electronics

MDPI

Article

On-Off Control of Range Extender in Extended-Range Electric Vehicle using Bird Swarm Intelligence

Dongmei Wu * and Liang Feng

Jiangsu Engineering Lab for IOT Intelligent Robots, College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, 210023 CHINA; 1218053626@njupt.edu.cn (L.F.)

* Correspondence: wudm@njupt.edu.cn

Received: 21 September 2019; Accepted: 22 October 2019; Published: 26 October 2019

Abstract: The bird swarm algorithm (BSA) is a bio-inspired evolution approach to solving optimization problems. It is derived from the foraging, defense, and flying behavior of bird swarm. This paper proposed a novel version of BSA, named as BSAII. In this version, the spatial distance from the center of the bird swarm instead of fitness function value is used to stand for their intimacy of relationship. We examined the performance of two different representations of defense behavior for BSA algorithms, and compared their experimental results with those of other bio-inspired algorithms. It is evident from the statistical and graphical results highlighted that the BSAII outperforms other algorithms on most of instances, in terms of convergence rate and accuracy of optimal solution. Besides the BSAII was applied to the energy management of extended-range electric vehicles (E-REV). The problem is modified as a constrained global optimal control problem, so as to reduce engine burden and exhaust emissions. According to the experimental results of two cases for the new European driving cycle (NEDC), it is found that turning off the engine ahead of time can effectively reduce its uptime on the premise of completing target distance. It also indicates that the BSAII is suitable for solving such constrained optimization problem.

Keywords: BSAII; Euclidean distance; energy management; E-REV

1. Introduction

Nowadays, society is facing the increasing depletion of petrochemical energy, the serious destruction of the ecological environment, and increasing car ownership. These factors promote the rapid development of new energy vehicles like the electric vehicle. However, the power battery of the pure electric vehicle has a series of problems, such as high cost, short range and over discharge, which is not conducive to long-distance driving.

As a transitional model of pure electric vehicle, the extended-range electric vehicle (E-REV) can effectively address the shortcomings above. The basic structure of a typical E-REV is shown in Figure 1. The auxiliary engine and power generation device has been added to the mechanism of the electric vehicle, which extends driving distance of electric vehicle. The integration of engine and generator constitute is called the range extender (RE), the main function of which is to charge the battery under the condition of insufficient power supply, for purpose of providing enough power to extend driving distance. Because of separation of the engine from the road load and the balance of the battery load, E-REV can keep the engine at the optimum working efficiency point (85%) and improve the fuel efficiency greatly. Additional E-REV has two energy sources: engine and power battery, so an efficient control strategy is essential to practice the coordination of the two devices, improve vehicle performance, e.g., fuel efficiency and exhaust pollution.

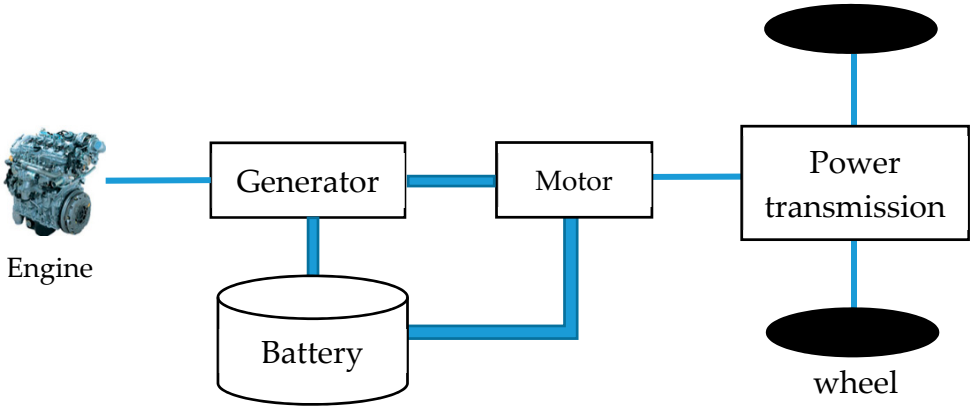


Figure 1. System structure of extended-range electric vehicle (E-REV).

The energy management of E-REV has always been a research hotspot [1–3]. Under the different driving conditions, the on-off time of RE for the E-REV is optimized with the target distance as the constraint condition. The main principle of the E-REV energy management strategy is that the use of engine is as little as possible as well as keeps the vehicle running in pure electric mode. The traditional control strategy is that when the battery power reaches the minimum threshold, the vehicle enters the extended range mode, the engine starts and drives the generator to produce electricity. Part of the generated electricity charges the battery, and the other part drives the vehicle to continue driving. When the battery power reaches the maximum threshold, the engine shuts down and vehicle enters the pure electric drive mode.

Control method like fuzzy control has been adopted in the energy management. It has been used for powering the battery, to keep the state of charge (SOC) in the designed threshold and avoid overcharge and over discharge [4]. As energy management can be considered an optimization problem, conventional planning methods were applied to the problem, such as dynamic programing, genetic algorithm (GA), and particle swarm optimization (PSO), etc. A hybrid genetic particle swarm optimization (GPSO) algorithm was proposed to optimize the parameters of energy management strategy [5]. In order to solve the problem of frequent start-stop of electric vehicle engines, a non-dominant sequencing genetic algorithm was used to optimize the start-stop interval of engines. The optimization effect of the running time of the extender under the two control modes of early opening and early closing is analyzed, in new European driving cycle (NEDC), urban dynamometer driving schedule (UDDS) cycles [6]. Energy management strategy of E-REV based on dynamic programming was designed, and optimal control rules of extender start-stop corresponding to SOC and motor power were established [7]. Driving behavior based on prediction of vehicle speeds was integrated into the energy management of the electric vehicle [8].

In recent years, with the unprecedented development of bionic optimization, a series of novel algorithms have emerged [9–17]. These include the teaching and learning optimization algorithm (TLBO, 2011) and its variants, the grey wolf optimizer (GWO, 2014) and its variants, the pigeon swarm algorithm (PSA, 2014), the whale optimization algorithm (WOA, 2016) and the bird swarm algorithm (BSA, 2016). Compared with GA, PSO and other mature algorithms, the optimization performance of these new bio-derived algorithms has been greatly improved. Therefore, the application of these algorithms in engineering attracted the attention of researchers.

The BSA as a novel algorithm, simulates the foraging behavior, defensive behavior and flight behavior of birds. It has the advantages of few parameters and it is easy to adjust. This paper extends the basic idea proposed in [17]. We propose a new method called BSAII with new coefficients for evaluating birds' ability to reach to the center. Then solve the optimization problem of engine on-off control in E-REV with the proposed algorithm.

The rest of paper will be organized as follows. Section 2 will outline the background to BSA, the formulation of motions for BSA, and its variants. We also propose another formulation of defense behavior of birds and explain the workflows of optimization with BSAII. Section 3 formulates the energy management in E-REV. Section 4 conducts extensive optimizing simulation, and analyzes the experiment results. Section 5 will present the experiment results of the application of BSAII on energy management of E-REV. Conclusions are drawn at the end of the paper.

2. Principle of Brid Swarm Intelligence

2.1. Bird Swarm Intelligence

Bird swarm foraging is easier to collect more information than individual foraging. It has survival advantages and good foraging efficiency. BSA is inspired by foraging behavior, defense behavior and flight behavior in the foraging process of birds. It is based on information sharing mechanism and search strategy in the foraging process of birds. The core of social behaviors and interactions in the bird swarm put forward a novel optimization algorithm BSA. Ideally, the basic principles of BSA can be elaborated as the following five rules [17].

(1) Each bird freely converts between defense and foraging behavior, which is a random behavior.

(2) In the process of foraging, each bird can record and update its own optimal information and global optimal information about the food. This information is used to find new sources of food. At the same time, the whole population share the social information.

(3) During the defense, each bird tries to move toward the center, but this behavior is influenced by competition among populations. Birds with high alertness are more likely to approach the center than low-alert birds.

(4) The swarm flies to another place each time. The identity of a bird converts between a producer and a beggar. That is, the most alert bird becomes a producer, while the lowest alert birds become a beggar. Birds with alertness between the two birds randomly become producers or beggars.

(5) Producers actively seek food, and beggars follow the producers at random.

The above five rules are described in mathematical terms as follows:

We suppose the size of the swarm is M, the number of dimensions is N. Foraging behavior in rule (1) is formulated;

$$x_i^{t+1} = x_i^t + c_1 r_1 (p_i - x_i^t) + c_2 r_2 (g - x_i^t) \tag{1}$$

where, x_i^t is the position of each bird, t represents the current number of iterations, i =1, 2... M. c_1 and c_2 are non-negative constants which represent cognitive and social acceleration coefficients independently. r_1 and r_2 are the random numbers with uniform distribution in [0,1]. p_i and g record the historical optimal location of the i^{th} bird and the historical optimal location of the whole swarm respectively.

According to the Rule (3), birds in the swarm are trying to get close to the central area, but there is a competitive relationship between birds. These behaviors can be expressed as follows;

$$x_{i,j}^{t+1} = x_{i,j}^t + A_1 r_3 \left(mean_j - x_{i,j}^t \right) + A_2 r_4 \left(p_{k,j} - x_{i,j}^t \right) \tag{2}$$

$$A_1 = a_1 \times e^{\left(-\frac{pFit_i}{sumFit+\varepsilon} \times M\right)} \tag{3}$$

$$A_2 = a_2 \times e^{\left(-\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \varepsilon} \times \frac{M \times pFit_k}{sumFit+\varepsilon}\right)} \tag{4}$$

Among them, a_1 and a_2 are the constants of [0,2], $pFit_i$ represents the optimal value of the i^{th} bird, $sumFit$ represents the sum of the optimal value of the whole swarm. ε is the smallest real number in a computer. $mean_j$ is average value of positions in the j^{th} dimension. r_3 is the random number between (0, 1), r_4 is the random number between (-1, 1). $k \neq i$. A_1 controls a bird approaching to center position of the whole swarm and $A_1 r_3 \in (0,1)$. A_2 represents the competitiveness of i^{th} bird versus k^{th} bird. The greater A_2 means compared with i^{th} bird, the k^{th} bird is more likely to move to the center of the swarm.

According to the Rule (4), every once in a while *FQ*, birds may fly to another place for seeking food, some birds may become producers, others will become beggars, behavior of producers and beggars are regulated their new position according to;

$$x_{i,j}^{t+1} = x_{i,j}^{t} + r_5 x_{i,j}^{t} \tag{5}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FLr_6(x_{k,j}^{t} - x_{i,j}^{t}) \tag{6}$$

r_5 is a Gaussian random number that satisfies the variance of 0 and the mean of 1. r_6 is the random number between (0, 1), and *FL* stands for the beggars getting food information from producers, $FL \in [0,2]$. The workflow of BSA for solving optimization problem is illustrated as Figure 2.

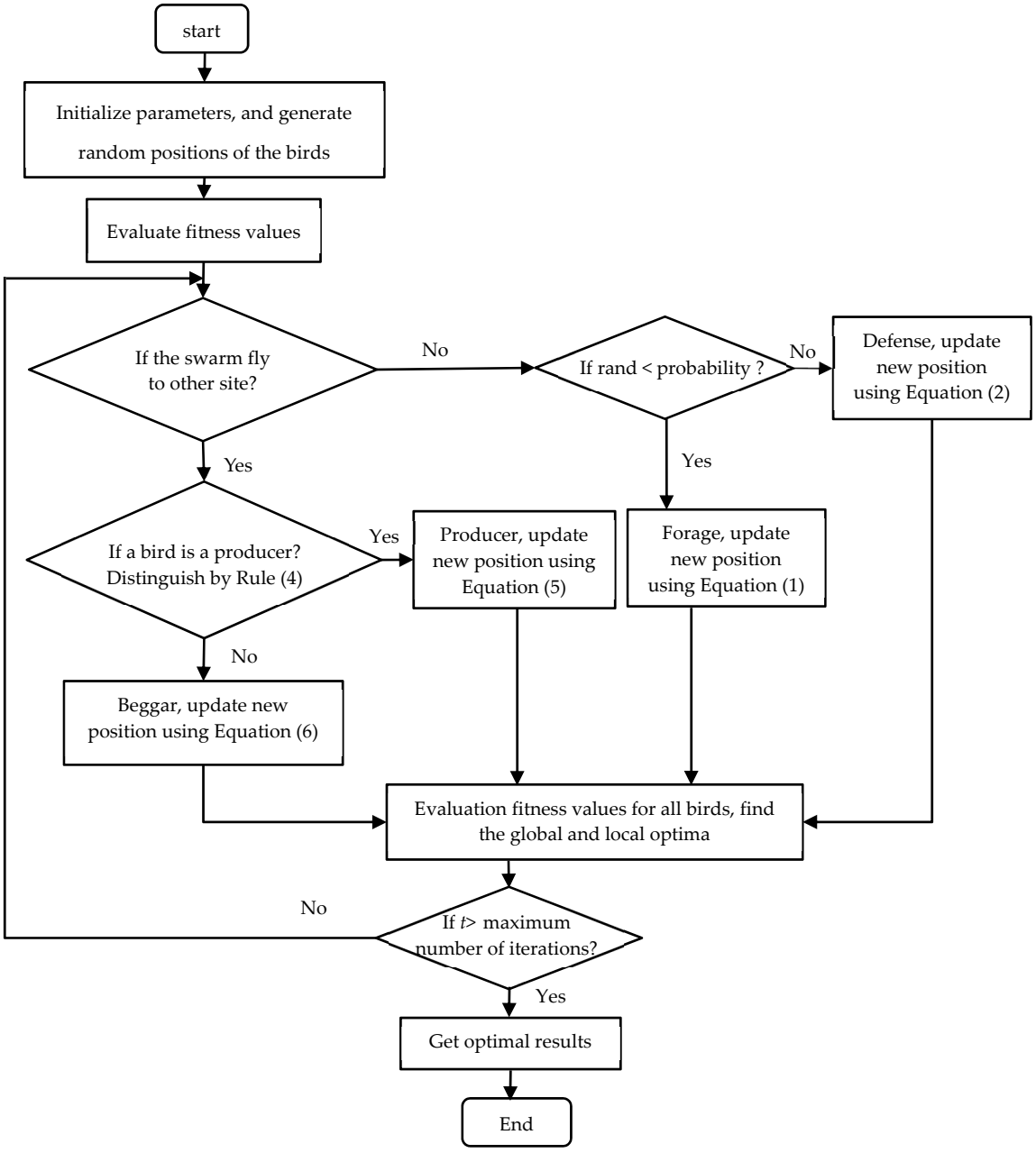


Figure 2. Workflow of bird swarm algorithm (BSA).

2.2. Related Improvement Methods

As a relatively new optimization algorithm, there is not much research on improvement of BSA. The algorithm is improved by defining inertia weight, with linear differential decline strategy, and linearly adjusting cognitive coefficient and social coefficient. Then different models are optimized

[18]. Levy flight strategy is applied to position initialization or iteration of BSA [19–21]. In [20], the random walk mode of Levy flight strategy increased the diversity of population and conduced to jumping out of local optimum. Inertia weight modified by random uniform distribution improved the search ability of BSA, besides, linear adjustment of cognitive and social coefficients was used to improve the solution accuracy. Boundary constraints were adopted to modify candidate solutions outside or on the boundary in the iteration process, which improves the diversity of groups and avoids premature problems. On the other hand, accelerated foraging behavior by adjusting the sine-cosine coefficients of cognitive and social components was achieved in [22].

2.3. BSAII

In the defensive state, a bird should not only move to the center as far as possible, but also compete with other neighbors. The parameters A_1 and A_2 are two factors that reflect ability of a bird moving to the center and competition with its neighbor bird respectively. In the traditional version of BSA, the fitness function was used to evaluate the weight coefficients of birds flying towards the center and affected by other birds. The function is one-dimensional, based on which the central position of the bird swarm is not accurate.

In this paper, we use spatial coordinates of birds to formulate A_1 and A_2. Based on the position of the bird group's center coordinate, the European distance between a bird's position and the center is calculated separately, and the traction and competitiveness of a bird flying toward the center are judged. We used other representations as;

$$A_1 = a_1 \times e^{pd_i} \tag{7}$$

$$A_2 = a_2 \times e^{\frac{pd_k - pd_i}{|pd_k - pd_i|}} \tag{8}$$

where pd_i is the normalized Euclidean distance between coordinates of a bird p_i and the center of the swarm *meanp*.

$$pd_i = norm(|p_i - meanp|) \tag{9}$$

An example of normalized Euclidean distance is shown in Figure 3. The red points are the four coordinate positions distributed in two-dimensional space, and the blue pentagonal star represents the central position determined by the average coordinate values of the four points, which is considered as the center point of the swarm. pd_1-pd_4 in the figure mean the normalized Euclidean distance between four points and the center one, which range from 0 to 1.

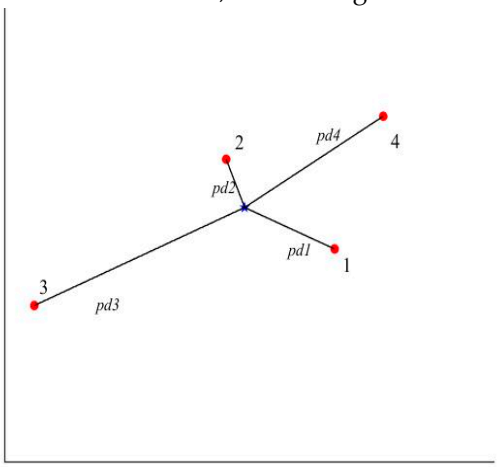


Figure 3. Normalized Euclidean distance in a two-dimensional (x-y) coordinate system.

Foraging and flying behaviour are formulized as Equations (1), (5) and (6), the same as in the previous version of BSA.

Initially, we set parameters, i.e., maximum number of iterations T, size of population M, flying interval FQ, c_1, c_2, a_1 and a_2, and created populations $\boldsymbol{x}$ randomly.

For each cycle, within each time interval, we only need to consider two behaviours of birds, foraging and defence. A bird behaviour is determined randomly, if the bird is looking for food, it would update position using Equation (1). Otherwise, the bird is on the defensive, and tries to move to the centre of the swarm. As each bird wants to fly to the centre, it is inevitable to compete with others. We used A_1 and A_2 related to normalized European distance to evaluate centralized flight of the bird, shown as Equations (7) and (8). Meanwhile the new position is regulated via Equation (2). If the swarm stays at one site for FQ, it needs to move to the next location as a whole. In the flying process, each bird plays a different role, i.e., beggar or producer. Birds move to new positions according to Equations (5) and (6) respectively. The outline of BSAII can be written as Algorithm 1.

Algorithm 1 BSAII

Step 1: Set parameters:
Population and Dimension of bird swarm [M, N].
Iteration T, flying interval FQ, a_1, a_2, FL, Foraging probability P, etc.

Step 2: Initialize the original positions of birds, $x_i = [x_{i1}, x_{i2}, \cdots, x_{iN}]$, i=1,2,…M.

Step 3: Calculate the fitness function $f(x_i)$, find local and global optimal solutions.

Step 4: For r = 1:T

1 **While** r is not an exact multiple of FQ **do**
  If (P_i<P)
    Bird forages for food according to Equation (1).
    Otherwise
    Bird conducts defensive action based on Equations (2), (7), and (8).
  End if
End while

2 Bird swarm is divided as producers and beggars, and flying to other site. The producers fly to new position of Equation (5), and the beggars follow the producers moving to Equation (6).

3 Calculate the fitness function $f(x_i)$, update the local and global optimal solutions.

Step 5: Output the optimal results.

3. Energy Management of E-REV

Energy management of E-REV in this paper mainly refers to optimize the on-off timing of RE in E-REV so as to reduce the running time of engine. This problem can be mathematically summed up as a constrained objective optimization problem. In this paper, the penalty function method is used to solve this optimization problem.

3.1. Constrained Optimization Problem

Optimization problem with constraints is formulized as,

$$\begin{aligned} &min\ f(x), x \in R^n \\ &\text{s.t. } h_i(x) = 0, i = \{1, 2, \cdots, l\} \\ &g_j(x) \geq 0, j = \{1, 2, \cdots, m\} \end{aligned} \tag{10}$$

h_i and g_j are equality and inequality constraints respectively. The feasible region Ω of the problem is defined as $\Omega = \{x \in R^n | h_i(x) = 0, g_j(x) \geq 0\}$. A popular method to solve constrained optimization problems is penalty function method. The penalty function [23] is constructed as;

$$\bar{P}(x) = \sum_{i=1}^{l} h_i^2(x) + \sum_{j=1}^{m} \left[min\{0, g_j(x)\}\right]^2 \tag{11}$$

Therefore, the objective function is transformed to;

$$P(x,\delta) = f(x) + \delta\bar{P}(x) \tag{12}$$

where $\delta > 0$ is penalty factor. The bigger δ is, the heavier the punishment will be. When $x \in \Omega$, x is a feasible point, $P(x,\delta) = f(x)$, the objective function is not subject to additional penalties. While $x \notin \Omega$, x is an infeasible point, $P(x,\delta) > f(x)$, the objective function is subject to additional penalties. When the penalty exists in the objective function, the penalty function should be sufficiently small to make $P(x,\delta)$ reach the minimum value, so that the minimum point of $P(x,\delta)$ approximates the feasible region Ω sufficiently, and its minimum value naturally approximates the minimum value of $f(x)$ on Ω sufficiently. The constrained optimization problem converts to unconstrained optimization problem, which is expressed as;

$$min\ \ P(x,\delta_k) \tag{13}$$

here δ_k is positive sequence, and $\delta_k \to +\infty$.

3.2. Problem Formulation

A certain type of electric vehicle is selected as research object, and basic parameters of vehicle are the same with that in [24]. Assuming that the SOC of battery power in electric vehicle should be kept between 20% and 80%. In order to reduce the uptime of engine in E-REV, and make full use of the power in the battery, this paper optimizes the uptime of the engine with constraint of distance. The objective function of the optimization problem is engine running time t, defined as Equation (14);

$$min\ \ t = \frac{L_{off} - L_{on}}{L} T_{cycle} \tag{14}$$

T_{cycle} is the time period, and L is the driving distance under one test condition, e.g., the NEDC cycle. We choose an E-REV as research object, whose main parameters are listed in Table 1.

Table 1. The main parameters for E-REV.

Parameters	Value
Curb weight (m/kg)	1700
Full mass (m/kg)	2100
Wheel radius (r/m)	0.334
Windward area (A/m^2)	1.97
Drag coefficients (CD)	0.32
Maximum speed (km/h)	>140
Total distance (km)	400
Climbing gradient (%)	>20%
Transmission efficiency	0.95
State of charge (SOC) range (%)	20–80
Maximum pure electric driving range (km)	>50

The vehicle parameters and component matching parameters (motor, battery, and RE) of the E-REV were entered into the advanced vehicle simulator (ADVISOR 2002) simulator, then the NEDC cycle condition was selected. The simulation results including SOC change curve under the NEDC cycle were obtained. The results showed an approximate linear relationship between the driving distance of electric vehicles and the SOC of battery.

$$y = kx + b \tag{15}$$

x is SOC, %. y is driving distance of E-REV, km. k and b are constant coefficients, in relation to battery working mode. When driving distance can be accomplished with one charge and discharge cycle, the distance when engine starts and shuts down are calculated based on Equations (16) and (17).

$$L_{on} = k_1(100 - t_{on}) \tag{16}$$

$$L_{off} = k_1(100 - t_{on}) + k_2(t_{off} - t_{on}) \tag{17}$$

t_{on} and t_{off} are timing of engine start-up and shutdown independently, which are represented as SOC. The equality constraint is the requirement of trip range distance D,

$$D = k_1(100 - t_{on}) + k_2(t_{off} - t_{on}) + k_3(t_{off} - 20) \tag{18}$$

k_i, i=1,2,3, are driving distance per unit charge under different conditions, the specific values are shown in Table 2.

Table 2. Driving distance per unit charge.

Distance of unit charge	Driving state	Value [km/%]
k_1	Initial pure electric	0.7310
k_2	Charging	0.3667
k_3	Pure electric after charging	0.7210

Inequality constraints satisfy,

$$t_{off} - t_{on} > 0$$

$$20 < t_{on}, t_{off} < 80$$

When the target distance exceeds one charge and discharge cycle, the engine repeatedly starts and closes. If there are n charge and discharge cycles in the whole trip. Evaluation is processed based on total uptime of the engine.

$$min \quad t = \frac{k_2[60(n-1)+(L_{off}-L_{on})]}{L} T_{cycle} \tag{19}$$

The target trip distance is calculated as Equation (20).

$$D = k_1(100 - t_{on}) + 60(n-1)(k_2 + k_3) + k_2(t_{off} - t_{on}) + k_3(t_{off} - 20) \tag{20}$$

4. Computational Experiment

In order to verify the effectiveness of BSAII algorithm, 20 benchmark functions were used in computational experiments, including unimodal and multimodal examples [25,26]. BSA, particle swarm optimization (PSO), artificial bee colony (ABC) and differential evolution (DE) were used as algorithms for comparison. In general, two aspects were taken into account to evaluate performance of algorithms: (1) proximity to the real optima in single operation, and (2) stability and accuracy of optimal results using different algorithms in multiple operations. Tables 3 and 4 illustrate concrete content of benchmark functions. In all cases, the population size was 50. The dimensions had two different types, dimension of population in f_4, f_6-f_8, f_{11}-f_{13}, f_{15}, f_{16} f_{18}-f_{20} were set to two, and in other cases were 20. The number of iterations was set to 1000. Other parameters used in simulation were tuned according to Table 5. *FQ* is the flying interval, and this was set to three. *FL* is the following coefficient, and this value was a random number between 0.5 and 0.9. c_3 and c_4 are acceleration constants. w is inertia weight linearly decreasing from 0.9 to 0.5 [27]. *CR* is crossover probability, and F is the mutation rate [28]. $foodnumber$ is the number of the food sources which is equal to the number of employed bees. *limit* is a predetermined number of cycles [29]. All algorithms were programmed with MATLAB 2018a. The simulation environment was on a computer with Intel ® core ™ i5-8400 CPU @ 2.80 GHz.

First of all, each algorithm ran once with 1000 iteration independently. Convergence performance on 20 benchmark functions is shown in Figure 4. Based on the graphical results, except for f_{11}, f_{14}, and f_{19}, BSAII could reach the real optima in other 17 functions. In addition, better results can be obtained by BSAII, compared with other four algorithms. Therefore, it can be shown that the performance of BSAII algorithm outperforms other algorithms.

Additionally, each algorithm was independently performed for 50 times on 20 test instances. Due to randomness of initial solutions for all algorithms, multiple performance indexes were used for comparing the performance of BSAII with different algorithms. Table 6 gives the minimum (MIN),

maximum (MAX), mean (MEAN) and standard variation (SD) of 50 trials on each case. We can make the following remarks from results of Table 6:

1. On instances of f_1~f_7, f_{12}, f_{17}, and f_{20}, BSAII performed better than other algorithms, in terms of convergence rate and accuracy of optimal solution. Since the four indexes had the smallest values compared with these obtained by other algorithms.
2. On instances of f_8, it was evident that both BSAII and DE could get the real optima (0 in Table 3), better than the other algorithms.
3. On instances of f_9 and f_{10}, BSAII and BSA were better than PSO, ABC, and DE in terms of performance indexes.
4. On instances of f_{13}, f_{15} and f_{18}, BSAII, BSA and DE converged to the real optimal value, with the same accuracy. But the convergence rate of BSAII and BSA in the early stage was faster than DE.
5. Only on instance of f_{11}, DE acquired the best performance over other algorithms. Solutions found by BSAII and BSA occasionally fell into local optimum.
6. Only on instance of f_{16}, BSA converged to the real optimal value (–3600) every time. It was obvious that BSA was better than the other algorithm on this example. While BSAII fell into local optimum at times and was not convergent to –3600. But the average value of 50 trials was closer to the optimum than that obtained by other algorithms.
7. The statistical results of f_{14} and f_{19} were somewhat complicated. The minimum of optimal fitness on f_{19} was found by BSAII, but the other three performance indexes of *MAX*, *MEAN*, and *SD* were slightly worse than in DE. BSAII would fall into local optimum when solving benchmark function of f_{14}.

Overall, we claimed that BSAII produced better convergence and more stable performance than BSA, PSO, ABC, and DE, in most cases.

Table 3. Benchmark functions.

Id.	Name	Dimension	Boundary	Optima
f_1	Sphere	20	[−100,100]	0
f_2	Sum of Different Powers	20	[−1,1]	0
f_3	Sum Squares	20	[–5.12,5.12]	0
f_4	Trid	2	[–4,4]	–2
f_5	Rotated Hyper-Ellipsoid	20	[–65.536,65.536]	0
f_6	Easom	2	[–100,100]	–1
f_7	Matyas	2	[–10,10]	0
f_8	Booth	2	[–10,10]	0
f_9	Griewank	20	[–600,600]	
f_{10}	Rastrigin	20	[–5.12,5.12]	0
f_{11}	Schwefel	2	[–500,500]	0
f_{12}	Shubert	2	[–5.12,5.12]	−186.7309
f_{13}	Schaffer Function No. 2	2	[–100,100]	0
f_{14}	Rosenbrock	20	[–2.048,2.048]	0
f_{15}	Beale	2	[–4.5,4.5]	0
f_{16}	Needle in a Haystack	2	[–5.12,5.12]	–3600
f_{17}	Zakharov	20	[–5,10]	0

Id.	Name	Dimension	Boundary	Optima
f_{18}	Drop-Wave	2	[–5.12,5.12]	–1
f_{19}	Bukin Function No. 6	2	$x_1 \in$ [–15,5], $x_2 \in$ [–3,3]	0
f_{20}	Three-Hump Camel	2	[–5,5]	0

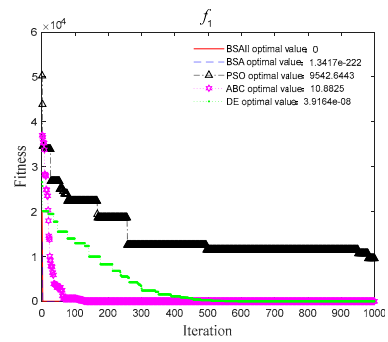
Table 4. Benchmark functions.

Id.	Function
f_1	$f_1(x) = \sum_{i=1}^{N} x_i^2$
f_2	$f_2(x) = \sum_{i=1}^{N} \lvert x \rvert^{i+1}$
f_3	$f_3(x) = \sum_{i=1}^{N} i x_i^2$
f_4	$f_4(x) = \sum_{i=1}^{N} (x_i - 1)^2 - \sum_{i=2}^{N} x_i x_{i-1}$
f_5	$f_5(x) = \sum_{i=1}^{N} \sum_{j=1}^{i} x_j^2$
f_6	$f_6(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$
f_7	$f_7(x) = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$
f_8	$f_8(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
f_9	$f_9(x) = \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1$
f_{10}	$f_{10}(x) = 10N + \sum_{i=1}^{N} [x_i^2 - 10\cos(2\pi x_i)]$
f_{11}	$f_{11}(x) = 418.9829N - \sum_{i=1}^{N} x_i \sin(\sqrt{x_i})$
f_{12}	$f_{12}(x) = (\sum_{i=1}^{5} i\cos((i+1)x_1 + i))(\sum_{i=1}^{5} i\cos((i+1)x_2 + i))$
f_{13}	$f_{13}(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$
f_{14}	$f_{14}(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
f_{15}	$f_{15}(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$
f_{16}	$f_{16}(x) = -(\frac{3}{0.05 + (x_1^2 + x_2^2)})^2 - (x_1^2 + x_2^2)^2$
f_{17}	$f_{17}(x) = \sum_{i=1}^{N} x_i^2 + (\sum_{i=1}^{N} 0.5 i x_i)^2 + (\sum_{i=1}^{N} 0.5 i x_i)^4$
f_{18}	$f_{18}(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$
f_{19}	$f_{19}(x) = 100\sqrt{\lvert x_2 - 0.01 x_1^2 \rvert} + 0.01 \lvert x_1 + 10 \rvert$
f_{20}	$f_{20}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2$

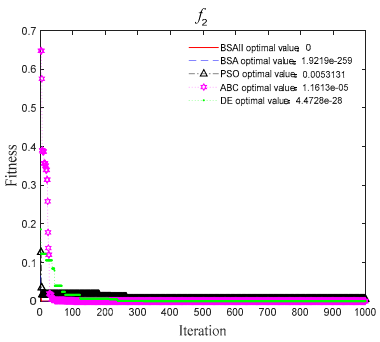
Table 5. Parameter setting.

Algorithm	Parameters
BSAII, BSA	$a_1 = a_2 = 1, c_1 = c_2 = 1.5, P \in [0.8,1], FL \in [0.5,0.9], FQ = 3$
PSO	$c_3 = c_4 = 2.0, w \in [0.5,0.9]$
DE	*CR*=0.9, *F*=0.5

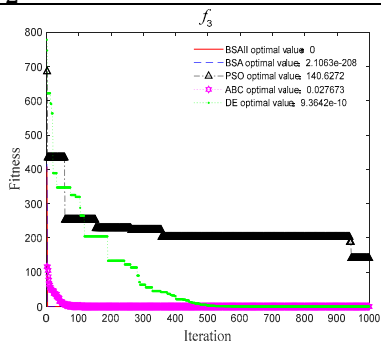
ABC	$foodnumber = \frac{M}{2}, limit = 10$



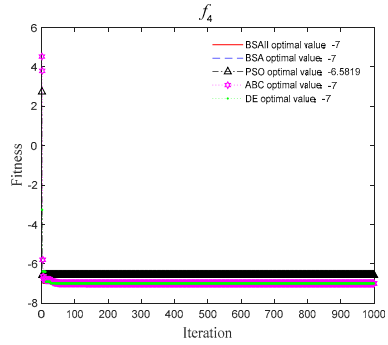
(a) f_1



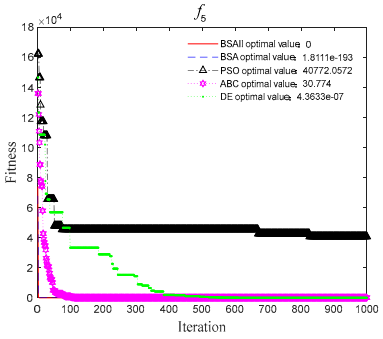
(b) f_2



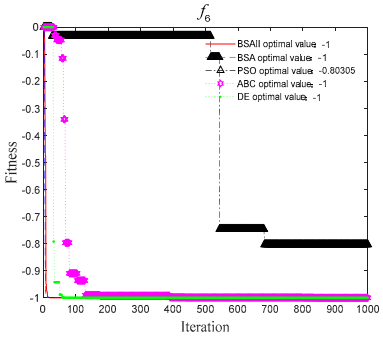
(c) f_3



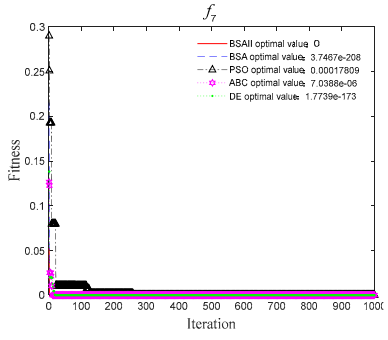
(d) f_4



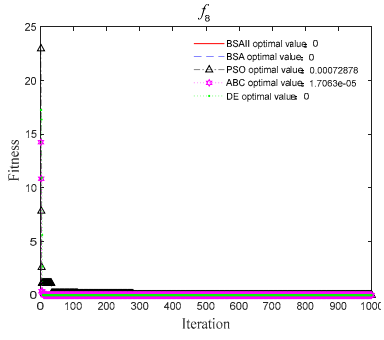
(e) f_5



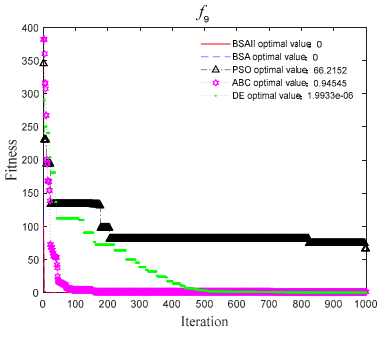
(f) f_6



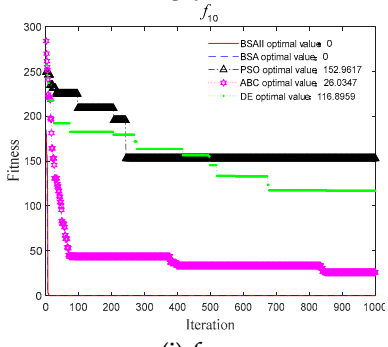
(g) f_7



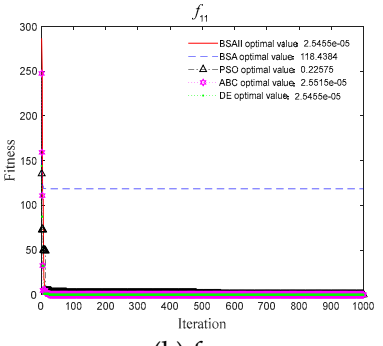
(h) f_8



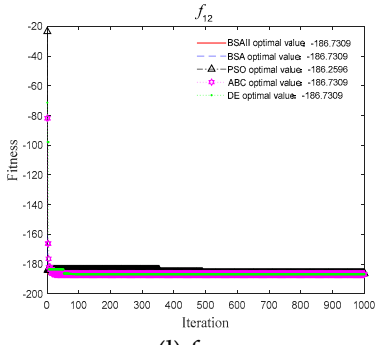
(i) f_9



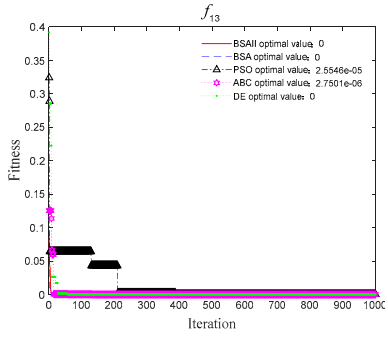
(j) f_{10}

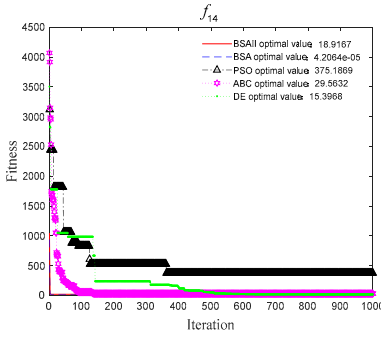


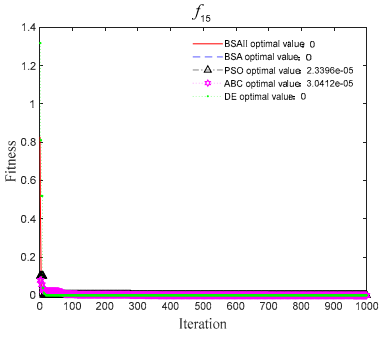
(k) f_{11}



(l) f_{12}







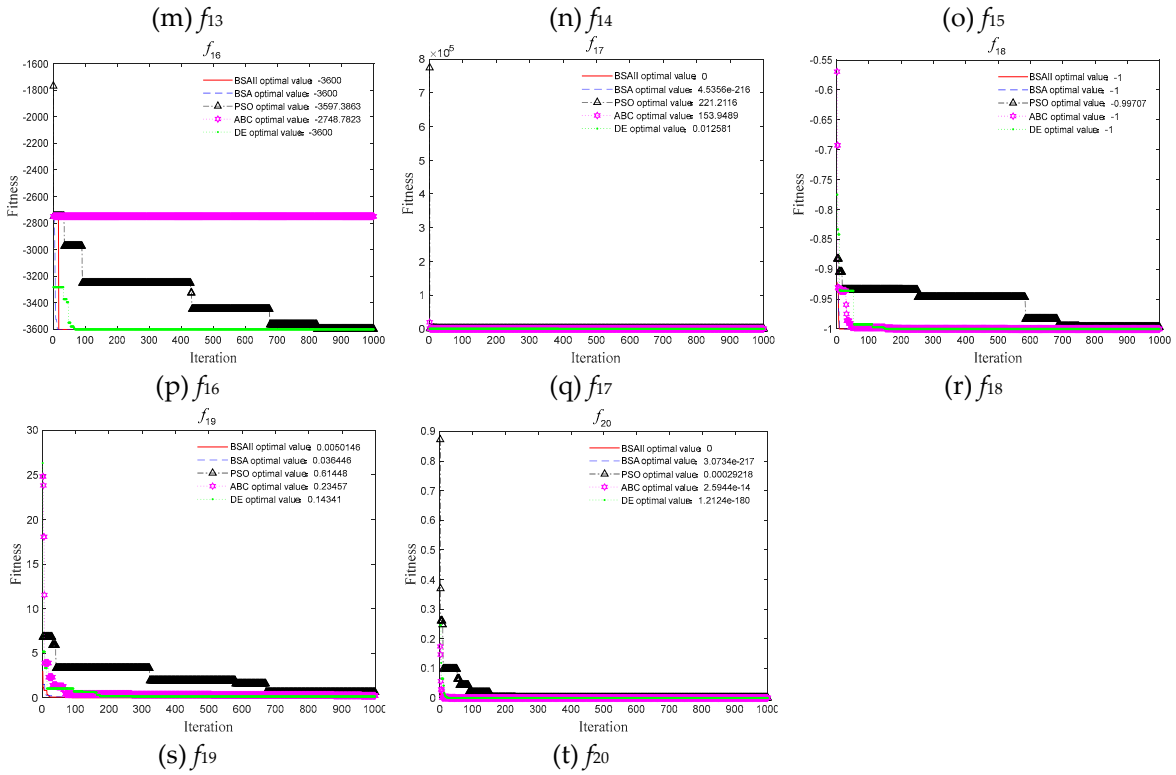


Figure 4. Convergence of benchmark functions using BSAII, BSA, particle swarm optimization (PSO), artificial bee colony (ABC) and differential evolution (DE).

Table 6. Statistical results obtained by BSAII, BSA, PSO, ABC, and DE in 50 trails (the best results are in bold).

Id.	Algorithm	MIN	MAX	MEAN	SD
f_1	**BSAII**	**0**	**0**	**0**	**0**
	BSA	6.15×10^{-266}	9.98×10^{-176}	2.40×10^{-177}	0
	PSO	3150.775	11924.62	7944.131	1938.963
	ABC	0.1627	12.08388	3.873986	2.878341
	DE	5.88×10^{-09}	8.31×10^{-07}	9.04×10^{-08}	1.23×10^{-07}
f_2	**BSAII**	**0**	**0**	**0**	**0**
	BSA	5.43×10^{-256}	2.53×10^{-63}	5.07×10^{-65}	3.55×10^{-64}
	PSO	0.000443	0.032429	0.006521	0.00539
	ABC	1.35×10^{-08}	1.64×10^{-05}	2.37×10^{-06}	3.26×10^{-06}
	DE	2.17×10^{-32}	1.02×10^{-20}	2.05×10^{-22}	1.43×10^{-21}
f_3	**BSAII**	**0**	**0**	**0**	**0**
	BSA	1.13×10^{-242}	1.06×10^{-184}	3.27×10^{-186}	0
	PSO	69.29799	315.1082	196.9478	56.54325
	ABC	0.001651	0.036862	0.016376	0.008962
	DE	2.98×10^{-10}	6.99×10^{-09}	1.51×10^{-09}	1.27×10^{-09}
f_4	**BSAII**	**−7**	**−7**	**−7**	**2.66×10^{-15}**
	BSA	−7	−7	−7	2.29×10^{-14}
	PSO	−6.99272	−6.45886	−6.90539	0.104025

Id.	Algorithm	MIN	MAX	MEAN	SD
	ABC	−7	−6.99997	−6.99999	6.26×10^{-06}
	DE	−7	−7	−7	2.66×10^{-15}
f_5	**BSAII**	**0**	**0**	**0**	**0**
	BSA	1.32×10^{-237}	1.62×10^{-177}	3.27×10^{-179}	0
	PSO	8375.27	48805.58	30703.65	7607.746
	ABC	1.394579	48.57402	14.07298	9.398897
	DE	2.91×10^{-08}	5.71×10^{-07}	1.59×10^{-07}	1.12×10^{-07}
f_6	**BSAII**	**−1**	**−1**	**−1**	**0**
	BSA	−1	−1	−1	3.94×10^{-14}
	PSO	−0.99599	-1.18×10^{-69}	−0.29917	0.334968
	ABC	−1	-8.11×10^{-05}	−0.9555	0.197304
	DE	**−1**	**−1**	**−1**	**0**
f_7	BSAII	**0**	**0**	**0**	**0**
	BSA	3.79×10^{-251}	2.53×10^{-179}	5.07×10^{-181}	0
	PSO	1.75×10^{-05}	0.002047	0.000348	0.000447
	ABC	2.20×10^{-07}	0.000157	3.22×10^{-05}	3.99×10^{-05}
	DE	1.42×10^{-178}	2.10×10^{-170}	6.36×10^{-172}	0
f_8	**BSAII**	**0**	**0**	**0**	**0**
	BSA	0	1.36×10^{-14}	2.72×10^{-16}	1.90×10^{-15}
	PSO	0.000117	0.054104	0.011096	0.012605
	ABC	2.91×10^{-08}	0.000174	3.09×10^{-05}	3.93×10^{-05}
	DE	**0**	**0**	**0**	**0**
f_9	**BSAII**	**0**	**0**	**0**	**0**
	BSA	**0**	**0**	**0**	**0**
	PSO	36.51811	113.051	77.5095	17.99438
	ABC	0.669883	1.17361	1.027435	0.086522
	DE	5.34×10^{-08}	0.027025	0.003197	0.006494
f_{10}	**BSAII**	**0**	**0**	**0**	**0**
	BSA	**0**	**0**	**0**	**0**
	PSO	118.5101	174.2391	150.3891	13.19794
	ABC	16.57457	41.5169	28.29116	5.126784
	DE	83.19845	134.5668	110.4572	12.10043
f_{11}	BSAII	2.55×10^{-05}	118.4384	40.26906	56.10528
	BSA	2.55×10^{-05}	118.4384	30.79399	51.95111
	PSO	0.004037	21.11573	2.520817	4.011557
	ABC	2.55×10^{-05}	6.29×10^{-05}	2.97×10^{-05}	6.14×10^{-06}
	DE	$\mathbf{2.55 \times 10^{-05}}$	$\mathbf{2.55 \times 10^{-05}}$	$\mathbf{2.55 \times 10^{-05}}$	**0**
f_{12}	**BSAII**	**−186.731**	**−186.731**	**−186.731**	$\mathbf{9.59 \times 10^{-14}}$
	BSA	−186.731	−186.731	−186.731	9.80×10^{-07}

Id.	Algorithm	MIN	MAX	MEAN	SD
	PSO	−186.717	−184.601	−186.243	0.436279
	ABC	−186.731	−186.731	−186.731	3.28×10^{-06}
	DE	−186.731	−186.731	−186.731	9.99×10^{-14}
	BSAII	**0**	**0**	**0**	**0**
	BSA	**0**	**0**	**0**	**0**
f_{13}	PSO	2.70×10^{-06}	0.017705	0.005071	0.004228
	ABC	8.24×10^{-10}	1.85×10^{-05}	2.82×10^{-06}	3.96×10^{-06}
	DE	**0**	**0**	**0**	**0**
	BSAII	18.83609	18.97269	18.90242	**0.029108**
	BSA	**9.70×10^{-13}**	18.80166	**1.304165**	4.264392
f_{14}	PSO	151.1679	709.0956	455.849	117.039
	ABC	19.55623	51.37853	28.38958	7.592035
	DE	12.17635	**16.29873**	14.56445	0.859034
	BSAII	**0**	**0**	**0**	**0**
	BSA	**0**	**0**	**0**	**0**
f_{15}	PSO	2.86×10^{-05}	0.012789	0.001782	0.002403
	ABC	2.63×10^{-08}	0.000207	5.00×10^{-05}	5.51×10^{-05}
	DE	**0**	**0**	**0**	**0**
	BSAII	**−3600**	−2748.78	-3565.95	166.8039
	BSA	**−3600**	**−3600**	**−3600**	**0**
f_{16}	PSO	−3599.74	−3024.95	−3504.13	103.0209
	ABC	−3598.49	−2748.78	−3307.88	297.6948
	DE	−3600	−2748.78	−2919.03	340.4871
	BSAII	**0**	**0**	**0**	**0**
	BSA	1.21×10^{-240}	3.65×10^{-54}	7.34×10^{-56}	5.11×10^{-55}
f_{17}	PSO	60.66649	438.8272	203.5698	81.22512
	ABC	101.2244	178.586	140.5003	16.80537
	DE	0.003087	0.190539	0.042672	0.037769
	BSAII	**−1**	**−1**	**−1**	**0**
	BSA	**−1**	**−1**	**−1**	**0**
f_{18}	PSO	−0.99988	−0.93622	−0.97521	0.020419
	ABC	−1	−0.99996	−0.99999	7.72×10^{-06}
	DE	**−1**	**−1**	**−1**	**0**
	BSAII	**7.97×10^{-05}**	0.14995	0.066484	0.041502
	BSA	0.002607	0.147479	0.070706	0.039185
f_{19}	PSO	0.169037	1.416421	0.729224	0.272299
	ABC	0.041325	0.298631	0.166209	0.043545
	DE	0.00036	**0.126775**	**0.063711**	**0.038273**
f_{20}	**BSAII**	**0**	**0**	**0**	**0**

Id.	Algorithm	MIN	MAX	MEAN	SD
	BSA	8.88×10^{-249}	6.76×10^{-170}	1.35×10^{-171}	0
	PSO	1.16×10^{-05}	0.003825	0.000797	0.000874
	ABC	1.00×10^{-16}	3.08×10^{-11}	1.96×10^{-12}	5.64×10^{-12}
	DE	2.31×10^{-187}	1.66×10^{-178}	3.68×10^{-180}	0

5. Optimization of On-Off Control Mode

This paper selects NEDC condition for research. The NEDC working condition test consists of two parts: the urban operation cycles and the suburban operation cycle (Figure 5). The urban operation cycle consisted of four urban operation cycle units. The test time of each cycle unit was 195 s, including idling, starting, accelerating and decelerating parking stages. The driving distance of the whole NEDC was 10.93 km and the testing time was 1184 s. The maximum velocity was 120 km/h, and average velocity was 33 km/h.

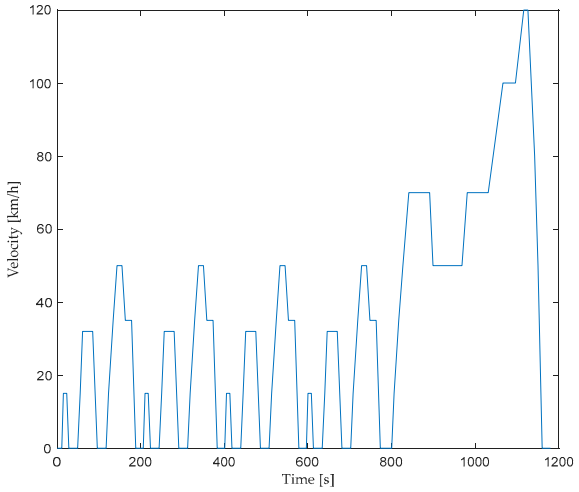


Figure 5. New European driving cycle (NEDC).

In this paper, two instances with different target distances for E-REV are analyzed. One is short-distance driving, in which the battery can complete the trip with one charge-discharge cycle. Another case is long-distance driving, and the battery needs multiple charge-discharge processes.

5.1. Traditional On-Off Control Mode

5.1.1. Distance = 100 km

The control strategy was to start the engine and recharge the battery when the power reduced to 20%, and shut down the engine when the battery was charged to 80%. According to the established model, the driving distance of the vehicle at engine start-up was 58 km. The driving distance of the vehicle was 80km when engine was shut down (Table 7). The engine was working in a full charge–discharge cycle, the uptime of the engine in E-REV was 2383 s. Figure 6 shows the variation of SOC. It can be observed from the figure, that when the battery dropped to 52% after charging, the vehicle's distance reached 100 km.

Table 7. On-off control mode.

Mode	Time [%]	Distance [km]	Working time [s]
t_{on}	20	58	2383 s
t_{off}	80	80	

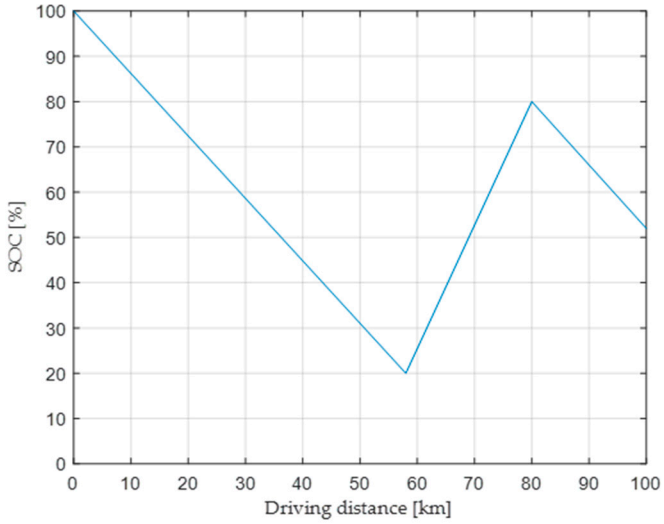


Figure 6. Battery volume of the electric vehicle, distance = 100 km.

5.1.2. Distance = 200 km

The distance exceeded 124 km, so more than one charge and discharge cycles would be repeated in the process of driving motion. According to Equation (20), there were three incomplete charge-discharge cycles. Working time of engine during vehicle in motion was 5959 s (Table 8). The SOC curve is shown as Figure 7. From the figure, it is evident that vehicle completed the target distance in the charging mode, while the battery SOC reached 50%.

Table 8. On-off control mode in the last cycle.

Mode	Time [%]	Distance [km]	Working time [s]
t_{on}	20	189	5959 s
t_{off}	50	200	

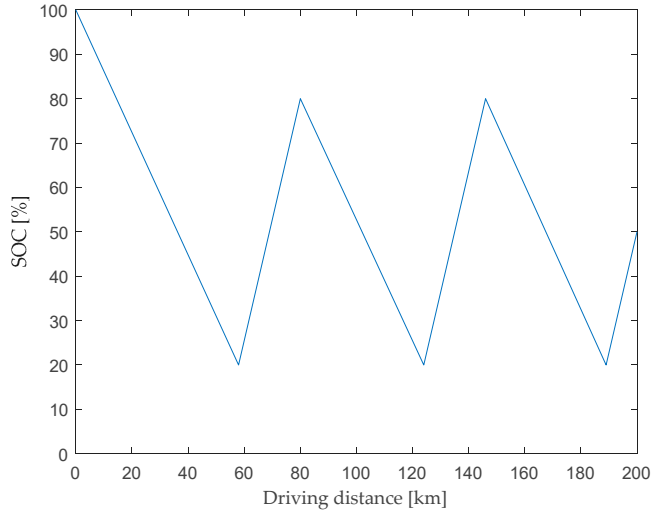


Figure 7. Battery volume of the electric vehicle, distance = 200 km.

5.2. On-Off Control Mode After Optimization

5.2.1. Distance = 100 km

BSAII was adopted to minimize the working time of engine so as to reduce fuel consumption and gas pollution. The trip distance was set to 100 km. The optimization problem and the constraints satisfied Equations (14)–(17). The number of iterations was set to 1000. Start-up and shut-down time of engine can be obtained by simulation, which are shown in Table 9. According to Table 9, it was obviously found that the early shutdown of the engine could help reduce fuel consumption and exhaust emissions of engine. The trade-off relation between the working time t with iterations is shown in Figure 8. The variation of SOC in the battery can be observed in Figure 9. The working time reduced to 36.4%, compared with the traditional control strategy.

Table 9. Optimal on-off control mode.

Mode	Time [%]	Distance [km]	Working time [s]
t_{on}	20	58	1515 s
t_{off}	58	80	

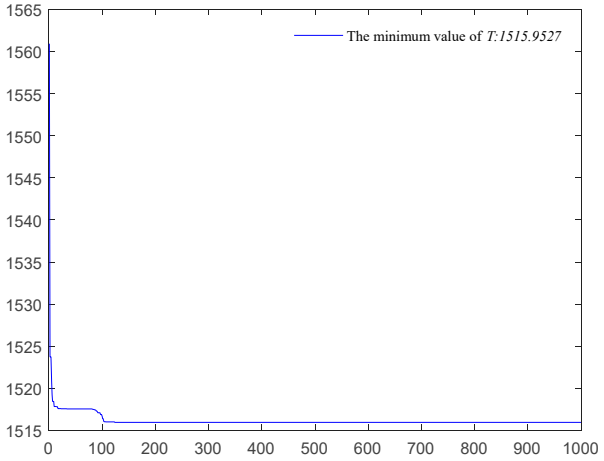


Figure 8. Convergence curve of engine uptime optimized via BSAII.

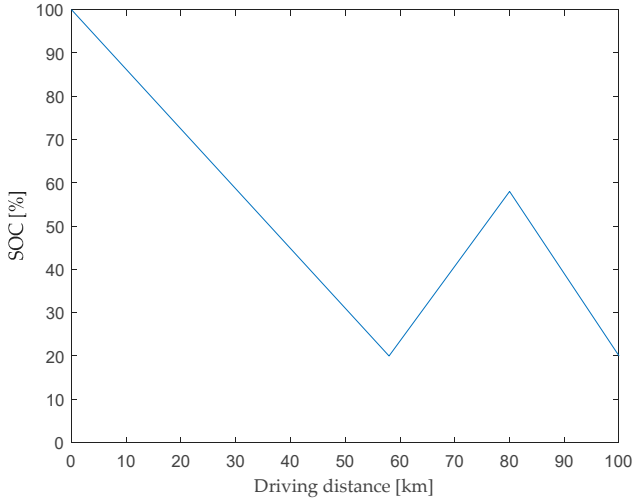


Figure 9. SOC of battery in electric vehicle after optimization, distance = 100 km.

5.2.1. Distance = 200 km

When the target distance was set to 200 km, it meant that the engine needed to be turned on and off repeatedly. The working hours of engine was calculated using Equation (19), and the distance constraint was formulized as Equation (20). The optimal on-off mode found by BSAII is illustrated in Table 10. According to the results, there were three charge–discharge cycles during driving process.

In the last cycle, the engine turned off when SOC in battery reached 30%, and the electric power which was just enough to complete the entire distance (200 km). The convergence curve in the optimization problem is shown as Figure 10. After optimization, the total running time of engine was 5168 s, a 13% reduction in contrast to 5959 s under the traditional control strategy. The SOC of the battery power for electric vehicles while the car was in motion is shown in Figure 11.

Table 10. Optimal on-off control mode in the last cycle.

Mode	Time [%]	Distance [km]	Working time [s]
t_{on}	20	189	5168 s
t_{off}	30	193	

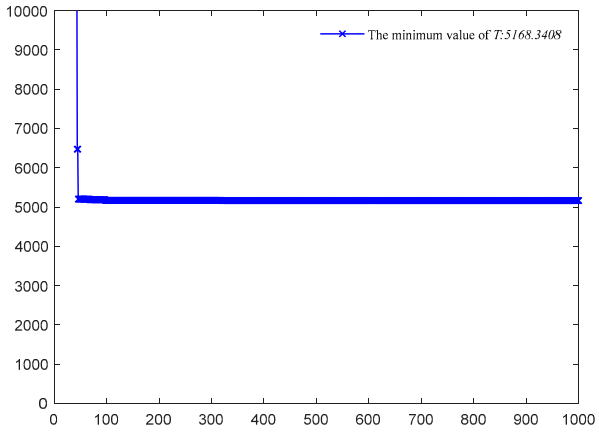


Figure 10. Convergence curve of t optimized with BSAII.

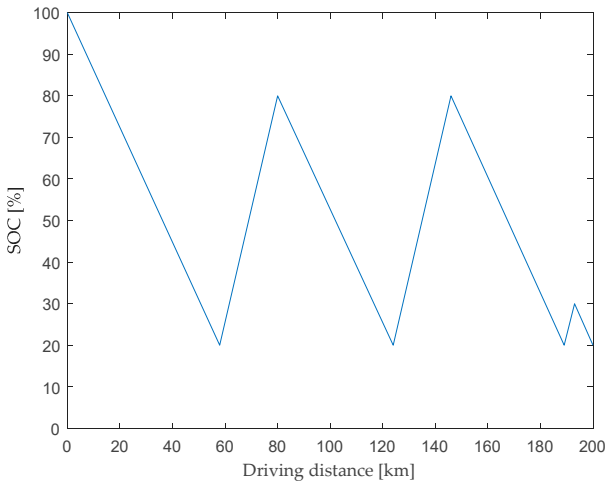


Figure 11. SOC of battery in electric vehicle after optimization, distance = 200 km.

6. Conclusions

This paper proposed another version of BSA, named as BSAII. In the version of BSAII algorithm, the spatial coordinates of birds in solution-space instead of the fitness function were used to determine the distance from the center of the whole bird group. Based on this method, the coefficients A_1 and A_2 were more accurate than that in BSA. We examined the performance of two different representations of defense behavior for BSA algorithms, and compared their experimental results with other bio-inspired algorithms, including PSO, ABC, and DE. It was evident from the statistical and graphical results highlighted that the BSAII outperformed other algorithms on most of the instances, in terms of convergence rate and accuracy of the optimal solution. Besides this, it was the

first time that BSAII has been applied to the energy management of electric vehicles, which helps to reduce engine fuel consumption and exhaust emissions. Based on the analysis in the previous section, it is clear that:

- The BSAII performed statistically superior to or equal to the BSA on 16 benchmark problems. On these problems, it obtained the real optimal solution. However, in the case of Rosenbrock function (f_{14}), it was prone to falling into local optimum.
- The energy management of electric vehicles in this paper referred to minimization uptime of RE. The uptime was determined by the time interval between engine on and off. Two instances with different target driving distances were optimized with BSAII. Results indicated that the uptime of engine could be reduced by 36.4% with a target distance of 100 km, and 13% with a target distance of 200 km, respectively, in the NEDC condition. Hence we can draw a conclusion that based on the optimization strategy of BSAII, the on/off timing of the engine can be accurately controlled, which can effectively shorten the uptime of the engine, reduce fuel consumption and exhaust emissions, and also facilitate the next external charging.

It is hoped that in future work, more approaches will be designed to avoid the problem of local optimum, without sacrificing the convergence rate in BSA.

Author Contributions: **C**onceptualization, D.W. and L.F.; methodology, D.W.; software, D.W. and L.F.; validation, D.W.; formal analysis, L.F.; investigation, L.F.; writing—original draft, D.W.; writing—review and editing, L.F.; project administration, D.W.

Funding: This research was funded by Natural Science Foundation of Jiangsu Province, grant number BK20130873.

Conflicts of Interest: The authors of the manuscript declare no conflicts of interest with any of the commercial identities mentioned in the manuscript.

References

1. Amp, N.J.; Su, Z. On/off timing optimization for the range-extender in extended-range electric vehicles. *Automot. Eng.* **2013**, *35*, 418–423.
2. Zhao, J.; Han, B.; Bei, S. Start-stop moment optimization of range extender and control strategy design for extended-range electric vehicle. In Proceedings of the 5th Asia Conference on Materials Science and Engineering, Tokyo, Japan, 9–11 June 2017; doi:10.1088/1757-899X/241/1/012025.
3. Jiang, Y.C.; Zhou, H.; Cheng, L.; Wang, L. A study on energy management control strategy for an extended-range electric vehicle. In Proceedings of the 2015 International Conference on Energy and Mechanical Engineering, Wuhan, China, 17–18 October 2015; pp. 159–169.
4. Pan, C.F.; Han, F.Q.; Chen, L.; Xu, X.; Chen, L. Power distribution strategy of extend range electric vehicle based on fuzzy control. *J. ChongQing Jiaotong Univ. Nat. Sci.* **2017**, *33*, 140–144.
5. Niu, J.G.; Pei, F.L.; Zhou, S. Zhang, T. Multi-objective optimization study of energy management strategy for extended-range electric vehicle. *Adv. Mater. Res.* **2013**, *694–697*, 2704–2709.
6. Abdelgadir, A.A.; Alsawalhi, J.Y. Energy management optimization for an extended range electric vehicle. In Proceedings of the IEEE International Conference on Modeling, Sharjah, UAE, 4–6 April 2017.
7. Xi, L.H.; Zhang, X.; Geng, C.; Xue, Q.C. Energy management strategy optimization of extended-range electric vehicle based on dynamic programming. *J. Traffic Transp. Eng.* **2018**, *18*, 148–156.
8. Vatanparvar, K.; Faezi, S.; Burago, L.; Levorato, M.; Faruque, M.A.A. Extended range electric vehicle with driving behavior estimation in energy management. *IEEE Trans. Smart Grid* **2019**, *10*, 2959–2968.
9. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **2001**, *43*, 303–315.
10. García-Monzó, A.; Migallón, H.; Jimeno-Morenilla, A.; Sanchez-Romero, J.L.; Rico, H.; Rao, R.V. Efficient Subpopulation Based Parallel TLBO Optimization Algorithms. *Electronics* **2018**, *8*, 19, doi:10.3390/electronics8010019.
11. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61.
12. Al-Moalmi, A., Luo, J., Salah, A., Li, K. Optimal virtual machine placement based on grey wolf optimization. *Electronics* **2019**, *8*, 283.

13. Fatima, A.; Javaid, N.; Anjum Butt, A.; Sultana, T.; Hussain, W.; Bilal, M.; ur Rehman Hashmi, M.A.; Akbar, M.; Llahi, M. An enhanced multi-objective gray wolf optimization for virtual machine placement in cloud data centers. *Electronics* **2019**, *8*, 218, doi:10.3390/electronics8020218.
14. Goel, S. Pigeon optimization algorithm: A novel approach for solving optimization problems. In Proceedings of the International Conference on Data Mining & Intelligent Computing, New Delhi, India, 5–6 September 2014.
15. Duan, H.; Qiao, P.X. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Intern. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37.
16. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67.
17. Meng, X.B. A new bio-inspired optimisation algorithm: Bird swarm algorithm. *J. Exp. Theor. Artif. Intell.* **2015**, *28*, 673–687.
18. Yang, W.R.; Ma, X.Y.; Xu, M.L.; Bian, X.L. Research on scheduling optimization of grid-connected micro-grid based on improved bird swarm algorithm. *Adv. Technol. Electr. Eng. Energy* **2018**, *2*, 53–60.
19. Yang, W.Y.; Ma, X.Y.; Bian, X.L. Optimization of Micro-grid operation using improved bird swarm algorithm under time-of-use price mechanism. *Renew. Energy Resour.* **2018**, *7*, 1046–1054.
20. Yang, W.Y.; Ma, X.Y.; Bian, X.L. Adaptive improved bird swarm algorithm based on levy flight strategy. *J. Hebei Univ. Technol.* **2017**, *5*, 10–16, 22.
21. Zhang, L.; Bao, Q.; Fan, W.; Cui, K.; Xu, H.; Du, Y. An improved particle filter based on bird swarm algorithm. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017.
22. Wang, Y.; Wan, Z.; Peng, Z. A novel improved bird swarm algorithm for solving bound constrained optimization problems. *Wuhan Univ. J. Nat. Sci.* **2019**, *24*, 349–359.
23. Zahara, E.; Hu, C.H. Solving constrained optimization problems with hybrid particle swarm optimization. *Eng. Optim.* **2008**, *40*, 1031–1049.
24. Wang, Z.X.; Bei, S.Y.; Wang, W. The optimization control of on and off timing for the engine in extended-range electric vehicle. *Mach. Des. Manuf.* **2017**, *4*, 100–103.
25. Yang, X.S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *24*, 169–174.
26. Virtual Library of Simulation Experiments: Test Functions and Datasets. Available online: http://www.sfu.ca/~ssurjano (accessed on 30 July 2019).
27. Eberhart, R.C.; Shi, Y.H. Particle swarm optimization: Development, application, and resourcesIn Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea, 27–30 May 2001, doi:10.1109/CEC.2001.934374.
28. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359, doi:10.1023/A:1008202821328.
29. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471.