Extending the Bat Foraging Metaphor for Optimisation Algorithm Design

Wei Cui

Complex Adaptive Systems Laboratory and School of Business
University College Dublin, Dublin, Ireland
E-mail: wei.cui.c@gmail.com

Anthony Brabazon

Complex Adaptive Systems Laboratory and School of Business
University College Dublin, Dublin, Ireland
E-mail anthony.brabazon@ucd.ie

Alexandros Agapitos

Complex Adaptive Systems Laboratory and School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
E-mail: alexandros.agapitos@ucd.ie

Abstract: A particular feature of most species of bats is that they use echolocation, or 'active sensing', in which pulses of acoustic energy are emitted and the resulting echo is resolved into an 'image' of their surrounding environment. This is used to detect objects and to locate food resources such as flying insects. Previous work has taken inspiration from the process of echolocation to develop the 'bat algorithm' (Yang, 2010) and this has demonstrated good results on a wide range of optimisation problems. In this paper we build on this work in order to stimulate further interest in exploration of a bat foraging metaphor as an inspiration for the design of optimisation algorithms. This study provides a review of some recent relevant literature on bat foraging and uncovers several aspects of the foraging process which have not been given explicit consideration in bat algorithm design thus far. We also outline a general framework of foraging behaviour which distinguishes between the role of 'perception', 'memory', and the use of the 'social' information available to a foraging bat. We demonstrate how some of these features can be integrated into an exemplar optimisation algorithm and test the performance of this algorithm on a series of benchmark problems. The study also provides several ideas for future work.

Keywords: Bio-inspired algorithms; Bat Algorithm; Perception; Memory; Social Communication.

Reference to this paper should be made as follows: Cui, W., Brabazon, A. and Agapitos, A. (2015) 'Extending the Bat Foraging Metaphor for Optimisation Algorithm Design', *International Journal of Metaheuristics,*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Dr Cui is a post doctoral researcher at University College Dublin working with the Natural Computing Research & Applications Group. His specialist areas of interest include natural computing, agent-based modelling, and applications of these methods for financial modelling.

Prof. Brabazon's research interests concern the development of natural computing algorithms and their application to real-world problems. He is co-founder and co-director of the Natural Computing Research & Applications Group at UCD (see http://ncra.ucd.ie). Anthony has published in excess of 200 peer-reviewed studies and has authored / edited thirteen books.

Dr Agapitos is a post doctoral researcher at University College Dublin working with the Natural Computing Research & Applications Group. His specialist areas of interest include machine learning, evolutionary computation, pattern recognition, combinatorial optimisation, robotics, and game artificial intelligence.

1 Introduction

Bats are small mammals and along with birds share the accolade of being the only vertebrates that are capable of powered flight (Jones and Teeling, 2006). Over 1,200 species of bat have been identified, ranging in size from the small Bumble Bee bat (weighing 2g) to the Indian Flying Fox bat which weights some 1500g (Arita and Fenton, 2011). Most species feed on small insects, with other species feeding on a variety of items including fruit, nectar, and small vertebrates (including fish).

1.1 Bat Sociality and Foraging

A notable aspect of bat behaviour is that many species of bat live in colonies and display complex social behaviours (Dechmann et al., 2010). Some species of bat can live for over 30 years (Fenton, 2013), providing ample time for extensive individual and social learning to occur.

Sociality is observed in many species of animals and a rich literature has developed in order to explain this phenomenon. Advantages of living in groups include, access to socially-acquirable information about mating partners, the presence or absence of predators, and information concerning resources such as shelter or food. There are also additional benefits in terms of physical protection against predators and, in the case of bats, thermoregulation which is important for the maturation of young bats. Sociality can also have costs which potentially include increased competition for food, or roosts, attraction of predators, and increased disease transmission rates.

Given the importance of foraging activities for all animals, it is reasonable to suggest that many evolutionary adaptations have occurred because they produced an improvement in foraging efficiency. Indeed, it is claimed that the benefits arising from group foraging have been an important driver of the evolution of sociality in bats (Dechmann et al., 2009; Kerth and Reckardt, 2003).

1.1.1 Social Foraging and Information

A key aspect of social foraging is the transmission of information between conspecifics about food finds and food locations. Although a huge array of specific natural mechanisms for social communication of information about food finds exist, these can be broadly classified as arising from either:

i. signals deliberately produced by an animal, or

ii. cues emitted as a by-product of an animal's behaviour.

Signals can be defined 'as stimuli produced by a sender and monitored by a receiver to the average net benefit of both parties', whereas cues are 'assessable properties that are at least partly correlated with a condition of interest' (Bradbury and Vehrencamp, 2011). In the first case, communication is intentional, for example, the emission of a 'food call' to deliberately alert other animals to a food find.

In the second case, communication to third parties occurs unintentionally as a result of the behaviour of an animal. An example of this is local enhancement effects (local enhancement can be defined as 'apparent imitation resulting from directing an animal's attention to a particular object or to a particular part of the environment' Thorpe (1963)) as a result of one animal observing another animal feeding at a food find. Hence, social transmission of information concerning food or other resources can arise as an unintentional side-effect of an animal's behaviour, rather than being deliberately communicated information.

More generally, social learning, of which social foraging is a behavioural subset, can be defined as 'learning that is influenced by observation of or interaction with another individual, or its products' (Rendell et al., 2011). Social learning is frequently contrasted with *asocial learning* (e.g. individual learning arising from trial and error) and is a potentially cheap way of acquiring valuable information.

1.2 Foraging Algorithms

The foraging literature (which is part of the wider field of behavioural ecology) indicates that there are four primary components to the foraging behaviour of animals (Giraldeau and Caraco, 2000; Stephens and Krebs, 1986), they:

i. have an individual perception capability, the prey-detection radius of which depends on the acuity of their senses (vision, touch, smell etc.),

ii. they may have memory of previously successful / unsuccessful food foraging locations. This is likely to be particularly useful when food locations are persistent for a period of time, or where food locations 'regenerate' cyclically,

iii. information about food locations may be socially transmitted between animals, and

iv. there may be a stochastic element to animal movement when searching for new food resources.

The stochastic element of animal movement whilst foraging has been modelled in a variety of ways in the foraging literature including power law distributions such as a Lévy distribution, whereby most animal movements are short, with occasional longer journeys (Viswanathan et al., 2011).

The weight given to each item by an animal can vary over time, and they are typically combined in a *foraging strategy* which can be context-sensitive. The precise strategy followed by an animal may depend on level of resource availability, level of hunger, degree of 'risk' in the immediate environment and so on.

A significant literature has emerged in natural computing (Brabazon, O'Neill and McGarraghy, 2015) which draws inspiration from the foraging behaviours of insects and animals to design algorithms for optimisation purposes. As we have an imperfect understanding of the exact interplay of the above four components in real-world animal foraging scenarios, most foraging algorithms in the natural computing literature emphasise one or a subset of the four components above. For example, ant colony optimisation algorithms (Bonabeau et al., 1999; Dorigo, 1992; Dorigo and DiCaro, 1999; Dorigo et al., 1996; Dorigo and Stützle, 2004; Romero and Robledo, 2013) and honey bee algorithms (Altwaijry and Menai, 2013; Karaboga and Basturk, 2007; Karaboga and Akay, 2009; Pham et al., 2006) place strong emphasis on social learning via recruitment.

Other families of optimisation algorithms such as particle swarm optimisation (PSO) (Kennedy and Eberhart, 1995; Kennedy et al., 2001; Samarghandi and Jahantigh, 2011) embed similar concepts. In the case of PSO, private and social memory (p_{best} and g_{best}) play a key role.

1.3 Bat Algorithm

A recent addition to the family of foraging algorithms in the natural computing literature is the *bat algorithm*, developed by Yang (2010) which draws inspiration from elements of the foraging processes of bats, specifically echolocation, in order to design an optimisation algorithm. The algorithm has produced very competitive results on both benchmark optimisation problems and across a variety of applications, encompassing constrained optimisation (Gandomi et al., 2013; Yang and Gandomi, 2012), multi-objective optimisation (Bora et al., 2012; Niknam et al., 2013; Yang, 2011), binary-valued representations (Nakamura et al., 2012), and clustering (Rui et al., 2012). A detailed review of applications of the bat algorithm is provided in Yang (2013) and Fister et al. (2014). Hybridisations of the bat algorithm are presented in Iztok et al. (2014, 2013).

At a high level, the process of bat foraging has a number of elements, and bats:

i. can directly detect items within a range of space based on the echoes to their own calls,
ii. can eavesdrop on echolocation calls from other bats,
iii. may possess a memory as to the location of food sources, and
iv. can move stochastically.

In the bat algorithm (Yang, 2010), two items are emphasised, namely a concept of information as to the current best foraging location in a population of bats, and a local search step which exploits information in the existing population. These processes embed some parallel with real-world bat foraging as they encompass social information transmission. The algorithm also seeks to maintain diversity in the population by means of a periodic random solution generation process.

1.4 Focus of This Study

In this study we seek to build on previous work on the bat algorithm in order to further stimulate research interest in exploring bat foraging metaphors as a source of inspiration for the design of optimisation algorithms. Accordingly, the primary focus of this paper is to develop a general framework to guide future research work. To this end we provide a review of a sample of relevant recent literature on bat foraging from the field of behavioural ecology. This uncovers several aspects of the real-world foraging process of bats which have not been given detailed consideration in bat algorithm design thus far. Drawing on literature from behavioural ecology we also we highlight the role of specific information sources such as ‘perception’, ‘memory’, and ‘social information’ in general foraging processes. We demonstrate how these features can be integrated into a general framework. Although the paper is not specifically focussed on developing a ‘better’ optimisation algorithm, we provide an exemplar of how the framework can be operationalised into a specific instance of an algorithm. We test this algorithm on a variety of test functions and find its performance to be promising.

The remainder of this contribution is organised as follows. Section 2 provides background on bat vocalisations, concentrating on their potential information content in a foraging context. Section 3 describes the developed framework and operationalises the framework to create a specific instance of a bat algorithm. The experimental design and results from testing this are presented and discussed in Section 4 and finally, conclusions and opportunities for future work are presented in Section 5.

2 Background

Although it was documented as long ago as 1793 by the Italian scientist Lazzaro Spallanzani that bats could avoid obstacles whilst flying in the dark, it was only in relatively recent times that the underlying mechanism of *echolocation*, or *active biosonar*, was identified (Griffin, 1944, 1958). Virtually all bat species, with the exception of some fruit bats, use echolocation in which they emit pulses of sound, and use the reflection (echo) of these pulses from surroundings to help detect, localise and classify objects, including obstacles and prey.

In essence, their brains process acoustic information to form ‘images’ of their surroundings by comparing the outgoing pulse with the returning echo. An echolocating animal obtains a snapshot of environmental information from each sound pulse and therefore, echolocation is a discrete-time sensory system. The computational complexity of processing echolocation information is underscored by the fact that it may involve both a moving source (bat) and target, and there may be a complex echo pattern arising from the many differing objects within detection range (Madsen et al., 2013). Information from echolocation is also integrated with information captured from other senses which can be acute. For example, in contrast to popular belief, no species of bat is blind and many have good vision (Fenton, 2013).

In addition to generating echolocation calls, bats are also capable of generating ‘social calls’ which they can use for communication purposes. In this section, we provide some background on each type of call and highlight how each can be used to assist a bat in foraging.

2.1 Why Did Echolocation Evolve?

Unlike day-time foraging animals which can primarily rely on vision to discover foraging opportunities or visual cues to obtain information on the foraging success of other group members (leading to local enhancement, where birds congregate at feeding sites), noctural hunters such as bats need to rely on non-visual signals.

It is speculated that echolocation in bats arose as a result of an evolutionary adaptation to hunt at night rather than compete for food during the day (Fenton, 2013; Jones and Teeling, 2006). A number of small nocturnal animals including rats (Jones, 2005) and cave-dwelling birds also use simplified versions of echolocation, as do a number of insects such as termites (Bradbury and Vehrencamp, 2011).

Echolocation is also advantageous in other environments where vision is less effective and is used by some marine species including dolphins and whales. The natural characteristics of water (sound travels 4.5 times faster and with much less attenuation than in air, Madsen et al. (2013)) mean that sounds can travel a considerable distance and hence, many marine animals have developed good hearing even when they cannot echolocate.

2.2 Bat Echolocation Vocalisations

Echolocating bats use a variety of frequencies, depending on species type, ranging from 8 to more than 200kHz (Fenton, 2013). Most, but not all, echolocating bat calls are ultrasonic and therefore beyond the range of human hearing. Bats can produce three canonical types of call, broadband, narrowband and long constant frequency. The calls can be complex and multi-harmonic (Jones and Teeling, 2006).

The nature of the echolocation calls produced by bats varies with some being broadband signals (wide range of frequencies with bandwidths of up to 100 kHz, typically of short duration) and others being narrowband signals (consisting of a narrow range of frequencies, circa 5kHz, and of relatively long duration) (Arita and Fenton, 2011; Jones, 2005). Narrowband signals are good for ranging distant items (or prey) and broadband signals are well adapted for the fined-grained localisation of items. This leads to a phenomenon whereby as insectivorous bats home in on their ariel prey, they switch from narrowband to broadband signals, which are emitted at an increasingly rapid rate as the bat approaches the prey, resulting in what is known as the 'feeding buzz'.

The key functions of echolocation pulses are to generate echoes that permit a bat to negotiate its three dimensional environment and for many bats, to find food. Echolocation pulses can be very loud, reaching $>$120 dB at 10cm in front of the bat's head in some cases. Even at such decibel levels, high frequency signals quickly attenuate in air. For example, a 12kHz pulse is audible at about 280 metres assuming a hearing threshold of 20 dB, whereas a 50kHz pulse is audible at only 35 metres (Jones and Siemers, 2011). As would be expected, returning echoes are much fainter.

The development of an echolocation capability in bats has in turn led to an evolutionary arms race, whereby some insect species have developed an ability to hear these echolocation pulses and take evasive action in response to them. A study by Nolen and Roy (1986) demonstrated negative phonotaxis in a number of species of flying crickets in response to detection of low intensity ultrasound of frequencies used by echolocating bats (indicative of bats flying about 10-18 metres from the insect), with this becoming 'last chance' avoidance behaviour (a very sudden directional shift) in response to the detection of high-intensity ultrasound by the insect (indicative of bats approaching within 1-2 metres).

2.3 Broadcast Information From Echolocation Calls

As bat call echoes (reflections) are strongly attenuated in air, bats can hear calls emitted by other bats from much further away than they can detect echoes from their own calls. Bats approaching feeding individuals or groups, and eavesdropping on their calls, can therefore increase the active space of their prey detection range between 10 to 50 times, depending on species, over that provided by their own echolocation ability.

Specific examples of the benefits to eavesdropping on echolocation calls include (Jones, 2008) finding that 'little brown bats', *Myotis lucifugus*, had an insect detection range of some 5 metres, whereas they can hear the echolocation calls of foraging conspecifics at some 50 metres, and Dechmann et al. (2010) reporting the case of *M. molossus* which can detect a small insect at a range of some 2 metres but can detect calls of conspecifics at some 54 metres under the same acoustic conditions. Other documented detection ranges for echolocation calls of other bats run between 20 and 50 metres (Fenton, 2013).

2.4 Summary

It is clear that acoustic calls by bats serve multiple purposes assisting them in night-time flying and resource hunting. There is also good evidence that many bat species are capable of decoding information from both the echolocation and social calls of other bats in order to assist with foraging or for other purposes. In the case of echolocation, information transfer is usually unintentional on the part of the sender and takes place via eavesdropping. In the case of social calls there is a greater likelihood that information transfer is intended. In the next section we outline a general framework which can integrate various aspects of bat foraging behaviour and then develop an optimisation algorithm using this framework.

3 Model Development

In the framework developed in this study, the movement of each bat is governed by the four components of a foraging behaviour as outlined in Sect. 1.2. In each iteration of the algorithm, a bat is displaced from its previous position through the application of a velocity vector:

$$p_{i,t} = p_{i,t-1} + v_{i,t} \tag{1}$$

where $p_{i,t}$ is the position of the i^{th} bat at current iteration, $p_{i,t-1}$ is the position of the i^{th} bat at previous iteration, and

$v_{i,t}$ is its current period velocity. The velocity update is a composite of four elements namely, prior period velocity, a social influence mechanism, a memory mechanism, and an individual perception mechanism. The velocity update for each bat is therefore described stylistically as:

$$v_{i,t} = v_{i,t-1} + Social_{i,t} + Memory_{i,t} + Perception_{i,t} \quad (2)$$

or more generally if we allow varying weights w on each element of the update,

$$v_{i,t} = w_1 v_{i,t-1} + w_2 Social_{i,t} + w_3 Memory_{i,t} + w_4 Perception_{i,t} \quad (3)$$

In all the experiments of this study, Equation 2 is used for calculating the velocity updates (i.e. we assume that all coefficients are set to one). Hence, in essence, Equation 2 is the 'foraging strategy' for bat i, as it provides the decision rule of the bat concerning its changes of location. The next subsection explains the operationalisation of each of the components of the update.

3.1 Prior Period Velocity

The inclusion of a prior period velocity can be considered as a proxy for momentum or inertia. This term is motivated by empirical evidence from the movement ecology literature which indicates that organisms move with a 'directional persistence' (Viswanathan et al., 2011). In other words, organisms are more inclined to continue in their current direction of travel than to reorientate themselves randomly.

3.2 Social Influence

The social influence simulating eavesdropping of the echolocation calls of other bats, for the i^{th} bat is determined by the following vector:

$$Social_i = p_j - p_i \ , \qquad j \neq i \quad (4)$$

where p_j is the position of the bat j that is currently foraging at the best location within the hearing range of the the i^{th} bat, and p_i is the current position of the i^{th} bat. If no other bat is currently within the hearing range of bat i, this term is set to zero.

3.3 Memory Influence

The influence of memory on the movement of the i^{th} bat is given by the following vector:

$$Memory_i = p_i^{best} - p_i \ , \quad (5)$$

where p_i^{best} is the best foraging location found by the i^{th} bat during its lifetime to date, and p_i is the current position of the i^{th} bat.

3.4 Perception Influence

Individual perception is implemented as follows. At each update, each bat seeks to detect any prey in its local area, within an assumed perception region of radius $r_{perception}$. While a real-world bat will have a specific arc of hearing detection depending on the direction which it is facing, we simulate an individual perception process by employing random sampling in a hypersphere around the bat on grounds of generality. The individual perception influence for the i^{th} bat is determined by:

$$Perception_i = s_i - p_i \ , \quad (6)$$

where p_i is the current position of the i^{th} bat, s_i is the position of the sampled point ($0 <\mid s_i - p_i \mid \leq r_{perception}$), and $r_{perception}$ is the radius of the assumed range within which the i^{th} bat can sense environmental information. The bat is motivated to move towards this point if it offers higher potential than its current location, otherwise this factor is ignored. The perception region (hearing range) $r_{perception}$ is given by:

$$r_{perception} = \frac{R}{1.8 \sqrt[D]{N}} \quad (7)$$

where D is the dimensionality of the benchmark problem, R is the radius of the search space (i.e. the search space is a sphere if D=3 and the corresponding R is the radius of the sphere), and N is the population size. The constant coefficient is an adjustable parameter. Hence, $r_{perception}$ is scaled appropriately as the benchmark problem is altered.

Pseudocode for the algorithm is provided in Algorithm 1. The framework could be operationalised in a multitude of ways, and the outlined algorithm represents one member of a large family of possible algorithms which would be consistent with this framework. On grounds of conciseness, for the rest of this paper we refer to the specific algorithm developed as the *Extended Bat Algorithm* or 'EBA'.

3.5 Comparison of Extended Bat Algorithm with Other Optimisation Heuristics

At a surface level the EBA developed above bears similarities with some other optimisation heuristics (here we consider PSO and the GA) and it is useful to examine its workings in order to clarify what features it shares with existing algorithms and what features are distinct.

To provide a framework for this comparison we employ the four elements of a foraging behaviour described in Sect. 1.2 above. Table 1 provides a synopsis of how the four elements can be mapped to each of the algorithms considered. While PSO and the GA are not typically considered as foraging algorithms, they are included in this table for discussion purposes.

3.5.1 Perception

In PSO and the GA, the particles / genotypes do not have an explicit perception mechanism for sampling (perceiving)

Algorithm 1: Bat Optimisation Algorithm

```
Randomly initialise the location ($p_i$) and current velocity ($t_i$) of each of the $n$ bats ($i : 1 \rightarrow n$);
Calculate the fitness of the current foraging location of each bat and store each of these in $p_i^{best}$, being the best position found so far by each bat;
Store location of best overall fitness across $n$ bats in $best$;
repeat
    for all bats in turn do
        Make a perception in the hypersphere of radius $r_{perception}$ around the bat's current location;
        For all bats within hearing range of bat $i$ (if any), determine which one is at the best location;
        Update velocity of bat $i$ using eq. 2;
        Update position of bat $i$ using eq. 1;
        Evaluate the fitness of that location and update $p_i^{best}$ if necessary;
    end
    Update $best$ if necessary;
until terminating condition;
```

Table 1 Mapping of four foraging behaviours to the EBA, PSO, and GA algorithms

Foraging Behaviour	EBA	PSO	GA
Individual Perception of Environment	✓	✗	✗
Personal Memory	✓	✓	✗
Social Transmission of Information	✓	✓	✓
Stochastic Search Component	✓	✓	✓

the quality of locations in the local environment as they traverse it. They only assess the worth of a single location in each iteration, in essence obtaining feedback information. In contrast, a simple perception mechanism is embedded in the EBA whereby a bat assesses its immediate surroundings (within its hearing range) and this information influences its ‘move’ decision.

3.5.2 Memory

Memory is implemented in distinct ways in each algorithm. In the canonical PSO algorithm, memory is embedded via a concept of personal memory in which each particle ‘remembers’ the location of the best solution it has found to date (p_{best}). The EBA embeds a similar concept. In contrast, in the GA, individuals do not explicitly maintain a memory of the best location found by them to date.

3.5.3 Social Transmission of Information

A core component of the EBA, PSO and the GA is the social transmission of information. In the canonical version of PSO, all particles are aware of the best location found to date by any member of the swarm (g_{best}) and this information is blended with personal information and a momentum factor in order to produce the position update for each particle. Hence, it can be considered that g_{best} is ‘publicly broadcast’ to all members of the swarm and this information always impacts on the position update of particles. In the GA, the selection process (for reproduction) is typically biased in favour of the ‘better’ genotypes in the current population and hence, ‘good’ genes are preferentially carried into the next generation. In the EBA, the social transmission of information takes place via a simulated hearing mechanism, whereby bats can ascertain the relative foraging success of conspecifics from eavesdropping on their echolocation calls. This mechanism is distinct from that of the PSO, as the social mechanism is distance-based (based on hearing range), and therefore is not based on a pre-defined topology of linked particles such as g_{best} or l_{best}.

3.5.4 Stochastic Search

All three algorithms embed stochastic search using different mechanisms. In canonical PSO algorithm, the relative weight accorded to p_{best} and g_{best} is varied stochastically and in the GA, the selection, crossover and mutation processes are typically stochastic. In the EBA, the perception mechanism embeds a stochastic element and the random initialisation of the location and velocity of the bats also injects a stochastic element.

3.5.5 Comparison with Canonical Bat Algorithm

The canonical bat algorithm (Yang, 2010) has shown itself to be a powerful optimisation heuristic, producing good results across a variety of applications. An interesting element of the search process in the canonical bat algorithm is that the velocity update process contains a g_{best} concept similar to that of PSO, in that the movement of the bats in each iteration is partly determined by the location of the current best foraging location across the entire population of bats. Thus, g_{best} is somewhat akin to the location of the ‘loudest’ current feeding buzz in a swarm of bats. Critically, in the algorithm all bats in the swarm know the location of this point, and thus in effect, have infinite sensory perception for echolocation calls across the search space.

In contrast, in real-world foraging, a bat must be within hearing range of a call to benefit from eavesdropping, and therefore, bats can only respond to echolocation calls and feeding buzzes which are local to them. Apart from eavesdropping, bats have a personal sensory perception region based on echoes to their own calls. The detection (or not) of prey within this area will likely influence a real-world bat's response to hearing a feeding buzz from further away.

Comparing our proposed framework with the canonical bat algorithm developed in Yang (2010), we note two important differences:

i. The canonical bat algorithm uses the concept of g_{best} that represents the the current best location across the *entire* population, whereas EBA incorporates the concept of personal best p_{best} that represents the best foraging location found by an individual bat during its life-time.

ii. In EBA, each simulated bat uses an individual perception mechanism to stochastically sample its local neighbourhood. This feature is not present in the canonical bat algorithm.

4 Results and Discussion

In this section we describe the experiments undertaken and present the results from these experiments.

4.1 Benchmark Functions

The algorithm outlined above was tested using a dozen optimisation benchmark functions (Table 2), at three levels of dimensionality (20, 40 and 60), giving a total of 36 experiments. The first five functions are unimodal and the remaining seven functions are multimodal. In foraging terms, the benchmarks can be considered as representing resource environments with differing degrees of resource 'patchiness'.

An interesting aspect of the functions is that the global optima are shifted or rotated (shift is given by the parameter $\boldsymbol{o}$, and the parameter M represents an orthogonal matrix which is used to rotate the function). The net effect of applying these transformations is to move the global optimum away from the origin in each case, due to the known issues with using standard benchmark functions which have their optimum at the origin (Liang et al., 2005). Known problems in conventional benchmark functions which can sometimes be exploited by algorithms to produce an upward bias in reported performance, include:

- many popular benchmark functions are symmetric, and hence have the same optimal parameter values for all dimensions (for example, a vector of zeros); and
- the global optimum may lie at the centre of the search space (this can produce problems if search agents are initialised randomly along the range of each dimension).

Hence, considering the conventional sphere function as an example,

$$f(x) = \sum_{i=1}^{D} x_i^2$$

the shifted sphere function is given by:

$$f(x) = \sum_{i=1}^{D} (x_i - o_i)^2$$

and the shifted rotated sphere function is given by:

$$f(x) = \sum_{i=1}^{D} [(x_i - o_i) * M]^2$$

where o is the shifted global optimum and M is a linear transformation matrix.

The aim in all the experiments is to find the vector of values which minimise the value of the test functions.

4.2 Overview of Experiments

In the experiments we undertake an initial assessment of the performance of the EBA on the test problems. We benchmark these results against those of the genetic algorithm (GA), and the canonical version of the PSO algorithm (Kennedy and Eberhart, 1995; Kennedy et al., 2001).

We employ the GA as a benchmark on the grounds that it is a well-known, widely-used, optimisation algorithm. In the experiments we have kept crossover and mutation rates at 0.70 and 0.01 respectively. Our rationale for selecting the PSO algorithm is that, in addition to being a widely-used heuristic for optimisation, it bears some similarity to the EBA in that it explicitly incorporates a concept of social and private information and blends this information in the search process. In our implementation we use the canonical version of the PSO algorithm, with the following update steps:

$$\begin{aligned} v_i(t) = \omega v_i(t-1) + c_1 rand(0,1)(p_{best}(t-1) \\ -x_i(t-1)) + c_2 rand(0,1)(g_{best}(t-1) - x_i(t-1)) \end{aligned} \tag{8}$$

$$x_i(t) = x_i(t-1) + v_i(t) \tag{9}$$

where $x_i(t)$ and $v_i(t)$ are the position and velocity of particle i at time t, ω is the inertia weight (initially set to 1), $rand(0,1)$ is a random number generator which generates a number in the interval $[0,1]$ and c_1 and c_2 are set to 2.

Of course, many variants of the GA and PSO algorithm exist in the literature and could have been chosen for implementation in our study. The focus of our experiments is to obtain initial insight into the performance of the developed EBA and assess whether it appears reasonably competitive against other heuristics. No claim is made that the canonical versions of the GA or PSO used in this study produces the best possible performance from the entire family of these algorithms on the test problems.

4.3 Experimental Settings

The experimental parameters are shown in Table 3. In each experiment, 30 bats in the case of the proposed bat algorithm or 30 particles / genotypes in the case of PSO / GA, are used, and an equivalent number of fitness evaluations are allowed

Table 2 Optimisation Problems (Suganthan et al., 2005)

Name	Function	Search Space	Optima
Shifted Sphere (F1)	$F_1(\mathbf{x}) = \sum_{i=1}^{D} z_i^2 - 450\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o}$	$[-100\ 100]^D$	-450
Shifted Schwefel's Problem 1.2 (F2)	$F_2(\mathbf{x}) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_i)^2 - 450\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o}$	$[-100\ 100]^D$	-450
Shifted Rotated High Conditional Elliptic (F3)	$F_3(\mathbf{x}) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 - 450\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-100\ 100]^D$	-450
Shifted Schwefel's Problem 1.2 (F4) with Noise in Fitness	$F_4(x) = (\sum_{i=1}^{D} (\sum_{j=1}^{i} z_i)^2) * (1 + 0.4\|N(0,1)\|) - 450\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o}$	$[-100\ 100]^D$	-450
Schwefel's Problem 2.6 (F5)	$F_5(\mathbf{x}) = max\|A_i\mathbf{x} - B_i\| - 310\ ,\ A\ is\ a\ D * D\ matrix, B_i = A_i * \mathbf{o}$	$[-100\ 100]^D$	-310
Shifted Rosenbrock (F6)	$F_6(\mathbf{x}) = \sum_{i=1}^{D-1} 100(z_i^2 - z_{i+1})^2 + (x_i - 1)^2 + 390\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o} + 1$	$[-100\ 100]^D$	390
Shifted Rotated Griewank (F7)	$F_7(\mathbf{x}) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{z_i}{\sqrt{i}}) + 1 - 180\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[0\ 600]^D$	-180
Shifted Rotated Ackley (F8)	$F_8(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e - 140,$ $\mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-32\ 32]^D$	-140
Shifted Rastrigin (F9)	$F_9(\mathbf{x}) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) - 330\ ,\ \mathbf{z} = \mathbf{x} - \mathbf{o}$	$[-5\ 5]^D$	-330
Shifted Rotated Rastrigin (F10)	$F_{10}(\mathbf{x}) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) - 330\ ,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-5\ 5]^D$	-310
Shifted Rotated Weierstrass (F11)	$F_{11}(\mathbf{x}) = \sum_{i=1}^{D} (\sum_{k=0}^{20} [0.5^k \cos(2\pi 3^k (z_i + 0.5))]) - D\sum_{k=0}^{20} [0.5^k \cos(\pi 3^k)] + 90$ $\mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-0.5\ 0.5]^D$	90
Schwefel's Problem 2.13 (F12)	$F_{12}(\mathbf{x}) = \sum_{i=1}^{D} (A_i - B_i(x))^2 - 460\ ,\ A_i = \sum_{j=1}^{D} (a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j),$ $B_i(x) = \sum_{j=1}^{D} (a_{ij}\sin x_j + b_{ij}\cos x_j),\ for\ i = 1, ..., D$	$[-\pi\ \pi]^D$	-460

to each algorithm. All reported results are averaged over 30 independent experiments for each algorithm and each test function. For each problem, we test the statistical significance of all differences in the means using a t-test. The experiments were undertaken on an PC Intel Core i7 (2.93 GHz) system with 12 GB RAM.

4.4 Presentation of Results

Figures 1-4 and Tables 4-6 present the results from our experiments. The Figures provide boxplots of the end-of-run mean best fitnesses for the EBA, and each comparator algorithm, averaged over 30 runs, across all twelve test functions (the Figures for the 20D case are qualitatively similar and are therefore omitted to save space).

Tables 4-6 summarise the end-of-run results, over 30 runs for each algorithm, across all three levels of dimensionality, where 'Best' represents the best fitness value obtained from all 30 runs, and 'Mean' represents the average of the best fitnesses obtained on each run. 'Std' represents the standard deviation of average best fitnesses over all 30 runs.

These tables also present the results from a number of statistical tests in which the null hypothesis H_0 is that there is no difference in performance between EBA and the relevant algorithm (PSO - particle swarm optimsation, GA - genetic algorithm) with which it is being compared. Hence, low p values (below 0.025) indicate that the null hypothesis of no difference in the means is rejected as we apply a 95% level.

4.5 Discussion of Results

4.5.1 Effectiveness of EBA

Initially we overview Figures 1-4 to get an idea of the general trends in the results. From the end-of-run boxplots (Figures 1-4) we can observe that in terms of 'Best' and 'Mean' the EBA, across all test functions and all three levels of dimensionality, generally outperforms the GA, as well as performing very competitively against PSO.

Looking at the detail in Tables 4-6, it is noticeable that as the dimensionality of the search space increases from 20 to 60, the general ordering of the performance of the algorithms remains similar, although as would be expected, the absolute performance of each algorithm tends to decrease as the problems become more difficult.

Initially, we consider the results for 'Best' performance for each algorithm on the test functions. In the case of GA, the 'Best' result from EBA exceeds that produced by GA in twelve out of twelve cases on each of the three levels of dimensionality (20D, 40D and 60D). Finally, in the case of PSO, the 'Best' result from EBA exceeds that produced by PSO in ten out of twelve cases (20D), twelve out of twelve cases (40D), and ten out twelve cases (60D). Hence, in terms of 'Best' performance, the EBA performs very competitively against all three comparators outperforming them in virtually all instances.

Next, we consider 'Mean' performance (averaged over all 30 runs) for each algorithm on the test functions. In the comparison with the results from GA, mean performance of the EBA exceeds that of GA in all cases at all three levels of dimensionality. In all cases, the difference in performance is statistically significant. Hence, we can conclude that the EBA outperforms GA on the tested problems. Finally, examining the results from PSO we note that 'Mean' performance of the EBA exceeds that of PSO in ten of twelve cases (20D), ten of twelve cases (40D), and six of twelve cases (60D). Hence, the results of EBA are competitive against those of GA and PSO, although there is some indication that the performance of EBA relative to PSO declines as the dimensionality of the problem increases.

4.5.2 Convergence and Stability of EBA

The boxplots (Figures 1-4) indicate that, in general, across the twelve test functions at all three levels of dimensionality, the end-of-run results for the EBA algorithm show quite good convergence. This suggests that the results are robust to the (random) choice of initial position and velocity for each bat.

Comparing the standard deviation (SD) of the 'Mean' results from EBA against that of GA, the SD of the results from the EBA are lower than those of the GA across all twelve test functions at all three levels of dimensionality. Finally, considering PSO, the SD of the results from EBA is lower than that produced by PSO on ten of twelve cases (20D), four of twelve cases (40D), and six of twelve cases (60D) respectively. Hence, we can conclude that EBA results for 'Mean' show greater convergence than those produced GA, and similar levels of convergence to those of PSO.

Table 3 Parameter Settings for Experiments

Parameters	Values
Runs	30
Size of Population	$N = 30$
Dimension of Problem	$D = 20, 40, 60$
Hearing Range	$r_{perception} = \frac{R}{1.8\sqrt[D]{N}}$

5 Conclusions

A significant stream of literature which draws inspiration from the foraging activities of various organisms to design optimisation algorithms has emerged over the past decade. As with many natural computing paradigms, these algorithms are populational in nature. A key component of many of them is a social communication mechanism which transfers information about good locations on the search landscape between members of the population. As discussed in this paper, social communication is not always intentional and may occur as a by-product of an animal's behaviours.

In this paper we contribute to the literature on bat algorithm design by reviewing recent literature from behavioural ecology on bat foraging. We uncover several aspects of the real-world foraging process of bats which have not been given explicit consideration in bat algorithm design thus far. We highlight the role of specific information sources such as 'perception', 'memory', and 'social information', demonstrate how these features can be integrated into a general framework, and provide an exemplar of how this can be operationalised into a specific instance of an optimisation algorithm.

The general framework developed in this paper for conceptualising the foraging process of bats opens up several ideas for future work. An interesting follow-on study would be to investigate the importance of each element of the framework in order to determine which (or which combinations) are most important in producing effective search behaviour. Of course, as with any empirical study, there is also an opportunity to investigate the performance of the algorithm(s) on additional benchmark functions and across a variety of parameter settings. In addition to the utility of such work for optimisation algorithm design, such simulation studies could potentially help us to better understand the real-world bat foraging process.

Other aspects of bat foraging behaviour could also be usefully investigated for their inclusion into an optimisation algorithm. As discussed in Section 2.3, the information content from hearing a feeding buzz is likely to be substantial as the greater the density of prey, and feeding bats, the louder the feeding buzz will be from that area. Plausibly, a bat will be more influenced by a heavy concentration (density) of feeding buzzes coming from a small area than a solitary feeding buzz coming from elsewhere.

In this study we have assumed equal weighting on each update component. More realistically, a bat will use a tacit decision rule or 'strategy' to process the multiple pieces of information, and may apply time-varying weights to each. For example, a bat will plausibly prioritise personal detection of a nearby food source rather than a further away feeding buzz as the energy cost of harvesting each resource will differ.

Apart from the processes of echolocation, we have commented on the role of social calls amongst bats concerning food resources. These calls can occur at a feeding site, providing 'real-time' information on food resources, or at a roost site (Wilkinson, 1992). Such information transfer at roosts bears parallel to the colony-based information transfer of other central place foragers such as honey bees, and the mechanisms of this process could inspire the design of an optimisation algorithm. Roost-based, or off-line, learning of central place foragers is more likely to be useful in communicating information about static resource locations such as fruiting trees for fruigivore bats. In contrast, information on ephemeral food resources such as a swarm of insects cannot be shared over repeated foraging sessions, so information can only be usefully exchanged in real-time during a foraging flight (Dechmann et al., 2010). Hence, mechanisms drawing on roost-based learning may be more useful for static rather than dynamic optimisation problems. This point reminds us that we need to consider the foraging context from which we draw inspiration when designing optimisation algorithms in order to apply the developed algorithms to appropriate problem settings.

Beyond this paper, Roger's paradox discussed in Section 1.1.1 provides some useful lessons for the design of foraging-inspired algorithms. The paradox points to the importance of a suitable balance between social and asocial learning in a population. The literature on social learning also emphasises that most animals employ more than one foraging strategy, and alter whom they choose to copy, and the balance with which they rely on social and asocial learning, depending on their internal state and the context of the current environment (Hoppitt and Laland, 2013). This suggests that adaptive algorithms, drawing on a library of known animal foraging strategies, could be particularly useful for solution finding in dynamic environments.

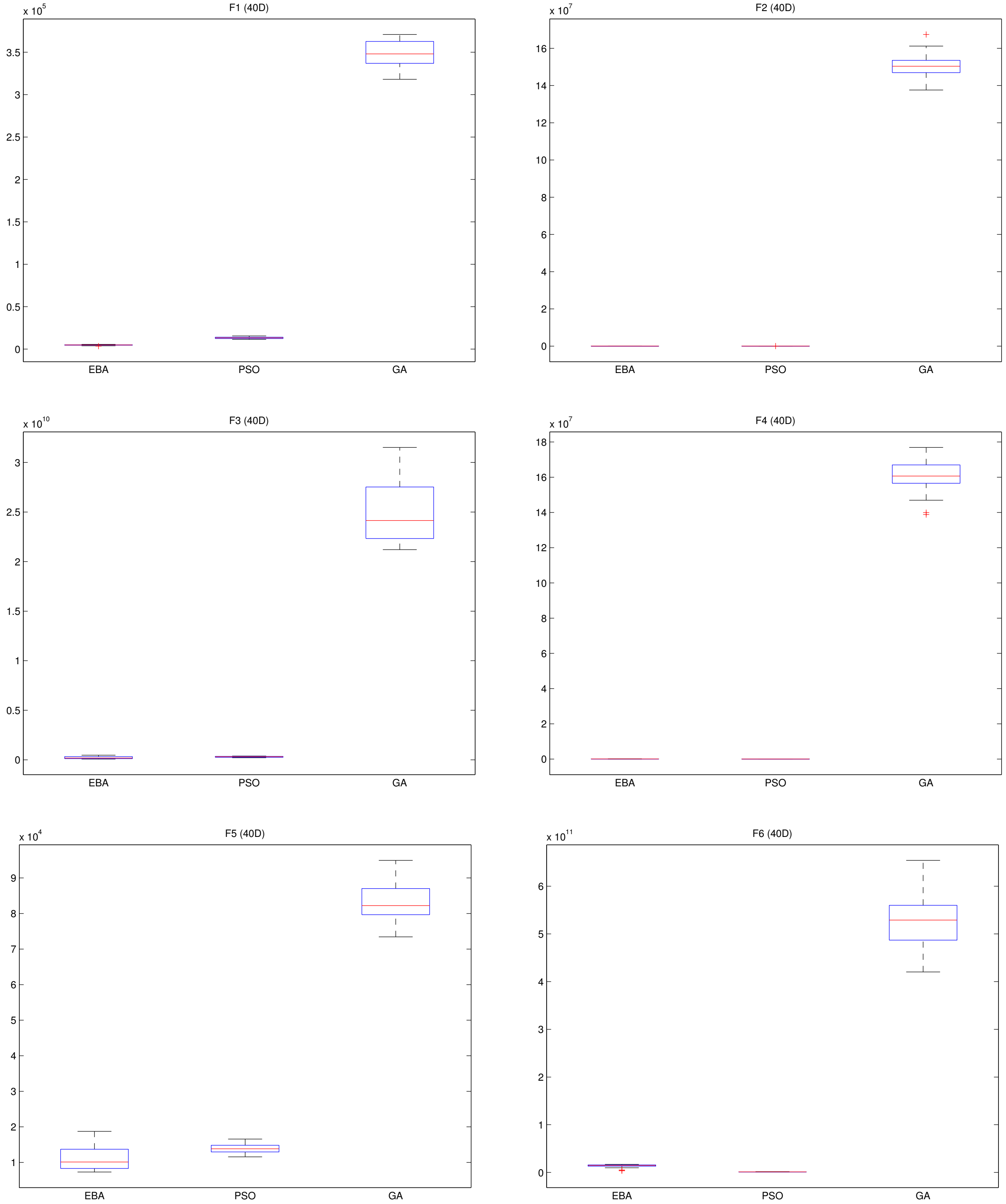

Figure 1 Average Best Fitness Across 30 Runs for the Benchmark Functions F1-F6 (N=30, D=40)

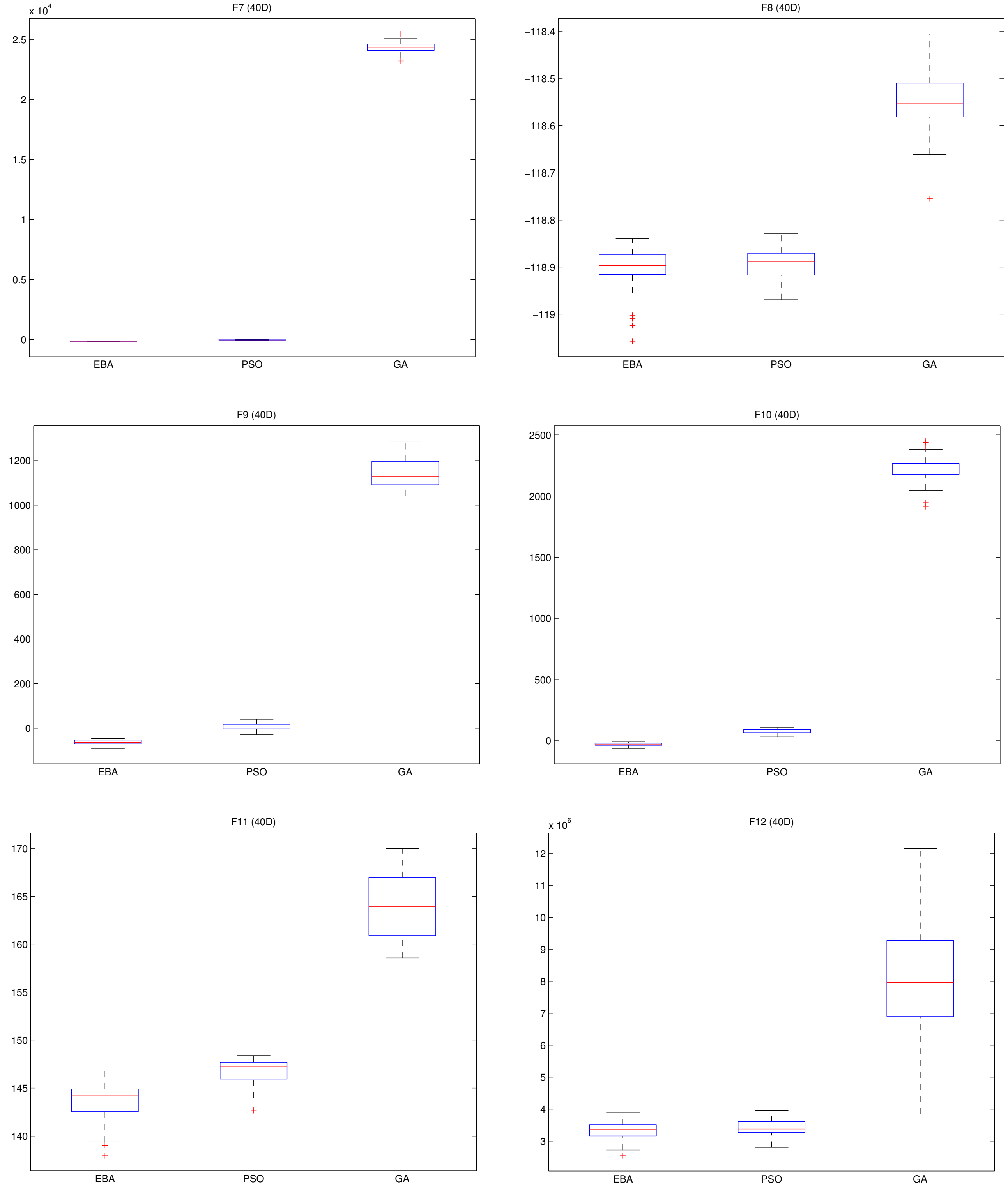

Figure 2 Average Best Fitness Across 30 Runs for the Benchmark Functions F7-F12 (N=30, D=40)

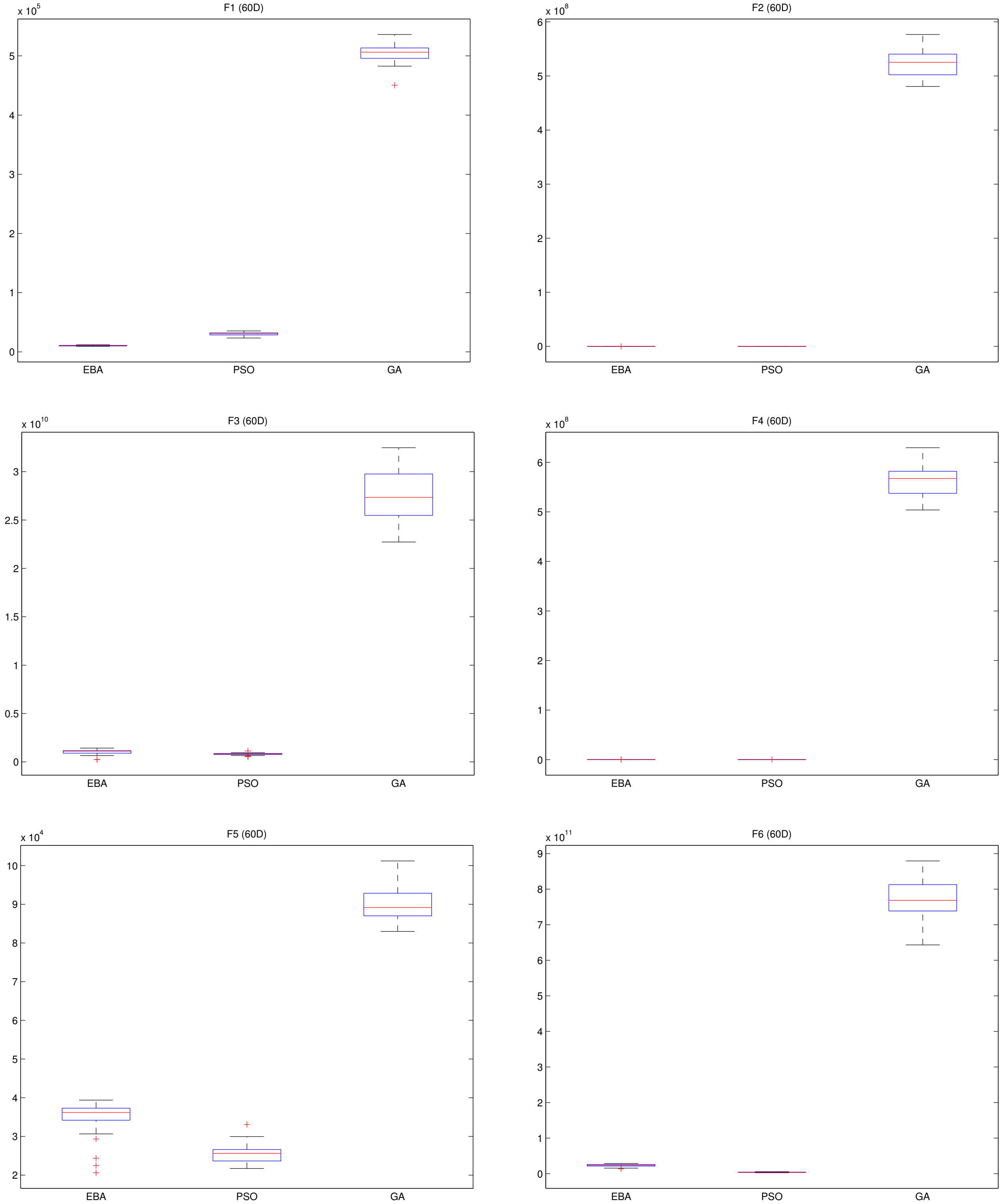

Figure 3 Average Best Fitness Across 30 Runs for the Benchmark Functions F1-F6 (N=30, D=60)

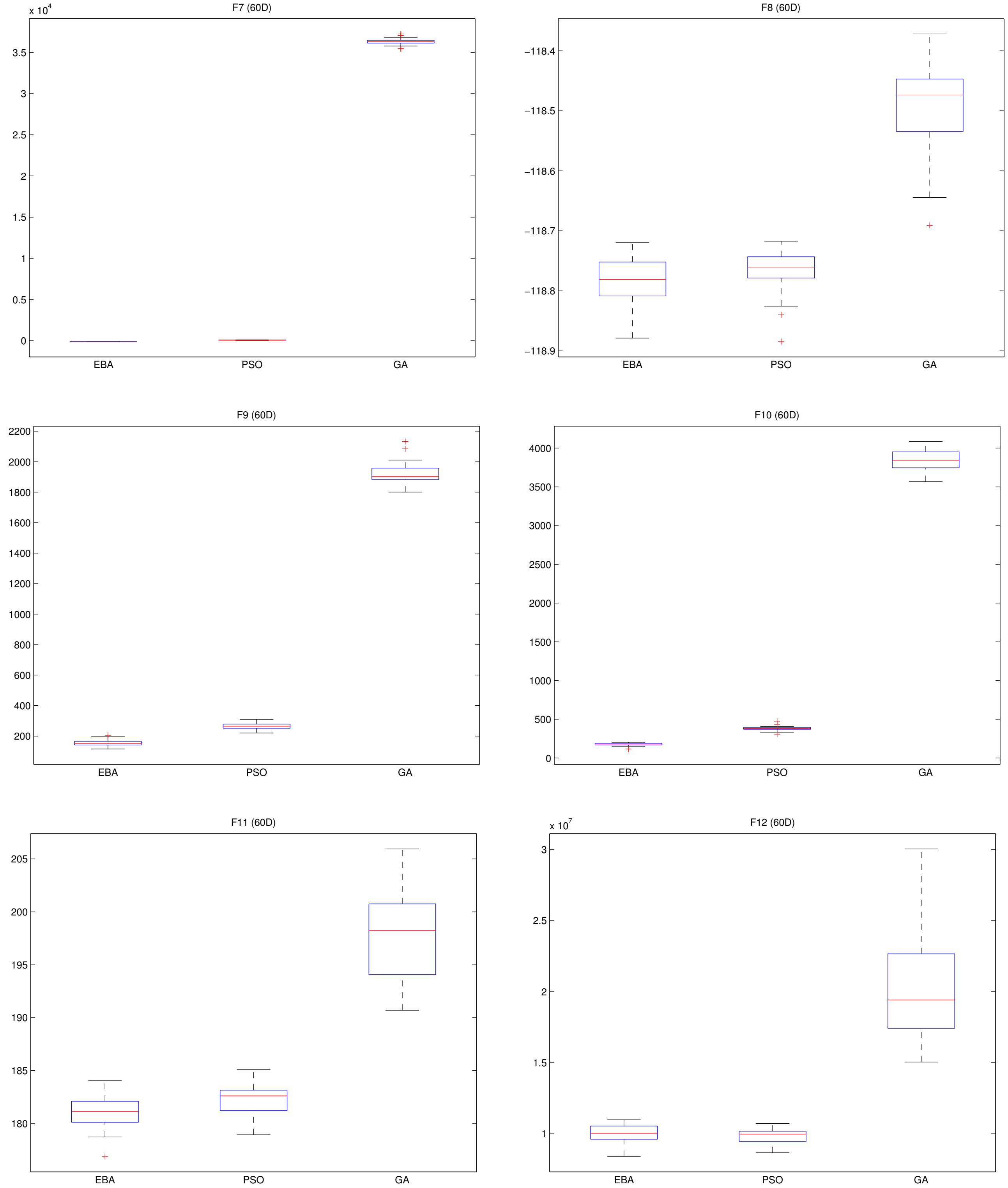

Figure 4 Average Best Fitness Across 30 Runs for the Benchmark Functions F7-F12 (N=30, D=60)

Table 4 Results of Algorithm Comparison (D=20). Best mean performance shown in bold.

		F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**F12**
EBA	**Best**	428.2378	639.258	7,701,057.35	677.5516	2,294.19	5,107,635	-175.8823	-119.3033	-253.5862	-257.025	107.9769	242,182.17
	Mean	**708.0956**	**1,176.78**	**13,261,519**	**1,518.05**	**2,897.94**	**15,534,495**	**-171.6913**	**-119.1969**	**-237.0346**	**-225.8455**	**111.9955**	317,815.42
	St.D.	151.7138	230.6098	3,155,706	319.9068	284.8901	4,801,054	1.3055	0.0509	7.4484	10.765	1.1541	31,990.65
vs. PSO	$p-value$	0	0	0	0	0	0	0	0.3442	0	0	0	0.8477
vs. GA	$p-value$	0	0	0	0	0	0	0	0	0	0	0	0
PSO	**Best**	1,179.73	2,955.46	12,924,035	2,280.32	3,925.74	22,459,519	-158.2391	-119.3475	-227.9241	-202.5852	111.4464	238,906.38
	Mean	2,521.06	4,110.20	38,321,951	4,145.58	4,983.26	92,221,055	-151.5417	-119.2102	-212.7773	-185.1564	113.4987	**316,176.36**
	St.D.	521.6977	590.4061	13,486,329	859.0715	498.7612	35,685,360	3.6005	0.0573	8.3421	8.2535	0.7758	33,809.97
GA	**Best**	137,517.64	14,881,234	13,788,889,580	16,188,622	41,868.52	97,896,293,156	8,765.69	-118.8864	192.4751	954.4524	121.6998	630,096.40
	Mean	153,880.34	18,460,009	18,532,968,075	19,622,626	48,958.38	137,959,084,479	9,044.43	-118.6485	264.5789	1,080.44	126.0775	1,340,545
	St.D.	10,906.38	1,667,132	2,290,938,680	1,233,916.	3,335.57	27,700,196,959	218.1268	0.0858	38.602	74.7103	2.2436	440,055.99

Table 5 Results of Algorithm Comparison (D=40). Best mean performance shown in bold.

		F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**F12**
EBA	**Best**	3,471.47	11,236.08	78,816,248	20,185.16	7,305.74	3,636,455,680	-145.1716	-119.0575	-91.1317	-63.6309	137.9536	2,544,066
	Mean	**4,738.39**	**33,691.64**	**216,317,530**	81,536.45	11,233.66	13,590,346,206	**137.9159**	**-118.9082**	**-64.0809**	**-29.1638**	**143.6124**	**3,305,079**
	St.D.	495.7435	21,804.80	114,117,372	28,352.57	3,382.50	3,534,733,328	3.4378	0.0526	12.6786	12.7743	2.1759	331,439.42
vs. PSO	$p-value$	0	0.1939	0.0005	0	0.0002	0	0	0.2041	0	0	0	0.3601
vs. GA	$p-value$	0	0	0	0	0	0	0	0	0	0	0	0
PSO	**Best**	11,391.56	28,190.46	213,100,264	35,568.58	11,556.67	976,554,443	-73.1379	-118.9692	-29.6469	30.3937	142.6781	2,801,153
	Mean	13,228.15	39,118.65	301,590,304	**51,659.43**	13,850.10	**1,261,243,631**	-34.1505	-118.8933	7.9481	76.7111	146.7097	3,383,275
	St.D.	1,047.47	5,993.02	52,968,599.8	9,276.38	1,344.10	160,773,177	18.2761	0.0358	16.7116	18.7017	1.3752	325,168.72
GA	**Best**	318,050.7	137,596,165	21,209,084,646	138,772,667	73,423.38	420,259,817,154	23,205.62	-118.7547	1,041.22	1,914.24	158.5732	3,846,763
	Mean	347,608.65	150,140,983	25,051,056,390	160,354,133	83,100.05	526,904,388,789	24,327.48	-118.5514	1,144.44	2,216.18	164.1693	8,261,784
	St.D.	14,317.05	6,314,061	3,245,666,865	9,260,832	5,037.62	53,430,189,982	497.6315	0.0708	69.6502	121.4512	3.4326	2,034,001

Table 6 Results of Algorithm Comparison (D=60). Best mean performance shown in bold.

		F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**F12**
EBA	**Best**	9,249.13	111,001.98	241,367,102	163,430.68	20,609.90	13,365,083,520	-97.0096	-118.8788	115.513	116.8995	176.8736	8,410,354
	Mean	**10,396.85**	233,848.88	997,910,202	293,978.69	34,553.97	23,361,520,577	**-82.024**	**-118.7818**	**155.3122**	**178.7974**	**181.0621**	9,979,340
	St.D.	666.2775	3,4674.65	305,125,038	39,604.44	4,753.83	3,563,926,580	6.661	0.0359	22.1426	17.3134	1.615	601,612.46
vs. PSO	$p-value$	0	0	0.0026	0	0	0	0	0.1665	0	0	0.0062	0.3906
vs. GA	$p-value$	0	0	0	0	0	0	0	0	0	0	0	0
PSO	**Best**	23,260.92	123,353.74	547,017,234	167,307.58	21,710.89	2,592,890,999	42.8205	-118.8844	220.2675	307.2836	178.9255	8,673,919
	Mean	30,066.11	**180,471.67**	**810,306,210**	**232,580.62**	**25,537.50**	**3,983,088,950**	73.3977	-118.7687	264.9804	380.9313	182.2257	**9,859,002**
	St.D.	2,371.23	37,953.84	116,634,978	55,854.28	2,420.37	735,905,424	18.9829	0.0368	20.8808	30.186	1.5572	467,541.26
GA	**Best**	450,361.20	480,671,568	22,732,181,837	503,840,666	83,000.91	643,278,743,103	35,432.39	-118.691	1,800.75	3,568.96	190.6897	15,048,090
	Mean	504,136.40	523,277,555	27,489,303,162	562,491,729	90,311.98	771,435,272,901	36,298.92	-118.4938	1,920.94	3,846.33	197.6501	20,261,034
	St.D.	16,841.97	22,814,187	2,636,979,477	31,443,478	4,898.29	55,887,093,629	435.6918	0.0718	70.4025	134.4552	4.2618	3,641,388

6 Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/SRC/FM1389. The authors would like to acknowledge the contribution of the anonymous reviewers to the improvement of this paper.

References

Altwaijry, N. and Menai, M. B. (2013) 'Artificial Bee Colony for the Standard Cell Placement problem', *Int. J. of Metaheuristics*, Vol. 2, No. 3, pp. 234–255.

Arita, H. and Fenton, B. (1997) 'Flight and echolocation in the ecology and evolution of bats', *Tree*, Vol. 12, No. 2, pp. 53–58.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford.

Bora, T., Coelho, L. and Lebensztajn, L. (2012) 'Bat-inspired Optimization Approach for the Brushless DC Wheel Motor Problem', *IEEE Transactions on Magnetics*, Vol. 48, No. 2, pp. 947–950.

Brabazon, A., O'Neill, M. and McGarraghy, S. (2015). *Natural Computing Algorithms*, Springer: Berlin, Germany.

Bradbury, J. and Vehrencamp, S. (2011). *Principles of Animal Communication* (2nd edition), Sinauer Associates: Sunderland, MA, USA.

Dechmann, D., Heucke, S., Giuggioli, l., Safi, K., Voigt, C. and Wikelski, M. (2009) 'Experimental evidence for group hunting via eavesdropping in echolocating bats', *Proceedings of the Royal Society B*, Vol. 276, pp. 2721–2728.

Dechmann, D., Kranstauber, B., Gibbs, D. and Wikelski, M. (2010) 'Group Hunting - A Reason for Sociality in Molossid Bats', *PLoS One*, Vol. 5, No. 2:e9012.

Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*, Ph.D. Dissertation, Politecnico di Milano.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) 'Ant system: optimization by a colony of cooperating agents', *IEEE Transactions on Systems, Man, And Cybernetics - Part B: Cybernetics*, Vol. 26, No. 1, pp. 29–41.

Dorigo, M. and DiCaro, G. (1999) 'Ant colony optimization: a new meta-heuristic', In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC 1999)*, IEEE Press, pp. 1470–1477.

Dorigo, M. and Stützle, T. (2004) *Ant Colony Optimization*, MIT Press, Cambridge, Massachusetts.

Fenton, B. (2013) 'Questions, ideas and tools: lessons from bat echolocation', *Animal Behaviour*, Vol. 85, No. 5, pp. 869–879.

Fister I Jr. and Fister I. and Yang X. and Fong S. and Zhuang Y. *Bat algorithm : recent advances*, 5th International Symposium on Computational Intelligence and Informatics, November 19-21, Budapest. 2014, pp. 163-167.

Gandomi, A., Yang, X.S., Alavi, A. and Talatahari, S. (2013) 'Bat algorithm for constrained optimization tasks', *Neural Computing and Applications*, Vol. 22, No. 6, pp. 1239–1255.

Giraldeau, L.A. and Caraco, T. (2000) *Social Foraging Theory*, Princeton University Press, New Jersey.

Griffin, D. (1944) 'Echolocation by blind men, bats and radar', *Science*, Vol. 100, No. 2609, pp. 589–590.

Griffin, D. (1958) *Listening in the Dark*, Yale University Press, New Haven, CT.

Jones, G. (2005) 'Echolocation', *Current Biology*, Vol. 15, No. 13, pp. R484–R488.

Jones, G. (2008) 'Sensory Ecology: Echolocation Calls are Used for Communication', *Current Biology*, Vol. 18, No. 1, pp. R34–R35.

Jones, G. and Teeling, E. (2006) 'The evolution of echolocation in bats', *Trends in Ecology and Evolution*, Vol. 21, No. 3, pp. 149–156.

Jones, G. and Siemers, B. (2011) 'The communicative potential of bat echolocation pulses', *Journal of Comparative Physiology A*, Vol. 197, No. 5, pp. 447–457.

Karaboga, D. (2005) 'An idea based on honeybee swarm for numerical optimization', *Technical Report TR06, Engineering Faculty, Computer Engineering Department, Erciyes University*, `http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf`, (accessed 20 May 2014).

Karaboga, D. and Basturk, B. (2007) 'A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm', *Journal of Global Optimization*, Vol. 39, No. 3, pp. 459–471.

Karaboga, D. and Akay, B. (2009) 'A survey: algorithms simulating bee intelligence', *Artificial Intelligence Review*, Vol. 31, No. 1-4, pp. 61–85.

Kennedy, J. and Eberhart, R (1995) 'Particle swarm optimization', In: *Proceedings of the IEEE International Conference on Neural Networks*, IEEE Press, pp. 1942–1948.

Kennedy, J., Eberhart, R. and Shi, T. (2001) *Swarm Intelligence*, Morgan Kaufman, San Mateo, California.

Kerth, G. and Reckardt, K. (2003) 'Information transfer about roosts in female Bechstein's bats: an experimental field study', *Proceedings of the Royal Society B*, Vol. 270, No. 1514, pp. 511–515.

Liang, J.J., Suganthan, P.N. and Deb, K. (2005) 'Novel composition test functions for numerical global optimization', In: *Proceedings of IEEE Swarm Intelligence Symposium*, IEEE Press, pp. 68–75.

Madsen, P., Aguilar de Soto, N., Arranz, P. and Johnson, M. (2013) 'Echolocation in Blainville's beaked whales (*Mesoplodon densirostris*)', *Journal of Comparative Physiol A*, Vol. 199, No. 5, pp. 451–469.

Nolen, T. and Hoy, R. (1986. 'Phonotaxis in flying crickets', *Journal of Comparative Physiology A*, 159:423-439.

Niknam, T., Azizipanah-Abarghooee, R., Zare, M. and Bahmani-Firouzi, B. (2013) 'Reserve Constrained Dynamic Environmental/Economic Dispatch: A New Multiobjective Self-Adaptive Learning Bat Algorithm', *IEEE Systems Journal*, Vol. 7, No. 4, pp. 763–776.

Nakamura, R., Pereira, L., Costa, K., Rodrigues, D., Papa, J. and Yang, X.S. (2012) 'BBA: A Binary Bat Algorithm for Feature Selection', In: *Proceedings of XXV SIBGRAPI Conference on Graphics, Patterns and Images*, IEEE Press, pp. 291–297.

Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. and Zaidi, M. (2006) 'The Bees Algorithm - A novel tool for complex optimisation problems', In: *Proceedings of International Production Machines and Systems (IPROMS 2006)*, Elsevier, UK, pp. 454–459.

Rendell, L., Fogarty, L., Hoppitt, W., Morgan, T., Webster, M., Laland, K. (2011). 'Cognitive culture: theoretical and empirical insights into social learning strategies', *Trends in Cognitive Sciences*, Vol. 15, No. 2, pp. 68–76.

Romero, P. and Robledo, F. (2007) 'An ant-colony approach for the design of optimal chunk scheduling policies in live peer-to-peer networks', *Int. J. Metaheuristics*, Vol. 2, No. 2, pp. 101–122.

Rui, T., Fong, S., Yang, X.S. and Deb, S. (2012) 'Nature-inspired Clustering Algorithms for Web Intelligence Data', In: *Proceedings of the IEEE, WIC, ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE Press, pp. 147–153.

Samarghandi, H. and Jahantigh, F. F. (2011) 'A particle swarm optimisation for fuzzy dynamic facility layout problem', *Int. J. of Metaheuristics*, Vol. 1, No. 3, pp. 257–278.

Stephens, D. and Krebs, J. (1986) *Foraging Theory*, Princeton University Press, Princeton, New Jersey.

Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A. and Tiwari, S. (2005) 'Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real-Parameter Optimization', Technical report, Nanyang Technological University, May 2005, `https://www.lri.fr/~hansen/Tech-Report-May-30-05.pdf`, accessed 20 May 2014.

Thorpe, W. (1963). *Learning and Instinct in Animals*, London: Methuen.

Viswanathan, G., da Luz, M., Raposo, E. and Stanley, E. (2011) *The Physics of Foraging: An Introduction to Random Searches and Biological Encounters*, Cambridge University Press, New York.

Wilkinson, G. (1992) 'Information transfer at evening bat colonies', *Animal Behaviour*, Vol. 44, No. 3, pp. 501–518.

Wynne, C., Udell, M. (2013). *Animal Cognition: Evolution Behavior and Cognition* (2nd ed.), Palgrave Macmillian, New York.

Yang, X.S. (2010) 'A New Metaheuristic Bat-Inspired Algorithm', In: *Proceedings of Fourth International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, pp. 65-74.

Yang, X.S. and Gandomi, A. (2012) 'Bat Algorithm: A Novel Approach for Global Engineering Optimization', *Engineering Computations*, Vol. 29, No. 5, pp. 464–483.

Yang, X.S. (2013) 'Bat Algorithm: Literature Review and Applications', *International Journal of Bio-Inspired Computation*, Vol. 5, No. 3, pp. 141–149.

Fister I. Jr. and Fong S. and Brest J. and Fister I. (2014) 'A Novel Hybrid Self-Adaptive Bat Algorithm, *The Scientific World Journal*, Vol. 2014, pp. 1–12.

Fister I. Jr. and Fister D. and Yang X. (2013) 'A hybrid bat algorithm, *Electrotechnical review*, Vol. 80, pp. 1–7.

Hoppitt, W. and Laland, K. (2013). *Social Learning: An introduction to mechanisms, methods and models*, Princeton University Press.

Yang X. (2011). *Bat algorithm for multi-objective optimisation*, Int. Journal of Bio-Inspired Computation, Vol. 3, No.5, pp. 267–274.