          |
|-----------|--------|-----------------|
| Frequency | \( f \) | \([0, 1], [0, 2]\) |
| Pulse rate| \( r_0 \) | \(0.1, 0.4, 0.7, 1\) |
| Loudness  | \( A \) | \(0.5, 1, 1.4, 1.9\) |
| Alpha     | \( \alpha \) | \(0.5, 1, 1.4, 1.9\) |
| Gamma     | \( \gamma \) | \(0.1, 0.5, 0.9\) |

This results in \(2 \times 4^3 \times 3 = 384\) combinations and \(384 \times 100 = 38,400\) runs in total. Each run is limited to 2500 epochs for a total maximum number of epochs of \(96 \times 10^6\) epochs. Since the number of epochs is limited, sometimes BA never hits the target. This is accounted for by adjusting the median FHT values of each configuration by dividing by the hit rate. For example, if the median FHT is 500 function evaluations and the hit rate is 90\%, then the actual FHT used for comparisons is \(500/0.9 = 555.5\) function evaluations. This is to prevent a bad configuration from winning by having a small number of good hits with low hit rate. The resulting best parameter set for BA is \( f \in [0, 2], r_0 = 0.7, A = 1, \alpha = 1.9, \gamma = 0.1\). These results can be reproduced or improved upon using the following MATLAB file in the referenced GitHub repository: ENG2019110845_Parameter_Selection_BA.m. A similar exercise with PSO yielded \( \omega = c_1 = c_2 = 0.7\) and corresponds to ENG2019110845_Parameter_Selection_PSO.m. The results of the experiments are reported in Appendix A. The final parameter settings are given in Table 2.

| Parameter | Value | Parameter | Value |
|-----------|-------|-----------|-------|
| \( f \)   | \([0, 2]\) | \( \omega \) | 0.7   |
| \( r_0 \) | 0.7   | \( c_1 \)  | 0.7   |
| \( A \)   | 1     | \( c_2 \)  | 0.7   |
| \( \alpha \) | 1.9   | —         | —     |
| \( \gamma \) | 0.1   | —         | —     |

It is interesting to note that the best \( \alpha \) value is 1.9 since this causes \( U(0, 1) < A_i^t \) to always be true (ie, because \( A^0 = 1 \) implies that \( A_i^t \geq 1 \) for all values of \( t \) and \( i \)). This suggests that line 11 of Algorithm can be simplified. This finding will be discussed in the results section in light of the hypothesis stated in Section 4 that the loudness mechanism is inferior or equal to the SA probabilistic acceptance mechanism.
5.3 | Experiments and nonparametric statistical tests

This section describes the experimental procedures used to investigate the hypotheses formulated in Section 4. Section 5.3.1 describes how the research hypotheses will be explored in Section 5.3.2 describes the two-sample Kolmogorov–Smirnov test used to determine if two samples come from the same parent distribution and finally Section 5.3.3 describes Wilcoxon–Mann–Whitney test to compare medians.

5.3.1 | Investigating the research hypotheses

This section describes the experimental procedures used to investigate the research hypotheses formulated in Section 4. The hypotheses are repeated here for convenience:

1. BA is suspected of being highly sensitive to initial positions.
2. The pulse rate mechanism is suspected of being no better than roulette wheel selection.
3. The loudness mechanism is suspected of being inferior or equal to the probabilistic acceptance mechanism of simulated annealing.

To measure the effect of different initial positions on BA’s performance (hypothesis 1), the two following schemes will be used:

1. Random uniform over the entire domain.
2. Random uniform in a restricted domain that does not contain the global minimum.

The domains are restricted to exclude a unit square centered on the global minimum for \( f_1, f_2 \) and \( f_3 \). Since the domain of \( f_4 \) is only \( 2.56 \times 2.56 \), the excluded square is \( 0.25 \times 0.25 \) and \( 5 \times 5 \) for \( f_5 \) because of its larger domain.

To compare the pulse rate mechanism (Algorithm) with roulette wheel selection (hypothesis 2), a variant of BA (\texttt{ba_roulette_wheel.m}) will be used to generate FHT distributions. A uniform probability roulette wheel with \( P(\text{true}) = .5 \) and \( P(\text{false}) = .5 \) will be used as shown in Algorithm. The line numbers on the left-hand side of Algorithm correspond to the line numbers of Algorithm.

\begin{verbatim}
Algorithm 6. BA with roulette wheel selection mechanism
8     if rouletteWheel(1/2, 1/2) == true then
9         \( \vec{x}_{i+1} \leftarrow \vec{x}_i + \delta \cdot \text{Normal}(0, 1) \)
10     end if
\end{verbatim}

To measure the effects of the loudness mechanism (hypothesis 3), a variant of BA with no loudness mechanism (\texttt{ba_no_loudness.m}) as shown in Algorithm will be used to generate FHT distributions.

\begin{verbatim}
Algorithm 7. BA without the loudness mechanism
11     if \( f(\vec{x}_{\text{temp},i}) < f(\vec{x}_i) \) then
12         \( \vec{x}_i \leftarrow \vec{x}_{\text{temp},i} \)
13         \( \vec{x}_{i+1} \leftarrow \vec{x}_{\text{temp}} \)
14     end if
\end{verbatim}

The resulting FHT distributions will be analyzed and compared using the nonparametric tests described in the next two sections. The reason why nonparametric tests are used is because the normality assumption does not hold for FHT distributions since FHT is naturally greater than or equal to 1.
5.3.2 THE TWO-SAMPLE KOLMOGOROV-SMIRNOV TEST

The FHT distributions generated from the experiments described in Section 5.3.1 will be compared using the Kolmogorov–Smirnov 2-sample test (KS2 hereafter). KS2 is a nonparametric statistical hypothesis test used to determine if two samples have the same parent distribution (see Reference 29 for more details). The procedure poses the following null ($H_0$) and alternative ($H_a$) hypotheses:

- $H_0$: The two samples come from the same distribution
- $H_a$: The two samples do not come from the same distribution

The corresponding test statistic is based on the maximum absolute difference between the empirical cumulative distribution functions (ECDF$_i$) of both samples represented by the following equation:

$$D = \sup_x |\text{ECDF}_1(x) - \text{ECDF}_2(x)|$$ (17)

The null hypothesis is rejected for a given confidence level $\alpha$ when the following condition is true where $n$ and $m$ represent the sample size of ECDF$_1$ and ECDF$_2$ respectively:

$$D > \frac{1}{\sqrt{n}} \cdot \sqrt{-\ln \left( \frac{\alpha}{2} \right) \cdot \frac{1 + \frac{n}{m}}{2}}$$ (18)

The corresponding $p$-values are given in Reference 30. This research uses the MATLAB implementation kstest2 (see https://www.mathworks.com/help/stats/kstest2.html).

5.3.3 The Wilcoxon–Mann–Whitney test

To compare performance between BA and PSO, FHT distributions are generated for both algorithms and the one with the smallest median FHT is the winner for a given benchmark. The differences in medians confirmed to a 0.05 confidence level using the Wilcoxon-Mann-Whitney test with MATLAB’s built-in ranksum function in the Statistics and Machine Learning Toolbox (https://www.mathworks.com/help/stats/ranksum.html). The technical details of the Wilcoxon–Mann–Whitney nonparametric test (WMW hereafter) are given in Reference 29.

In short, WMW takes two FHT populations, say $X_1$ and $X_2$ of size $n_1$ and $n_2$ respectively and tests the following hypotheses:

- $H_0 : P(X_1 > X_2) = P(X_2 > X_1)$
- $H_a : P(X_1 > X_2) \neq P(X_2 > X_1)$

If $H_0$ is true and $S_{X_1}$ is the sum of the ranks of the $n_1$ elements in $X_1 \cup X_2$, then it can be shown that the even $S_{X_1} = S_{X_2}$ approximately follows $N \left( \mu = \frac{n_1 n_2}{2}, \sigma^2 = \frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \right)$ provided that $n_1 > 20$ and $n_2 > 20$. From this, the inverse normal distribution can be used to reject the null hypothesis if the test statistic $\varepsilon = |S_{X_1} - \mu|/\sigma$ is greater than 1.96 for a 0.05 confidence level or 2.58 for a 0.01 confidence level.

6 RESULTS

This section contains the results obtained by following the protocol given in Section 5.3.1. Section 6.1 contains the results of the effect of two different initialization schemes, Section 6.2 contains the results of the effect of replacing the pulse rate mechanism with roulette wheel selection, Section 6.3 contains the results of the effect of removing the loudness mechanism and 6.4 contains additional observations.

6.1 Sensitivity to initial conditions

The FHT distributions were evaluated for BA and PSO with random uniform initialization on the full domain (Adjusted Median FHT A) and random uniform initialization on a restricted domain (Adjusted Median FHT B) as described in
Section 5.3.1. The adjusted medians are calculated as in Section 5.2 and statistical significance is assessed using KS2 (Section 5.3.2). The results are presented in Tables 3 and 4.

**Table 3** Impact of initialization scheme on median FHT for BA

| Function | Adjusted median FHT A | Adjusted median FHT B | Difference (%) | Statistically significant? |
|----------|-----------------------|-----------------------|-----------------|---------------------------|
| \( f_1 \) | 608                   | 816                   | + 34.2          | Yes                       |
| \( f_2 \) | 3604                  | 9843                  | + 173.1         | Yes                       |
| \( f_3 \) | 452 870              | 487 360              | + 7.6           | No                        |
| \( f_4 \) | 2670                  | 3340                  | + 25.1          | Yes                       |
| \( f_5 \) | 76 000                | 55 789               | − 26.5          | No                        |

**Table 4** Impact of initialization scheme on median FHT for PSO

| Function | Adjusted median FHT A | Adjusted median FHT B | Difference (%) | Statistically significant? |
|----------|-----------------------|-----------------------|-----------------|---------------------------|
| \( f_1 \) | 280                   | 360                   | + 28.6          | Yes                       |
| \( f_2 \) | 4220                  | 4960                  | + 17.7          | No                        |
| \( f_3 \) | 11 099                | 11 808                | + 6.4           | No                        |
| \( f_4 \) | 1740                  | 2240                  | + 28.7          | Yes                       |
| \( f_5 \) | 8191                  | 9710                  | + 18.5          | Yes                       |

The results for BA on \( f_3 \) and \( f_5 \) are consistent with those obtained during the parameter selection process in Appendix A. BA severely underperforms PSO on these functions and initialization does not appear to play a statistically significant role. It is believed that BA mostly finds the global minima for \( f_3 \) and \( f_5 \) by chance. The mean difference in adjusted median FHT values for \( f_1 \), \( f_2 \) and \( f_4 \) is 77.5% for BA and 25% for PSO. Both algorithms are similarly affected by the restricted initialization scheme on \( f_1 \) and \( f_4 \), so it could be argued that BA’s adjusted median FHT difference of 173.1% on \( f_2 \) is an outlier, but it is believed that this is not the case. It is proposed instead that the structural resemblance between \( f_1 \) and \( f_4 \) explain their similar sensitivities to initialization (ie, both are bowl-shaped convex or almost convex functions). This hypothesis is supported by the fact that the same differences for PSO in Table 4 are almost exactly identical (ie, 28.6% and 28.7%). The conclusion is that BA is in fact very sensitive to initialization and that the observed differences are statistically significant to a .05 confidence level using the KS2 test with a sample of 1000 observations.

### 6.2 | BA with roulette wheel selection instead of pulse rate

The FHT distributions (\( n = 1000 \)) were evaluated for BA with the pulse rate mechanism (Adjusted Median FHT A) and with roulette wheel selection (Adjusted Median FHT B) as shown in Algorithm of Section 5.3.1. The adjusted medians are calculated as in Section 5.2 and statistical significance is assessed using KS2 (Section 5.3.1). The results are presented in Table 5.

**Table 5** Effect of replacing pulse rate with roulette wheel selection

| Function | Adjusted median FHT A | Adjusted median FHT B | Difference (%) | Statistically significant? |
|----------|-----------------------|-----------------------|-----------------|---------------------------|
| \( f_1 \) | 611                   | 254                   | − 58.4          | Yes                       |
| \( f_2 \) | 4070                  | 3683                  | − 9.5           | Yes                       |
| \( f_3 \) | 478 550              | 3470                  | − 99.2          | Yes                       |
| \( f_4 \) | 2550                  | 2000                  | − 21.6          | Yes                       |
| \( f_5 \) | 184 510               | 14 894                | − 91.9          | Yes                       |
Using a uniform probability roulette wheel instead of the pulse rate mechanism has increased performance by 56.1% on average. The adjusted median FHT has decreased in all 5 test functions and the differences are statistically significant based on the KS2 test with a sample of 1000 observations. The differences are particularly important for the functions where BA did poorly, namely $f_3$ and $f_5$. With this simple change, BA suddenly becomes competitive with PSO (see Tables 4 & A2 for comparison). It is conjectured that this is due to the lessened impact of component 2 during the early phases of the optimization process. If this is true, then the initialization scheme is expected to have less of a negative impact. This was confirmed during a later experiment. Table 6 shows the adjusted median FHT with random uniform initialization (Adjusted Median FHT A) and initialization on a restricted domain (Adjusted Median FHT B) as described in Section 5.3.1. It can be seen that the differences are smaller with an average increase in FHT of 8.7% against 42.7% (Table 3).

### TABLE 6 Effect of initialization on BA with roulette wheel selection

| Function | Adjusted median FHT A | Adjusted median FHT B | Difference (%) | Statistically significant? |
|----------|-----------------------|-----------------------|----------------|---------------------------|
| $f_1$    | 254                   | 281                   | +10.8          | Yes                       |
| $f_2$    | 3855                  | 4642                  | +20.4          | No                        |
| $f_3$    | 3581                  | 3551                  | −0.8           | No                        |
| $f_4$    | 1980                  | 1950                  | −1.5           | No                        |
| $f_5$    | 12,672                | 14,548                | +14.8          | No                        |

#### 6.3 BA without the loudness mechanism

The FHT distributions ($n = 1000$) were evaluated for BA with the loudness mechanism (Adjusted Median FHT A) and without it (Adjusted Median FHT B) as shown in Algorithm of Section 5.3.1. The adjusted medians are calculated as in Section 5.2 and statistical significance is assessed using KS2 (Section 5.3.2). The results are presented in Table 7.

### TABLE 7 Effect of removing the loudness mechanism

| Function | Adjusted median FHT A | Adjusted median FHT B | Difference (%) | Statistically significant? |
|----------|-----------------------|-----------------------|----------------|---------------------------|
| $f_1$    | 622                   | 426                   | −31.6          | Yes                       |
| $f_2$    | 4185                  | 1058                  | −74.7          | Yes                       |
| $f_3$    | 377,760               | 493,110               | +30.5          | No                        |
| $f_4$    | 2680                  | 1820                  | −32.1          | Yes                       |
| $f_5$    | 147,440               | 236,630               | +60.5          | Yes                       |

Adjusted median FHT decreased by an average of 46.1% for $f_1$, $f_2$ and $f_4$ and increased by an average of 45.5% for $f_3$ and $f_5$. The difference is not statistically significant for $f_3$ despite the difference being large because the sizes of the samples that did find the target is small ($n_A = 134$, $n_B = 119$). It is believed that this difference is caused by fewer improving positions being rejected therefore making BA greedier (i.e., less explorative). This is supported by the fact that performance decreased for those functions where more exploration is beneficial, namely $f_3$ and $f_5$ and improved by a substantial amount otherwise.

#### 6.4 Additional observations

Figures 11 and 12 show that the original BA does behave like a stochastic hill climber as posited in Section 4. The global best positions are indicated by yellow stars (⭑) and the movements of the swarm centroids are shown by the red lines with the red cross markers (+). The successive generations of swarms are plotted as black circles in decreasing order of
transparency. The first generation is the most transparent and the most recent generations are the opaques. These figures show more or less exactly how one would expect a hill climber to navigate the landscape.

The stochastic hill climbing behaviour of BA is clearly beneficial for convex functions and ridges with non-zero gradients, but it does appear to pose problems for functions with discontinuities and/or noise as one would expect. Figure 13 shows BA stagnating on the plateaus of $f_3$ and Figure 14 shows BA centroids stagnating for over 1000 epochs on $f_4$ in the two regions circled in green.

The Foxholes function ($f_5$) is absent simply because BA consistently fails early on it (i.e., the swarm “falls” into one of the local minima and stays there the entire time) so the centroid movements are uninteresting. This observation suggests that BA could possibly be used to navigate objective functions that possess deep ridges like the HappyCat function described in Reference 31 as shown in Figure 15 along with the level curves that inspired its name. This is left as a future research project.

7 CONCLUSION

The results presented in the previous sections disprove the claim made in Reference 3, that BA is generally superior to PSO. In fact, the results support the opposite. This research also exposed two weaknesses of BA: plateaus and noisy objective functions. This might be attributable to parameter selection, but the results of the parameter selection process described in
Section 5.2 does not support this hypothesis. Instead, it is assumed that BA is simply not well-suited to these objective function characteristics. This being said, the tight swarm configuration of BA observed in Section 6.4 does appear to be good at finding promising directions in regions where they are rare (e.g., ridges) and the gradients are low (e.g., the banana-shaped valley of $f_2$) and exploiting them. This finding suggests that BA might be beneficial to the study of neural network landscapes since it was reported in Reference 32, that low-energy regions are connected to one another by relatively simple curves. In other words, the optimal regions share similarities with the banana-shaped valley of $f_2$. The results also support the research hypotheses made in Section 4. The first hypothesis is that BA is highly sensitive to initial conditions. This was confirmed to be true by comparing the effects of two initialization schemes between BA and PSO. The second hypothesis posited that the pulse rate mechanism is no better than roulette wheel selection. Not only was this confirmed in Section 6.2, but the experiments also showed that BA’s sensitiveness to initial conditions almost entirely disappeared when uniform probability roulette wheel selection was used instead of the pulse rate mechanism. The third and last hypothesis was that the loudness mechanism was inferior or equal to SA’s probabilistic acceptance mechanism. Section 6.3 proved that, in our study, the loudness mechanism could be abandoned entirely for those functions where BA already performed well relative to PSO. All of this suggests that the best-performing configuration of BA is actually a hybrid between PSO and SA. Further studies could confirm this by opposing this hybrid configuration against BA both with optimal parameters determined with design of experiments and response surface methodology. To conclude, it is important to mention that the findings presented in this article should, in the opinion of the authors, have been made well before presenting BA as an original contribution that outperforms PSO when it is easy to show that both of these statements are incorrect.

**Peer Review Information**

*Engineering Reports* thanks Ke-Lin Du and other anonymous reviewers for their contribution to the peer review of this work.

**Conflict of Interest**

The authors declare no potential conflict of interest.

**Author’s Contribution**

Iannick Gagnon: Conceptualization-Equal, Data curation-Equal, Formal analysis-Equal, Investigation-Equal, Methodology-Equal, Resources-Equal, Software-Equal, Validation-Equal, Visualization-Equal, Writing-original draft-Equal, Writing-review & editing-Equal. Alain April: Investigation-Equal, Methodology-Equal, Project administration-Equal, Supervision-Equal, Validation-Equal, Writing-original draft-Equal, Writing-review & editing-Equal. Alain Abran: Methodology-Equal, Supervision-Equal, Writing-original draft-Equal, Writing-review & editing-Equal.

**ORCID**

Iannick Gagnon [https://orcid.org/0000-0002-6446-5187](https://orcid.org/0000-0002-6446-5187)
REFERENCES

1. Rere LR, Fanany MI, Arymurthy AM. Simulated annealing algorithm for deep learning. *Proce Comput Sci*. 2015;72(1):137-144.
2. Kalra M, Singh S. A review of metaheuristic scheduling techniques in cloud computing. *Egypt Inform J*. 2015;16(3):275-295.
3. Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICOSSO 2010)* (pp. 65–74). Springer, Berlin, Heidelberg.
4. Eberhart R., & Kennedy J. (1995, November). Particle swarm optimization. Proceedings of the IEEE international conference on neural networks (Vol. 4, pp. 1942-1948).
5. Colorni A, Dorigo M, & Maniezzo V. (1992, December). Distributed optimization by ant colonies. Proceedings of the first European conference on artificial life (Vol. 142, pp. 134-142).
6. Dorigo M, Thomas S. *Ant Colony Optimization*. Cambridge, MA: MIT Press; 2004.
7. Jakobsen L, Brinklov S, Surlykke A. Intensity and directionality of bat echolocation signals. *Front Physiol*. 2013;4:89.
8. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw*. 2016;95:51-67.
9. Sörensen K. Metaheuristics—the metaphor exposed. *Int Trans Oper Res*. 2015;22(1):3-18.
10. Barr RS, Golden BL, Kelly JP, Resende MG, Stewart WR. Designing and reporting on computational experiments with heuristic methods. *J Heuristics*. 1995;1(1):9-32.
11. Boussaid I, Lepagnot J, Siarry P. A survey on optimization metaheuristics. *Inform Sci*. 2013;237:82-117.
12. Hooker JN. Testing heuristics: we have it all wrong. *J Heuristics*. 1995;1(1):33-42.
13. McGeoch CC. Toward an experimental method for algorithm simulation. *Inform J Comput*. 1996;8(1):1-15.
14. Brownlee J. (2007). An note on research methodology and benchmarking optimization algorithms. Complex intelligent systems laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, Victoria, Australia, technical report ID, 70125.
15. Bishop JM. (1989, October). Stochastic searching networks. Proceedings of the 1st IEE International Conference on Artificial Neural Networks, (Conf. Publ. No. 313); IET. pp. 329–331.
16. Millonas MM. (1993). Swarms, phase transitions, and collective intelligence. arXiv preprint adap-org/9306002.
17. Reeves WT. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans Graph*. 1983;2(2):91-108.
18. Duman E, Uysal M, Alkaya AF. Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem. *Inform Sci*. 2012;217:65-77.
19. Bastos Filho CJ, de Lima Neto FB, Lins AJ, Nascimento AI, & Lima MP. (2008, October). A novel search algorithm based on fish school behavior. 2008 IEEE International Conference on Systems, Man and Cybernetics; IEEE. pp. 2646-2651.
20. Clerc M. *Guided Randomness in Optimization*. Vol 1. Hoboken, NJ: John Wiley & Sons; 2015.
21. Maurice Clerc (2012). Standard Particle Swarm Optimisation. Ffhal-00764996f
22. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science*. 1983;220(4598):671-680.
23. Suárez P, Iglesias A, Gálvez A. Make robots be bats: specializing robotic swarms to the bat algorithm. *Swarm Evol Comput*. 2019;44:113-129.
24. Xue F, Cai Y, Cao Y, Cui Z, Li F. Optimal parameter settings for bat algorithm. *Int J Bio-Inspir Comput*. 2015;7(2):125-128.
25. Nourani Y, Andreassen B. A comparison of simulated annealing cooling strategies. *J Phys A Math Gen*. 1998;31(41):8373-8385.
26. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evolut Comput*. 1997;1(1):67-82.
27. De Jong, KA. (1975). Analysis of the behavior of a class of genetic adaptive systems [Doctoral dissertation].
28. Maulana, A. (2018). Many objective optimization and complex network analysis [Doctoral dissertation].
29. Sheskin DJ. *Handbook of Parametric and Nonparametric Statistical Procedures*. Boca Raton, FL: CRC Press; 2003.
30. Smirnov N. Table for estimating the goodness of fit of empirical distributions. *Ann Math Stat*. 1948;19(2):279-281.
31. Beyer HG, & Finck S. (2012, September). HappyCat—A simple function class where well-known direct search algorithms do fail. International conference on parallel problem solving from nature; Springer, Berlin, Heidelberg, pp. 367-376.
32. Garipov T, Izmailov P, Podoprikhin D, Vetrov DP, Wilson AG. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press; 2018:8789-8798.

How to cite this article: Gagnon I, April A, Abran A. A critical analysis of the bat algorithm. *Engineering Reports*. 2020;2:e12212. https://doi.org/10.1002/eng2.12212
APPENDIX A

This section contains the results used in Section 5.2 by conducting an exhaustive grid search on the parameter spaces of BA and PSO in order to make their comparison as fair as possible. Tables A1 and A2 report the hit rates (i.e., the fraction of runs in which the $\epsilon$-neighbourhood defined in Section 5 was found), the median FHT and the adjusted FHT for 1000 observations. An example of adjusted median (AM) calculation for the first row of Table A1 is given here

$$AM_{\text{Sphere}, BA} = \frac{480}{0.89} \approx 539$$  \hspace{1cm} (A1)

| Function | Name        | % Hit | Median | Adjusted median |
|----------|-------------|-------|--------|-----------------|
| $f_1$    | Sphere      | 89    | 480    | 539             |
| $f_2$    | Rosenbrock  | 72    | 3310   | 4597            |
| $f_3$    | Step        | 7     | 26920  | 384570          |
| $f_4$    | Noisy quartic | 100  | 2150   | 2150            |
| $f_5$    | Foxholes    | 6     | 2640   | 44000           |

**TABLE A2**  \hspace{1cm} PSO grid search results

| Function | Name        | % Hit | Median | Adjusted median |
|----------|-------------|-------|--------|-----------------|
| $f_1$    | Sphere      | 100   | 280    | 280             |
| $f_2$    | Rosenbrock  | 99    | 3480   | 3515            |
| $f_3$    | Step        | 99    | 10120  | 10222           |
| $f_4$    | Noisy quartic | 100  | 1880   | 1880            |
| $f_5$    | Foxholes    | 94    | 13260  | 14106           |

It is clear from Table A1 that BA does poorly with discontinuities (i.e., $f_3$) and multimodal functions (i.e., $f_5$). This finding is in line with the analysis given in Section 4, namely that BA tends to act like a stochastic hill climber. The best parameter set found during this exercise is the following: $f \in [0, 2]$, $r_0 = 0.7$, $A = 1$, $a = 1.9$, $\gamma = 0.1$.

PSO achieved hit rates of nearly 100% for the entire test set and dominated BA based on adjusted median FHT values. BA only outperformed PSO based on the non-adjusted median FHT for the Rosenbrock function.

Even though BA outperformed PSO on $f_2$, the difference is not statistically significant at the 0.05 confidence level based on the two-sample Kolmogorov–Smirnov test (Section 5.3.1). In BA’s defense, even if PSO has lower median and adjusted median values on $f_4$, the difference is also not statistically significant at the same confidence level. The differences on $f_1$, $f_3$ and $f_5$ are statistically significant. To summarize, BA appears to be competitive with PSO only on $f_2$ and $f_4$.

APPENDIX B

The purpose of this section is to show that BA often behaves like a stochastic hill climber as mentioned in Section 4 and referred to throughout the text. This is evident from Figures 11 and 12, but a more illustrative example is given here. BA was used with the parameters reported in Table 2 except that the swarm size was reduced to 5 to make the plots easier to understand. Of course, parameterization can help alleviate this tendency, but it was shown to be the best-ranked parameter set for the studied test set in Appendix A.

Figures B1 and B2 show the initial swarm configuration on the left-hand side and the subsequent configurations on the right-hand side. The initial best position (i.e., $t = 1$) is circled in red and an arrow points to a red box containing the entire swarm at $t = 2$. The iteration number corresponds to the northwest annotation. From there, the swarm remains tightly packed and effectively acts like a stochastic hill climber until it reaches the neighbourhood of the global best position represented by a yellow star.
The inference made in Section 4 that BA behaves like a stochastic hill climber was shown graphically. In Figure B1, it can be seen that when $t = 100$, the swarm is very close to the global best position, but it only hit target ($\epsilon = 10^{-2}$) at $t = 1910$. It does appear to behave like a stochastic hill climber. Perhaps this could be improved by using an adaptive random step size ($\delta$) or some other adaptive intensification procedure when the swarm appears to stagnate (ie, the centroid does not move by much from iteration to iteration). This could be the subject of future research.