APPLICATION OF ADAPTIVE BATS SONAR ALGORITHM FOR SOLVING A SINGLE OBJECTIVE OF PRACTICAL BUSINESS OPTIMISATION PROBLEM

Nafrizuan Mat Yahya and Muhammed Nafis Osman Zahid

Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang
nafrizuanmy@ump.edu.my
nafis@ump.edu.my

Abstract

An adaptive bats sonar algorithm to solve single objective optimisation problem is presented. The proposed algorithm utilised the concept of echolocation of a colony of bats to find prey. The proposed algorithm is applied to solve two practical business optimisation problems. The problems are cost optimisation of shipping refined oil and profit optimisation of selling television sets. The acquired results show that the proposed algorithm suitable to produce the appropriate optimum solution of the considered problems. The proposed algorithm can thus be an effective method for solving of single objective optimisation problems.

Keywords: *adaptive bats sonar algorithm, swarm intelligence, single objective optimisation, practical business optimisation problems.*

1. Introduction

In general, optimisation is the process of obtaining either the best minimum or maximum result under specific circumstance (Yang and Deb, 2014). Nowadays, a vast range of business, management and engineering applications utilise the optimisation approach to save time, cost and resources while gaining better profit, output, performance and efficiency (Yang and Deb, 2014). Optimisation can be divided into single objective optimisation and multi objective optimisation (Rao, 2009). Naturally, solving a single objective optimisation is about finding an optimised solution to the problem at hand based on the single objective (Yang, 2011). The single objective optimisation can be designated as either unconstrained or constrained depending on whether or not the problem contains constraints (Rao, 2009). Con et al. (1997) elaborates the unconstrained single objective optimisation problem (or widely known as single objective optimisation problem) as a problem that has no constraints specified on the variables and usually is less complicated.

For the past decades, swarm intelligence algorithms raised a lot of attention from the research community to deal with the complexity of a wide variety of single optimisation problems (Yang and Hossien, 2012). Swarm intelligence algorithms are inspired by the collective behaviour of swarms through a complex interaction between individuals and their neighbourhood with nature such as a colony of ants, bacteria, bees, bats, birds and fishes (Hashmi et al., 2013). In general, swarms have self-organisation and decentralised control features and all the swarm follows the same system where a population of swarm cooperates and interacts with each other in the group and the environment under certain rules during foraging or socialising (Hashmi et al., 2013). The most remarkable features of any swarm intelligence algorithms are that is has advantages of memory, diverse multi-characters capability, rapid solution improvement mechanism and is adaptable to internal and external changes (Garg, 2014).

Particle swarm optimisation (PSO), artificial bee colony (ABC), ant colony optimisation (ACO), bat algorithm (BA) and bacterial foraging optimisation (BFO) for instance are some example of swarm intelligence algorithms that already captured the attention from the researchers today. This article applies an adaptive bats sonar algorithm (ABSA) as proposed by Yahya et al. (2016) for solving real-world single objective optimisation problems. The problems are cost optimisation of shipping refined oil and profit optimisation of selling television sets.

The remainder of the paper is organised as follows. Section 2 discusses the single objective optimisation problem. Section 3 describes the real bat echolocation behaviours and the ABSA. The results obtained from the computer simulation of the ABSA to solve two single objective of practical business optimisation problems are presented and discussed in Section 4. The conclusion is finally drawn in Section 5.

2. Single objective optimisation problem

A single objective optimisation is an objective function of *n* numbers of variables (*x*) that tie to lower bound and upper bound variables as:

$$\begin{aligned} &\textit{Optimise } F(x),\ x=(x_1,x_2,x_3) \\ &\textit{where} \\ &\qquad x_i^{(L)} \le x_i \le x_i^{(U)},\quad i=1,2,\dots,n \end{aligned} \tag{1}$$

Here $x_i^{(L)}$ represent the lower bounds and $x_i^{(U)}$ represent the upper bounds of variable x_i with n variables respectively.

Bandyopadhyay and Saha (2013) have identified three major techniques available to solve the single objective optimisation problem. The techniques are calculus-based techniques or numerical methods; enumerative techniques; and guided random techniques. According to Bandyopadhyay and Saha (2013), the guided random techniques are enumerative methods-improved where additional information about the search space is used to lead to potential solution points. The randomly guided techniques are further classified into single-point search and multi-point search. Swarm intelligence algorithms as part of evolutionary algorithms utilise the multi-point search where a highly explorative searching process with a random choice of parameters are adopted to search for several points at a time (Bandyopadhyay and Saha, 2013). These robust techniques have advantages to find acceptable near-optimum solution of the problems that have large search space, and are multimodal and discontinuous.

3. Bat echolocation

3.1. Real behaviour of bat echolocation

As one of the diverse and most extraordinary mammalian order, bats have more than 900 species distributed all around the world (Altringham et al., 1996). According to Voigt-Heucke et al. (2010), bats live in a large colony with 700-1000 individuals under sharing roosts.

The social calls and echolocation calls are two types of acoustic communication used by a colony of bats (Voigt-Heucke et al., 2010). A colony of bats can construct good communication and sharing information between each other about roost site or foraging area (Altringham et al., 1996). According to Altringham et al. (1996), there are four basic information transfer mechanisms in a colony of bats:

1. Intentional signalling: in the form of mating calls, territorial calls, alarm calls or food calls (advertisement of food and also to attract bats into foraging groups as they leave their cave roosts).

2. Local enhancement: involves unintentionally directing another bat to a specific part of the habitat.
3. Social facilitation: an increase in individual foraging success brought about by group foraging behaviour.
4. Imitative learning: bats can learn foraging techniques from other bats.

In echolocation, a bat emits ultrasonic pulses in short burst through mouth Altringham et al. (1996) as shown in Figure 1. The sound reflects back as echoes bump into an object in the bat's path. Altringham et al. (1996) and Suga (1990) agreed that by computing the time of reflection of modulates echoes, the bat is able to recognise the object and its distance.

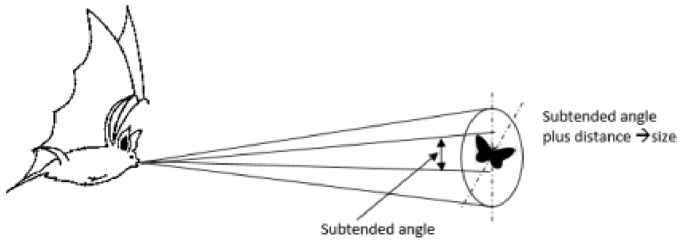


Figure 1: Sonar signal of a bat (Suga, 1990)

The echolocation process of bats involves three phases to search and capture prey: search phase, approach phase and terminal phase (Altringham et al., 1996). During the search phase, the bat will start to hunt for prey by emitting the pulse at the low rate with frequency around 10 *Hz*. Then, the pulses have to get shorter as the time between the pulse and echo is decreased to avoid overlap when the bat spots and gets nearer to the specific prey during the approach phase (Altringham et al., 1996; Suga, 1990). In this phase too, pulse emission rate gets steadily increased up to 200 per second since the bat keeps updating the position of the prey (Altringham et al., 1996; Suga, 1990). In the terminal phase, the frequency of emitted pulses upsurges more than 200 *Hz* as the pulse emission rate also starts to accelerate at only fraction of a millisecond long just before the prey is netted (Altringham et al., 1996).

A colony of bats has two special features during echolocation process to avoid them from colliding with one another. According to Vogler and Neuweiler (1983), the pulse characteristic (frequency range, the time course of sweep and sound type) emitted by each of bat differ from those of others. Second, every bat marks its emitting pulse with a unique time structure so that they only retrieve echoes caused by their pulses. The concept of reciprocal altruism of food sharing also exists during the echolocation process in a colony of bats (Altringham et al., 1996; DeNault and McFarlane, 1995; Wilkinson, 1988). This social behaviour is based on bats returning favours to their mutual benefits (Altringham et al., 1996). For instance, vampire bats species share the blood-meals between the individuals in a colony as a response to balance energy budget amongst in a colony (Altringham et al., 1996; DeNault and McFarlane, 1995). The bats successfully establish an individual survivorship in a colony after implementing this behaviour such that the fitness of the recipient is allocated comparatively to a non-recipient (Wilkinson, 1988).

3.2. An adaptive bats sonar algorithm

An adaptive bats sonar algorithm (ABSA) was proposed by Yahya et al. (2016). The purpose of ABSA is to solve single objective optimisation problems. In ABSA, the *number of iterations* (*MaxIter*) or generations used is 100. Hundred generations are favourably enough for the bats to explore fully the *d* numbers of search space *dimension* (Dim) for the best prey or *global best fitness*, (F_{GB}). The chosen value is in line with maximum *MaxIter* which was used in the particle swarm optimisation (PSO) algorithm when the algorithm was first introduced by Kennedy and Eberhart (1995).

Inspired by a description of the number of bats in a colony by biologists, the *number of bats* (*Bats*) or population in ABSA was selected in the range 700-1000 bats. By having a larger number of bats, a discovery of the F_{GB} value becomes more resourceful such that there will be a pool of solutions (prey) that can be evaluated to obtain the best ones.

In the ABSA, the *solution range* (SS_{size}) is defined as the value between the *upper search space* (SS_{Max}) limit and the *lower search space* (SS_{Min}) limit as Equation 2:

$$SS_{size} = SS_{Max} - SS_{Min} \tag{2}$$

Then, the ABSA sets the *beam length* (*L*) in relation to SS_{size} as Equation 8:

$$L \leq Rand \times \left(\frac{SS_{size}}{10\% \times Bats} \right) \tag{3}$$

The solution range is divided into micron scale, such as 10% of the overall population of bats in the search space. The percentage is marked as possible search space size of each bat to emit sound without colliding with one another. The value of *L* is different for every iteration. A *momentum term* (*m*) is used in ABSA as Equation 9:

$$L_{new} = L_{old}(1 \pm \mu) \tag{4}$$

where $0 < \mu < 1$. The above has been used by Yahya et al. (2016) to control the risk of convergence to a local optimum.

Altringham et al. (1996) and Suga (1990) have reported that the pulse emission rate grows bit by bit up to 200 per second as the bat keeps updating the location of the object until it catches the prey. This phenomenon is incorporated into the ABSA approach as *beam number increment* (*BNI*). The *BNI* is defined in terms of the *maximum number of beams* ($NBeam_{Max}$) and *minimum number of beams* ($NBeam_{Min}$) as Equation 5:

$$BNI = \left(\frac{NBeam_{Max} - NBeam_{Min}}{MaxIter} \right) \times iter \tag{5}$$

where $NBeam_{Max} = 200$ and $NBeam_{Min} = 20$. Thus, *NBeam* is defined as Equation 6:

$$NBeam = NBeam_{Min} + BNI \tag{6}$$

The *BNI* method mimics the original pulse rate emitted by the bat as it increases gradually toward the end of the search. As a result, *BNI* will provide a balance between global exploration and local exploitation thus requiring less iteration on average to find a sufficiently optimum solution.

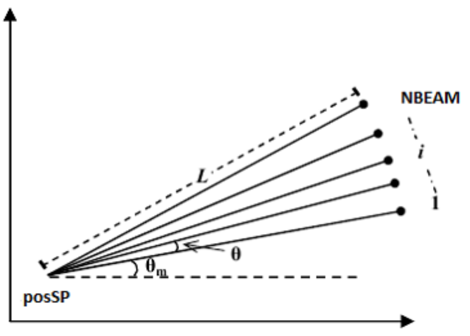


Figure 2: Single batch of beams transmitted by a bat (Yahya et al., 2016)

Each *NBeam* with *L* is emitted from the *starting position* (pos_{SP}) with specific angle location, see Figure 2. In ABSA, Yahya et al. (2016) limits the first beam to have θ_m not more than 45° from horizontal axis and the *angle between beams* (θ_i) is set as Equation 7 follows:

$$\theta_i = \frac{(2\pi - \theta_m)}{NBeam} \tag{7}$$

where $\theta_m = rand \leq 0.7854$.

By setting θ_i as such, the beams will sweep at random 360° around the bats through iterations in such a way that the searching process will neither be too aggressive (overlay a circle) nor too slow (underlay a circle).

The *end point position* (pos_i) for each transmitted beam in ABSA is calculated as Equation 8:

$$pos_i = pos_{SP} + L\cos[\theta_m + (i-1)\theta] \tag{8}$$

where $i = 1, \ldots, N$.

In ABSA, there are four stages of best fitness solution found in the algorithm. The duo are mentioned before; F_{LB} and F_{GB}, while another two levels are *starting position fitness* (F_{SP}) and *regional best fitness* (F_{RB}).

During the first iteration of ABSA, pos_{SP} of F_{SP} for each bat to transmit the *NBeam* is randomly selected within the designated search space. Next, the pos_i for each transmitted beam from pos_{SP} of each bat will be evaluate to produce *end point fitness* (F_i) where the best F_i is declare as F_{LB} and its position as *local best position* (pos_{LB}) of each bat. Later, the F_{SP} and F_{LB} of each bat is compared where the best will be F_{RB} and its position as *regional best position* (pos_{RB}). Finally, the best of the F_{RB} will be declared as F_{GB} and its position as *global best position* (pos_{GB}). According to Engelbrecht (2005), there are three levels of best solution found by the algorithm in PSO. The levels are *personal best* (*pb*) which is the best solution for every particle, *local best* (*lb*) which is the neighbourhoods best solution and *global best* (*gb*) is the global best solution of among the *pb*. These three levels are similar to F_{LB}, F_{RB} and F_{GB} of ABSA respectively.

In PSO, the *lb* improve the overall performance of algorithm where the individual *lb* influenced the performance of immediate neighbours (Kennedy, 1999). Ultimately, the neighbourhoods preserve swarm diversity by hindering the flow of information through the network (Peer et al., 2003). This move prevents the particles from reaching the global best particle immediately or getting trap in a local optimum but allows them to explore larger search space (Peer et al., 2003). This beneficial element inspired the existence of F_{RB} which is functioning as neighbourhoods best solution-ABSA version. In addition, F_{RB} also forms the main link between F_{LB} and F_{GB} values. So F_{RB} acts as a leverage instrument to balance finely between exploration (diversification) and exploitation (intensification) processes of the algorithm and so to help the algorithm escape from premature convergence.

The initialisation of these levels will help the ABSA to refine the search for the best solution by a colony of bats in the search space in each step and leave out bad solutions immediately. As a result, the algorithm takes less time to converge to the optimum solution. In point of fact, Kennedy (1999) mentioned that many types of research show that communication between individuals within a group is important where the overall performance of the group is affected by the structure of the social network. Besides, Kennedy and Mendes (2002) argued that the distribution of information via distant acquaintances is crucial, such that it possesses information that a colleague might not. In conjunction to that, the four levels of the best solution created in ABSA ideally match with the information transfer mechanisms practised by a colony of bats as explored by Altringham et al.

(1996). These are intentional signalling match to *FSP*, local enhancement match to F_{LB}, social facilitation match to F_{RB} and imitative learning match to F_{GB}.

Algorithm 1 Adaptive bats sonar algorithm

Objective function $F(x), x = (x_1, \ldots, x_d)^T$
Initialise: *Bats*, *MaxIter*, *Dim*, *SSSize*, $NBeam_{MAX}$ and $NBeam_{MIN}$
for $n \leftarrow 1$ **to** *Bats* **do**
  for $d \leftarrow 1$ **to** *Dim* **do**
    Generate random pos_{SP}
    Evaluate F_{SP} value for $F(pos_{SP})$
  end for
end for
Assign the most optimum value as F_{GB} and its position as pos_{GB}
while $t \leq MaxIter$ **do**
  Define *NBeam* to transmit by using *BNI* (**Equation 5** and **Equation 6**)
  Set L and limit μ (**Equation 3** and **Equation 4**)
  Generate random θ_m and θ_i (**Equation 7**)
  for $n \leftarrow 1$ **to** *Bats* **do**
    Transmit *NBeam* starting from pos_{SP}
      for $N \leftarrow 1$ **to** *NBeam* **do**
      for $d \leftarrow 1$ **to** *Dim* **do**
        Determine pos_i for each transmitted beam (**Equation 8**)
      end for
      Evaluate F_i value for $F(pos_i)$
      end for
    Assign the optimum value of F_i as F_{LB} and its position as pos_{LB}
      if $F_{LB} \leq F_{SP}$ **then**
        Assign F_{LB} as F_{RB} and pos_{LB} as pos_{RB}
      else
        Assign F_{SP} as F_{RB} and pos_{SP} as pos_{RB}
      end if
  end for
  Select the optimum value among F_{RB} as current F_{GB} and its pos_{RB} as current pos_{GB}
  if current $F_{GB} \leq$ previous F_{GB} **then**
    Update current F_{GB} as new F_{GB} and current pos_{GB} as new pos_{GB}
  else
    Retain previous F_{GB} and pos_{GB}
  end if
  for $n \leftarrow 1$ **to** *Bats* **do**
    Determine new pos_{SP} using (**Equation 9**)
    Evaluate new F_{SP} value for $F(x)$
  end for
end while
Declare F_{GB} as optimum fitness evaluated and pos_{GB} as its optimum value(s)

The reciprocal altruism characteristic has further been incorporated into ABSA to strengthen the procedure of colony searching for the best solution. This reciprocal altruism behaviour widely runs through a colony of bats as reported by many researchers in bats ecology (Altringham et al., 1996; DeNault and McFarlane, 1995; Wilkinson, 1988). By inserting this behaviour into the algorithm, a member of the colony will disseminate and share the location of the best fitness found so far to other bats. As a result, all bats will fly to the best prey ever found when the search process comes to an end. The adoption of this real prey hunting behaviour of the colony of bats into the algorithm is symbolised by two levels of arithmetic mean.

For every bat, the arithmetic mean evaluates the balancing point between pos_{SP}, pos_{LB} and pos_{RB} in current iteration (t) with pos_{GB} of the latest F_{GB} to be appoint as a new pos_{SP} for next iteration ($t+1$). The first level of arithmetic mean involves measuring of central tendency between pos_{SP}, pos_{LB} and pos_{RB} of each bat for current iteration only. Next, the second level of arithmetic mean finds the central tendency between the position value resulted from the first level of arithmetic mean and pos_{GB}. As a result, during new iteration, every bat will start to transmit a set of new beams from the pos_{SP} which has been specified after considering (or sharing) the balancing point of the positions of all four level of best fitness solutions; F_{SP}, F_{LB}, F_{RB} and F_{GB}. The two levels of arithmetic mean is expressed as Equation 9 follows:

$$pos_{SP}(t+1) = \frac{\frac{pos_{SP}(t) + pos_{LB}(t) + pos_{RB}(t)}{3} + pos_{GB}}{2} \qquad (9)$$

Based on these modifications, the basic steps of the ABSA are represented as the pseudo code in Algorithm 1.

4. Computer simulation

4.1. Introduction

This section demonstrates the capability of ABSA algorithm to solve single objective optimisation problems. The problems are cost optimisation of shipping refined oil and profit optimisation of selling television sets. The chosen problems became the platform to show the capability of the ABSA because both problems are the real-world single objective optimisation problems. Yahya et al. (2016) proposed the algorithm to solve the engineering optimisation problems only. Even so, the algorithm has merely been tested on the single objective optimisation benchmark test functions (Yahya et al., 2016). So, the application of the ABSA to the said problems will justify the performance of the algorithm to the real-world single optimisation problems as well as the non-engineering optimisation problems.

The developed algorithm was coded using MATLAB software version MATLAB® R2013a. Computer simulations of the ABSA algorithm on the two single objective optimisation problems were performed on Intel® Core™ i5 processor of 2400 CPU @ 3.10GHz with 4.00GB RAM.

4.2. Cost optimisation of shipping refined oil

This single objective optimisation problem is taken from Edgar et al. (2001). The problem is about finding the minimum cost of refined oil (F) when shipped via the Malacca Straits to Japan in dollar per kilolitre (*$/kL*). The optimum tanker size (x_1) in *dwt* and optimum refinery capacity (x_2) in *bbl/day* are variables of the problem.

The problem has to include the crude oil cost, insurance cost, customs cost, freight cost for the oil, loading and unloading cost, sea berth cost, submarine pipe cost, storage cost, tank area cost, refining cost and freight cost of products in the linear sum as (note that $1kL = 6.29bbl$):

$$Minimise\ \ F(x) = C_c + C_i + C_x + \frac{2.09e^4 (x_1)^{-0.3017}}{360} + \frac{1.064e^6 a (x_1)^{0.4925}}{52.47 (x_2)(360)} + \frac{0.1094 (x_1)^{0.671}}{360}$$
$$+ \frac{4.242e^4 a (x_1)^{0.7952} + 1.813 ip (n (x_1) + 1.2 (x_2))^{0.861}}{52.47 (x_2)(360)} + \frac{5.042e^3 (x_2)^{-0.1899}}{360}$$
$$+ \frac{4.25e^3 a (n (x_1) + 1.2 (x_2))}{52.47 (x_2)(360)}$$

where

$$\begin{aligned} a &= \text{annual fixed charges, fraction } (0.20) \\ c_c &= \text{crude oil price, \$/kL } (12.50) \\ c_i &= \text{insurance cost, \$/kL } (0.50) \\ c_x &= \text{customs cost, \$/kL } (0.90) \\ i &= \text{interest rate } (0.10) \\ n &= \text{number of ports } (2) \\ p &= \text{land price, \$/m}^2 \ (700) \\ x_1 &\geq 0 \text{ and } x_2 \geq 0 \end{aligned} \tag{10}$$

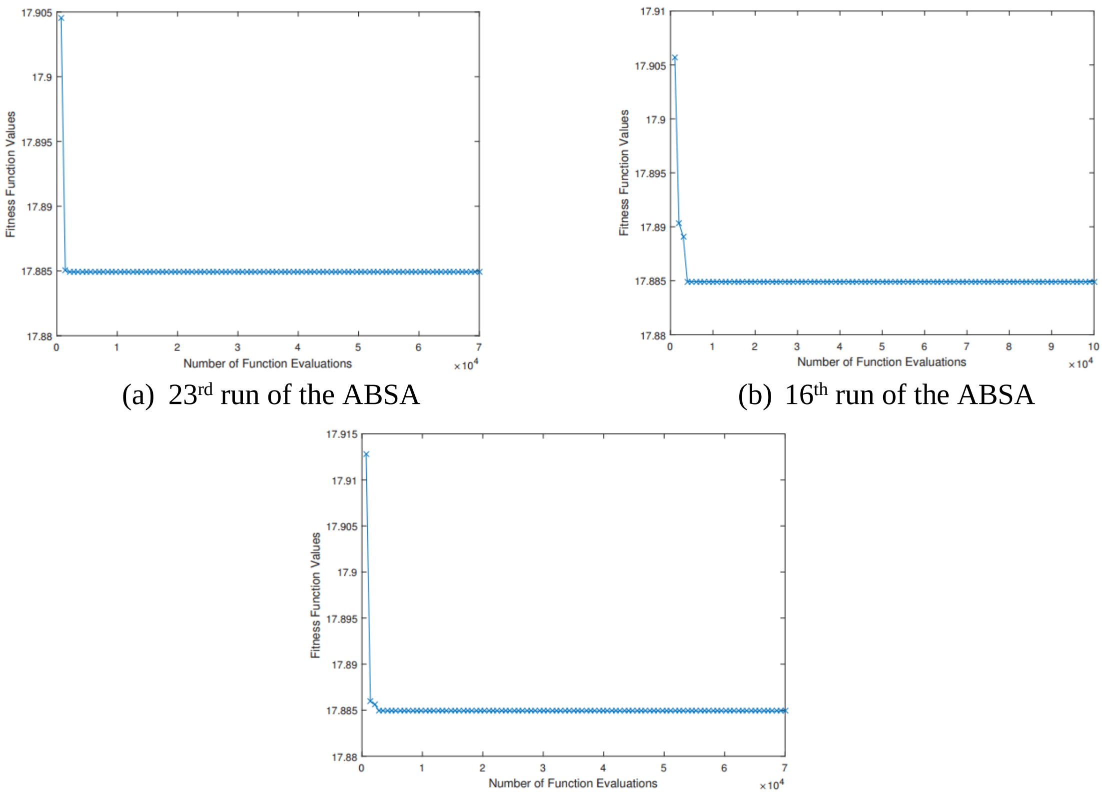
The ABSA is applied to find the optimum cost for this problem. The ABSA is capable of finding the minimum cost of refined oil (*F*) in dollar per kilolitre (*$/kL*). The results of 30 independent runs by the ABSA to solve this problem are shown in Table 1. According to the results, the minimum cost achieved by using ABSA is $17.8849*/kL*. The value was similar for all 30 independent runs, so the best, worst or mean are equal as well as standard deviation is zero.

The results also recorded that 53.33% out of 30 ABSA independent runs successfully finished in less than 10 *seconds*. 23th run of the algorithm as shown in Figure 3(a) appeared as the fastest among runs that are 5.0343 *seconds* where the ABSA started to converge to optimum value during 19th iteration. Meanwhile, the 16th of the ABSA as shown in Figure 3(b) finished the slowest among runs; 99.9512 *seconds* where the convergence only occurred during the 100th iteration. Figure 3(c) shows the 8th runs of ABSA where the algorithm started to converge to the optimum value in the shortest iteration among the all 30 independent runs, which was during 18th iteration. Finally, Figure 4 shows the quality of the obtained variables where small ranges of variation for the tanker size and refinery capacity were achieved in all 30 independent runs of the ABSA.

Table 1: Result for 30 runs of ABSA to optimise the cost of shipping refined oil problem

Run no.	Cost of shipping refined oil, *F* (*$/kL*)	Variables		Time to finish (*seconds*)	Numbers of bats used	Iteration to converge	Number of function evaluation (*NFEs*)
		Tanker size, x_1 (*dwt*)	Refinery capacity, x_2 (*bbl/day*)				
1	17.8849	446967.4908	179845.3736	5.8591	700	21	70000
2	17.8849	446967.5156	179845.3803	5.4619	700	20	70000
3	17.8849	446967.5103	179845.3674	5.7646	700	21	70000
4	17.8849	446967.4991	179845.3667	13.5368	700	37	70000
5	17.8849	446967.5089	179845.3761	39.3762	1000	58	100000
6	17.8849	446967.5251	179845.3873	5.4673	700	20	70000
7	17.8849	446967.4977	179845.3759	11.4670	1000	26	100000
8	17.8849	446967.5080	179845.3874	17.8500	700	18	70000
9	17.8849	446967.5210	179845.3825	6.8512	856	20	85600
10	17.8849	446967.5104	179845.3894	5.4480	700	20	70000
11	17.8849	446967.5057	179845.3770	35.6761	983	55	98300
12	17.8849	446967.5036	179845.3764	38.2098	1000	57	100000
13	17.8849	446967.4864	179845.3969	7.1871	1000	19	100000

14	17.8849	446967.5182	179845.3793	11.3154	1000	26	100000
15	17.8849	446967.5110	179845.3752	28.1802	700	59	70000
16	17.8849	446967.5138	179845.3800	99.9512	1000	100	100000
17	17.8849	446967.5593	179845.3855	16.5342	876	36	87600
18	17.8849	446967.5286	179845.3780	27.1227	1000	46	100000
19	17.8849	446967.5190	179845.3755	8.2677	1000	21	100000
20	17.8849	446967.5027	179845.3721	5.4758	700	20	70000
21	17.8849	446967.4913	179845.3769	7.8470	1000	20	100000
22	17.8849	446967.5320	179845.3843	5.7775	700	21	70000
23	17.8849	446967.4972	179845.3779	5.0343	700	19	70000
24	17.8849	446967.4928	179845.3691	7.1951	1000	19	100000
25	17.8849	446967.5162	179845.3848	5.4591	700	20	70000
26	17.8849	446967.4817	179845.3711	5.4330	700	20	70000
27	17.8849	446967.5156	179845.3781	29.7657	1000	49	100000
28	17.8849	446967.5118	179845.3763	43.6045	898	65	89800
29	17.8849	446967.5176	179845.3795	16.4321	700	42	70000
30	17.8849	446967.5428	179845.3875	7.0305	884	20	88400



(a) 23rd run of the ABSA

(b) 16th run of the ABSA

(c) 8th run of the ABSA

Figure 3: Convergence performances toward optimum fitness function of optimising the cost of shipping refined oil problem

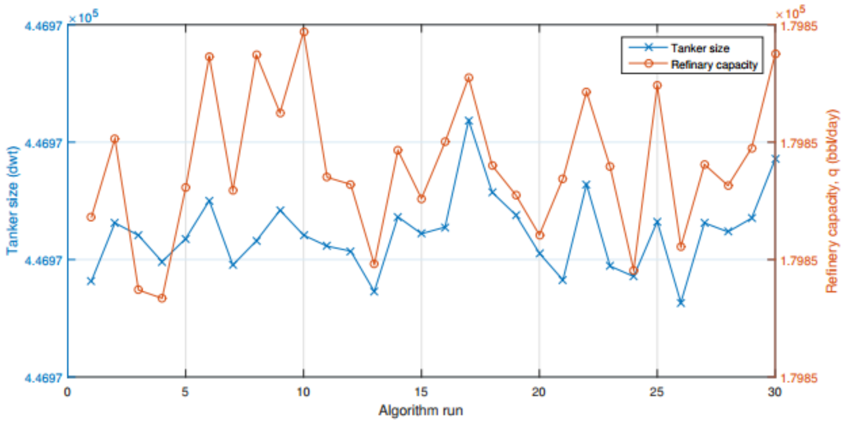


Figure 4: Tanker size and refinery capacity obtained in 30 independent runs of the ABSA to optimise the cost of shipping refined oil problem

4.3. Profit optimisation of selling television sets

This single objective optimisation problem is adopted from De Leon (2012). The problem is to estimate the maximum yearly profit (*F*) in *$/year* will be gained by the manufacturer of colour television (TV) sets when two types of TV sets are sold. There are two variables for this problem that are a number of 19" flat screen TV sets sell per year (x_1) and a number of 22" flat screen TV sets sell per year (x_2).

The problem has to consider the information such as:

- A manufacturer's suggested retail price (MSRP) of a 19" flat screen TV and a 21" flat screen TV are $339 and $399 respectively.
- A company cost to produce a 19" flat screen TV and a 21" flat screen TV are $195 and $225 respectively.
- A fixed cost of $400000.
- An estimation that for each type of TV set, the average selling price drops by $0.01 for each additional unit sold.
- An estimation that average selling price of the 19" flat screen TV will be reduced by an additional $0.003 for each 21" flat screen TV and the price of the 21" flat screen TV will be reduced by an additional $0.004 for each 19" flat screen TV sold.

The problem is formulated as:

$$\text{Maximise} \;\; F(x) = R(x) - C(x)$$

where

$$\begin{aligned}
&C(x) = 400000 + 195(x_1) + 225(x_2) \\
&R(x) = p(x)(x_1) + q(x)(x_2) \\
&p(x) = 339 - 0.01(x_1) - 0.003(x_2) \\
&q(x) = 399 - 0.004(x_1) - 0.01(x_2) \\
&\text{p} = \text{selling price of one 19" flat screen TV, \$} \\
&\text{q} = \text{selling price of one 21" flat screen TV, \$} \\
&\text{C} = \text{cost of manufacturing flat screen TV sets, \$/year} \\
&\text{R} = \text{revenue from sale of flat screen TV sets, \$/year} \\
&x_1 \geq 0 \;\text{ and }\; x_2 \geq 0
\end{aligned} \qquad (11)$$

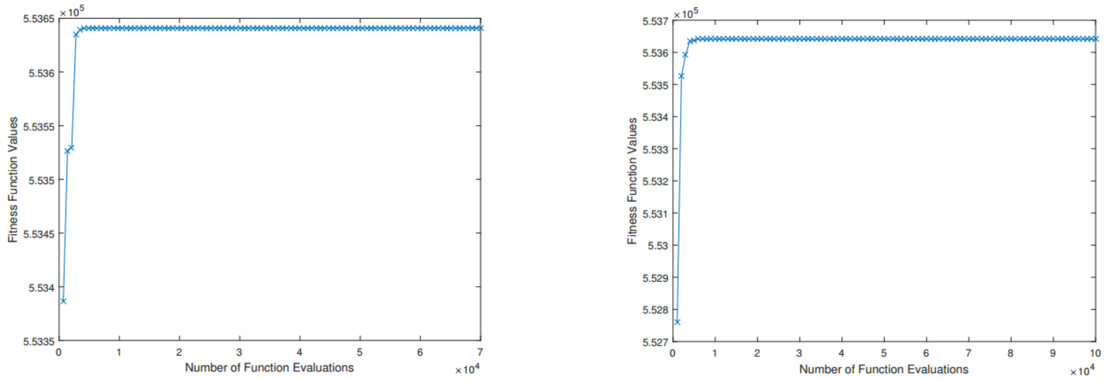
The ABSA is adopted to find the optimum profit for this problem. The ABSA is capable of estimating the maximum yearly profit (F) in *$/year* will be gained by a manufacturer of colour TV sets. Table 2 shows the results of 30 independent runs by the ABSA to solve this problem. All 30 independent runs of ABSA achieved a similar maximum profit of $553641.0256 by selling 4735 sets of 19" flat screen TV and 7043 sets of 21" flat screen. This mean that the best, worst or mean maximum profits are equal as well as standard deviation is zero.

In term of time for the algorithm to finish, the mean time taken by all 30 independent runs of ABSA to solve this problem is 11.910748 *seconds*. From 30 independent runs, 12th run recorded the fastest time, 2.158887 *seconds* and 4th run recorded the slowest time, 44.516322 *seconds* where the results are shown in Figure 5(a) and Figure 5(b) respectively. In addition, the 12th also ran the fastest it started to converge to the optimum value during 17th iteration out of 100 total iterations. 25th and 30th runs recorded the slowest and they started to converge to the optimum value where both only began during 97th iteration respectively.

To solve this problem, the ABSA randomly used 70000 to 100000 number of function evaluations (NFEs). As shown in Figure 6, the considered range of NFEs did not much affect the time for the algorithm to finish for all 30 independent runs. Except for 4th, 22nd, 25th and 30th runs, other independent runs of ABSA consistently recorded time below 25 *seconds*.

Table 2: Result for 30 runs of ABSA to optimise the profit of selling television sets problem

Run no.	Best fitnesss, F (*$/year*)	Variables		Time to finish (*seconds*)	Numbers of bats used	Iteration to converge	Number of function evaluation (*NFEs*)
		19" TV sets, x_1 (*unit sold/year*)	21" TV sets, x_2 (*unit sold/year*)				
1	553641.0256	4735	7043	5.8904	700	35	70000
2	553641.0256	4735	7043	3.8427	1000	20	10000
3	553641.0256	4735	7043	10.8029	854	44	85400
4	553641.0256	4735	7043	44.5163	1000	96	100000
5	553641.0256	4735	7043	15.5226	700	65	70000
6	553641.0256	4735	7043	6.7420	1000	30	100000
7	553641.0256	4735	7043	5.1212	1000	25	100000
8	553641.0256	4735	7043	4.5473	1000	23	100000
9	553641.0256	4735	7043	6.7754	1000	30	100000
10	553641.0256	4735	7043	3.8550	700	26	70000
11	553641.0256	4735	7043	4.3521	1000	22	100000
12	553641.0256	4735	7043	2.1589	700	17	70000
13	553641.0256	4735	7043	13.8235	1000	49	100000
14	553641.0256	4735	7043	5.3774	700	33	70000
15	553641.0256	4735	7043	22.2577	837	70	83700
16	553641.0256	4735	7043	20.6073	1000	63	100000
17	553641.0256	4735	7043	7.0210	1000	31	100000
18	553641.0256	4735	7043	6.3223	1000	29	100000
19	553641.0256	4735	7043	3.8573	700	26	70000
20	553641.0256	4735	7043	3.3606	762	21	76200
21	553641.0256	4735	7043	17.3687	1000	56	100000
22	553641.0256	4735	7043	28.7085	700	95	70000
23	553641.0256	4735	7043	4.0104	700	27	70000
24	553641.0256	4735	7043	3.4784	700	24	70000
25	553641.0256	4735	7043	39.5317	878	97	87800
26	553641.0256	4735	7043	12.7783	819	50	81900
27	553641.0256	4735	7043	6.1315	1000	28	100000
28	553641.0256	4735	7043	11.0377	1000	42	100000
29	553641.0256	4735	7043	7.7467	700	42	70000
30	553641.0256	4735	7043	29.7764	700	97	70000



(a) 12th run of the ABSA (b) 4th run of the ABSA

Figure 5: Convergence performances toward optimum fitness function of optimising the profit of selling television sets problem

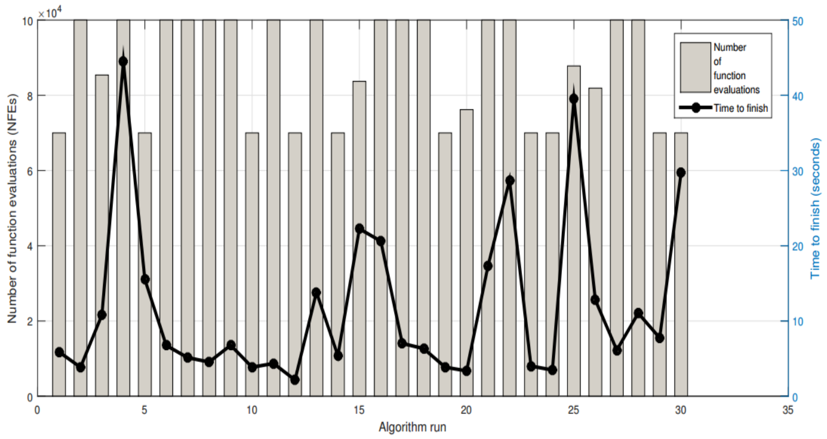


Figure 6: Number of function evaluations and time to finish recorded in 30 independent runs of the ABSA to optimise the profit of selling television sets problem

5. Conclusion

This paper has introduced an adaptive bats sonar algorithm (ABSA) to solve single objective optimisation problems. The single objective optimisation problems has been briefly defined respectively. The real echolocation behaviour of a colony of bats has been discussed. Then, ABSA was developed inspired from the bat echolocation. The proficiency of the ABSA to solve single objective optimisation problems has been examined through two practical business optimisation problems. The problems are; cost optimisation of shipping refined oil and profit optimisation of selling television sets. The computer simulation results have proved the ability of the ABSA to obtain the optimum results for both single objective optimisation problems.

References

Altringham, J. D. et al. (1996). *Bats: biology and behaviour*. Oxford, United Kingdom: The Oxford University Press.

Bandyopadhyay, S. & Saha, S. (2013). Some single-and multiobjective optimization techniques. In *Unsupervised Classification*, pages 17–58. Springer.

Conn, A. R., et al. (1997). Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79(1):397–414.

De Leon, D. (2012). *Unconstrained optimization with several variables*, in Math 232- Mathematical Models with Technology, Spring 2012 lecture notes, pages 1-9, Department of Mathematics, California State University, Fresno, USA. http://zimmer.csufresno.edu/doreendl/232.12s/handouts/multivaroptimization.pdf.[Online], accessed on October 6, 2015.

Deb, K. (2014). *Optimization for engineering design. Algorithms and examples*. New Delhi, India: PHI Learning Private Limited, 2nd edition.

DeNault, L. K. & McFarlane, D. A. (1995). Reciprocal altruism between male vampire bats, desmodus rotundus. *Animal Behaviour*, 49(3):855–856.

Edgar, T. F. et al. (2001). *Optimization of chemical processes*. University of Texas, USA: McGraw Hill, 2nd edition.

Garg, H. (2014). Solving structural engineering design optimization problems using an artificial bee colony algorithm. *Journal of Industrial and Management Optimization*, 10(3):777–794.

Hashmi, A. et al. (2013). Firefly algorithm for unconstrained optimization. *IOSR Journal of Computer Engineering*, 11(1):75–78.

Kennedy, J. (1999). *Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance*. In Evolutionary Computation (CEC'99), Proceedings of the 1999 Congress on, volume 3, pages 1931–1938, Indianapolis, USA.

Kennedy, J. & Eberhart, R. C. (1995). *Particle swarm optimization*. In International Conference on Neural Network IV, Proceedings of the IEEE, pages 1942–1948, Perth, Australia.

Kennedy, J. & Mendes, R. (2002). *Population structure and particle swarm performance*. In Evolutionary Computation (CEC'02), Proceedings of the 2002 Congress on, volume 2, pages 1671–1676, Honolulu, USA.

Peer, E. S. et al. (2003). *Using neighbourhoods with the guaranteed convergence pso*. In Swarm Intelligence Symposium (SIS'03), Proceedings of the 2003 IEEE, pages 235–242, Indianapolis, USA.

Rao, S. S. (2009). *Engineering optimization: theory and practice*. New York, USA: John Wiley & Sons, 4th edition.

Suga, N. (1990). Biosonar and neural computation in bats. *Scientific American*, 262(6):60–68.

Vogler, B. & Neuweiler, G. (1983). Echolocation in the noctule (nyctalus noctula) and horseshoe bat (rhinolophus ferrumequinum). *Journal of Comparative Physiology*, 152(3):421–432.

Voigt-Heucke, S. L. et al. (2010). A dual function of echolocation: bats use echolocation calls to identify familiar and unfamiliar individuals. *Animal Behaviour*, 80(1):59–67.

Wilkinson, G. S. (1988). Reciprocal altruism in bats and other mammals. *Ethology and Sociobiology*, 9(2):85–100.

Yahya, N.M. et al. (2016). 'A new bats echolocation-based algorithm for single objective optimisation', *Evolutionary Intelligence*, Vol. 9, pp.1–20.

Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 3(5):267–274.

Yang, X.-S. & Deb, S. (2014). Cuckoo search: recent advances and applications. *Neural Computing and Applications*, 24(1):169–174.

Yang, X.-S.& Hossein, G. A. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5):464–483.