IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 6, 2018, accepted September 11, 2018, date of publication September 17, 2018, date of current version October 17, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2870531

A Switch-Mode Firefly Algorithm for Global Optimization

JIAN HUANG, (Senior Member, IEEE), XIAOCHAO CHEN, (Student Member, IEEE), AND DONGRUI WU, (Senior Member, IEEE)

Key Laboratory of the Ministry of Education for Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Dongrui Wu (drwu@hust.edu.cn)

The work of J. Huang was supported in part by the National Natural Science Foundation of China under Grant 61473130, in part by the Special Program for Technology Innovation of Hubei Province under Grant 2016AAA039, in part by the Applied Basic Research Program of Wuhan under Grant 2016010101010014, and in part by the Open Fund Project of the MOE Key Laboratory of Image Processing and Intelligence Control under Grant IPIC2017-06.

ABSTRACT The firefly algorithm has been successfully used in many optimization problems. However, the standard firefly algorithm uses a fixed randomization parameter in the optimization, which emphasizes more on exploration than exploitation, and hence impacts its convergence. This paper proposes a switch-mode firefly algorithm, which first focuses on exploration and then switches to exploitation. A fixed randomization parameter is used in exploration, and a gradually decreasing random randomization parameter is used in exploitation. The condition for the switching from exploration to exploitation is identified automatically. Extensive experiments on 15 benchmark functions were performed to verify the effectiveness of the proposed approach.

INDEX TERMS Firefly algorithm, global optimization, switch-mode.

I. INTRODUCTION

Many real-world optimization problems are nonlinear, nonconvex, and high-dimensional, and the derivatives of the objective function may not be computed easily [1], [2]. Thus, traditional optimization techniques, including exhaustive grid search, mathematical programming, and backpropogation, may not be readily applicable. Nature-inspired metaheuristic algorithms [3]–[5], including genetic algorithms [6], [7], particle swarm optimization [8]–[10], ant colony optimization [11], artificial bee [12], etc, have been proposed for such problems and achieved great success in practice.

Generally, a metaheuristic algorithm considers the following two objectives in its operation [13], [14]:

1) *Exploration* (diversification), which probes the entire search space with the hope of finding all promising regions. Its goal is to diversify the search in order to avoid getting trapped in a local optimum.
2) *Exploitation* (intensification), which probes a local region of the search space with the hope of improving a promising solution that has already been identified through exploration.

A good metaheuristic approach needs to have a well-balanced tradeoff between exploration and exploitation.

By mimicking the communication behavior of fireflies, in 2009 Yang [15] introduced a novel metaheuristic approach called firefly algorithm (FA), which can find the global and local optima of the objective function simultaneously [1]. So far, FA has found successful applications in a variety of problem domains, including economic load dispatch [16], capacitated facility location [17], image compression [18], dynamic economic dispatch [19], knapsack problems [20], mobile robot path planning [21], RFID network planning [22], hyperspectral image classification [23], etc.

However, the standard firefly algorithm (SFA) used a fixed randomization parameter in optimization, which emphasizes more on exploration than exploitation [24]. Thus, its convergence speed is slow, and can be easily trapped into local optima. Multiple improvements to the SFA have been proposed in the literature, e.g., FA using simulated annealing step strategy (SAFA) [14], FA using chaotic sequence to serve as the absorption coefficient (CFA) [16], wise-step FA which taking the fireflies' personal and global best positions into consideration (WSSFA) [25], FA with variable strategy step size (VSSFA) [26], FA with neighborhood attraction (NAFA) [27], hybrid multi-objective FA (HMOFA) using a random attractiveness β_0 [28], etc.

Algorithm 1 The Standard Firefly Algorithm (SFA) [15]

Input: $f(\mathbf{x})$, the object function to be maximized;
n, the size of the firefly population;
γ, the light absorption coefficient;
α, the randomization parameter;
K, the maximum number of iterations.
Output: $\mathbf{x}^*$, which maximizes $f(\mathbf{x})$.
Generate the initial firefly population $\{\mathbf{x}_i\}_{i=1,\ldots,n}$;
Compute $f(\mathbf{x}_i)$, $i = 1, \ldots, n$;
Set $k = 1$;
while $k \leq K$ **do**
  for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, n$ **do**
      if $f(\mathbf{x}_i) < f(\mathbf{x}_j)$ **then**
        Move Firefly i to Firefly j according to (1);
        Compute the updated $f(\mathbf{x}_i)$;
      end
    end
  end
  $k = k + 1$;
end
Return $\mathbf{x}^* = \arg\max_{i=1,\ldots,n} f(\mathbf{x}_i)$.

Although there are a lot of variants of FA, most of them have not fully considered the different characteristics of different evolutionary stages. Intuitively, exploration should play a more important role at the early stage of FA to identify promising search regions. As the evolutionary process goes on, however, exploitation should be emphasized to fine tune the solutions. In consideration of these characteristics, this paper proposes a switch-mode FA (SMFA), which first focuses on exploration, and then switches to exploitation. A fixed randomization parameter is used in exploration, and a gradually decreasing random randomization parameter is used in exploitation. The condition for the switching from exploration to exploitation is identified automatically. Extensive experiments on 15 benchmark functions were performed to verify the effectiveness of SMFA.

The remainder of this paper is organized as follows: Section II introduces the SFA. Section III proposes the SMFA. Section IV compares the performance of SMFA with another six FA variants on 15 benchmark functions. Finally, Section V draws conclusions.

II. THE STANDARD FIREFLY ALGORITHM (SFA)

Fireflies use light for communication. The SFA was inspired by this phenomena, and it uses the following three simplified assumptions [15]:

1) All fireflies are unisex and are attracted to each other.
2) The attractiveness is proportional to the brightness of a firefly, and is inversely proportional to the distance. The brightest firefly moves randomly since there is no other brighter fireflies to attract it.
3) The brightness of a firefly is determined by the fitness of the objective function.

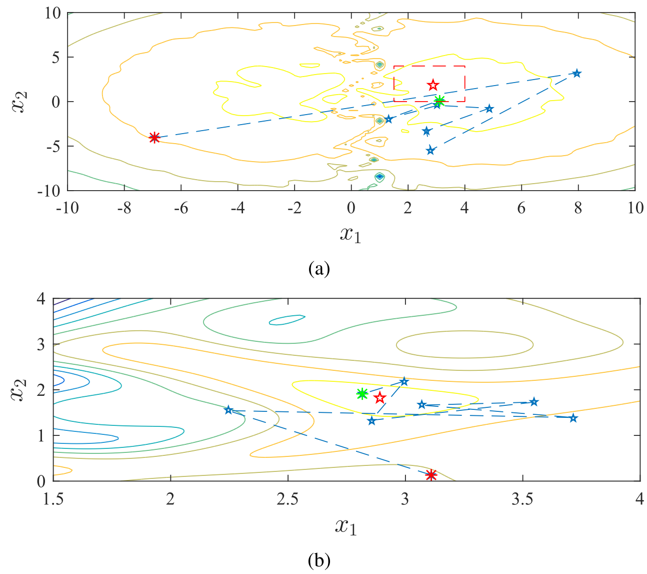


FIGURE 1. Search trajectory of a firefly in the SMFA. (a) Exploration; (b) Exploitation. The region in (b) is the red framed region in (a).

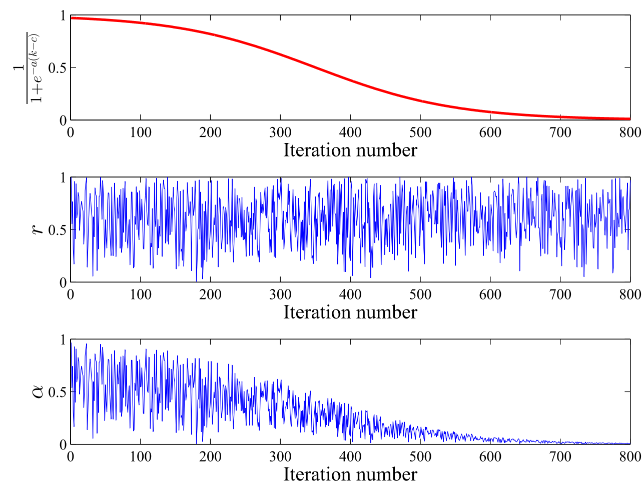


FIGURE 2. The gradually decreasing random α. $\alpha_0 = 1$ was used.

Mathematically, the movement of Firefly i, attracted by Firefly j, is computed as [15]:

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2}(\mathbf{x}_j - \mathbf{x}_i) + \alpha\boldsymbol{\epsilon} \tag{1}$$

where r_{ij} is the Euclidean distance between the two fireflies, β_0 is the attractiveness at $r = 0$, γ is a fixed light absorption coefficient, α is a randomization parameter, and $\boldsymbol{\epsilon}$ is a random vector usually uniformly distributed in $[-0.5, 0.5]$.

The pseudo-code of the SFA is shown in Algorithm 1.

III. THE SWITCH-MODE FIREFLY ALGORITHM (SMFA)

This section introduces our proposed SMFA, whose pseudo-code is given in Algorithm 2.

Algorithm 2 The Switch-Mode Firefly Algorithm (SMFA)

Input: $f(\mathbf{x})$, the object function to be maximized;
n, the size of the firefly population;
γ, the light absorption coefficient;
α_0, the randomization parameter;
K, the maximum number of iterations;
K_0, the maximum number of exploration iterations;
η, the threshold for the change rate.
Output: $\mathbf{x}^*$, which maximizes $f(\mathbf{x})$.
Generate the initial firefly population $\{\mathbf{x}_i\}_{i=1,\ldots,n}$;
Compute $f(\mathbf{x}_i)$, $i = 1, \ldots, n$;
Set $k = 1$;
while $k \leq K$ **do**
    for $i = 1, \ldots, n$ **do**
        for $j = 1, \ldots, n$ **do**
            if $f(\mathbf{x}_i) < f(\mathbf{x}_j)$ **then**
                if $\dot{f} > \eta$ **then**
                    $\alpha = \alpha_0$;
                else
                    Compute α by (3);
                end
                Move Firefly i to Firefly j according to (1);
                Compute the updated $f(\mathbf{x}_i)$;
            end
        end
    end
    $k = k + 1$;
end
Return $\mathbf{x}^* = \arg\max_{i=1,\ldots,n} f(\mathbf{x}_i)$.

A. MODE SWITCH

As mentioned in the Introduction, all metaheuristic algorithms need to consider the trade-off between exploration and exploitation. Generally, at the beginning of the optimization, exploration is more important, as we want to quickly and coarsely explore the entire search space to identify the most promising regions. After that, exploitation is used to examine the promising regions more carefully to fine tune the solution.

Our proposed SMFA starts with exploration, using the SFA with a fixed randomization parameter α. Then, it switches to exploitation, using an FA with a gradually decreasing random α (Section III-B). Fig. 1 illustrates this concept using a real search trajectory generated by the SMFA, where $\mathbf{x} = [x_1, x_2]$, and the "☆" represents the global optimum of the objective function. The firefly starts at the "*" position in Fig. 1(a). After several exploration iterations,

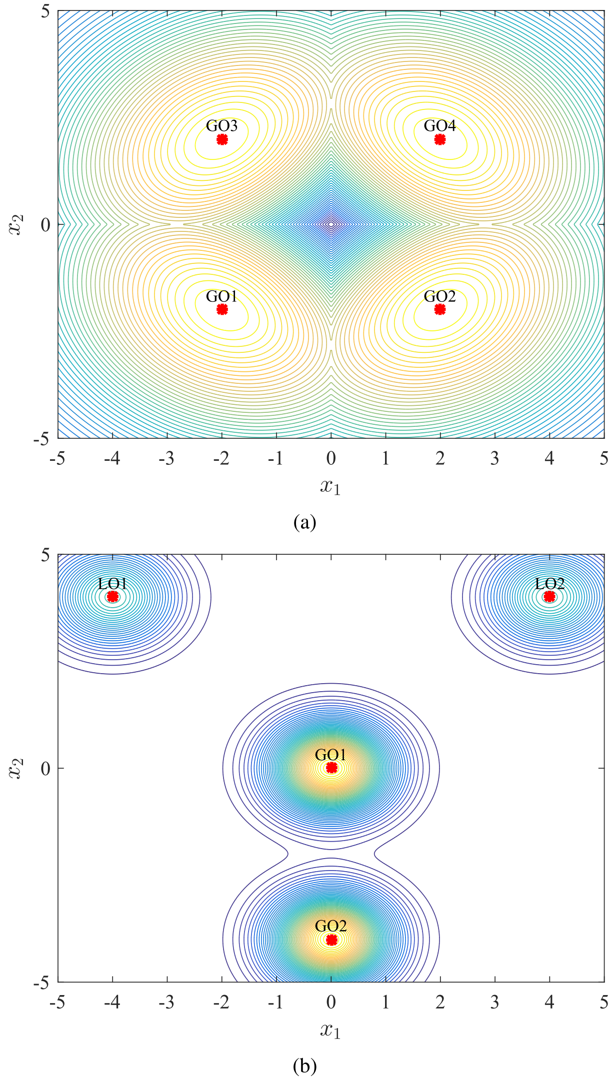


FIGURE 3. (a) Contour and the four global optima of f_1; (b) contour and the two global optima and two local optima of f_2.

it reaches the "*" position in Fig. 1(a). Then the SMFA enters the exploitation mode. The initial position of the firefly in exploitation is represented by the "*" in Fig. 1(b), which is the same as the "*" in Fig. 1(a) (note that the entire region in Fig. 1(b) is corresponding to the red framed region in Fig. 1(a); we enlarge that region to better visualize the search trajectory in exploitation). After a few exploitation iterations, the firefly reaches the "*" position in Fig. 1(b), which is very close to the global optimum.

It is very important to determine when we should explicitly switch from exploration to exploitation. Let $f_k = \max_{i=1,\ldots,n} f(\mathbf{x}_i)$ be the best fitness in the kth iteration. We then compute $\dot{f}$, the change rate of f_k, as:

$$\dot{f} = \frac{f_k - f_{k-1}}{f_{k-1} - f_{k-2}} \tag{2}$$

It is difficult to determine when the exact and best switching time is. Thus, we use a simple heuristics, i.e., $\dot{f} < \eta$,

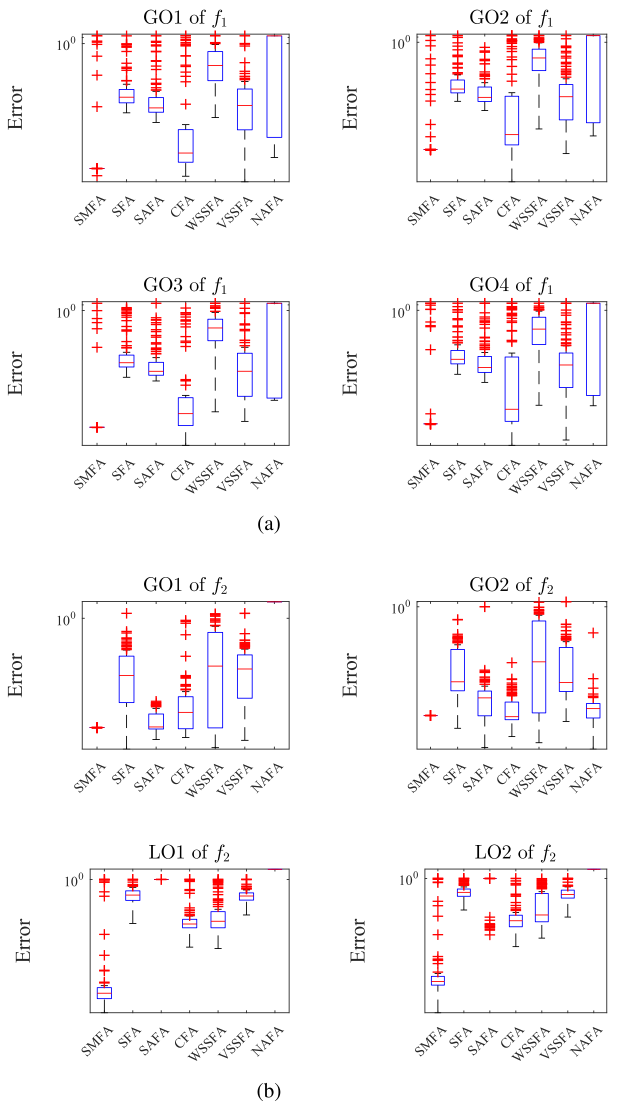


FIGURE 4. Errors in 100 runs of the six algorithms. (a) The four global optima for f_1; (b) The two global optima and two local optima for f_2.

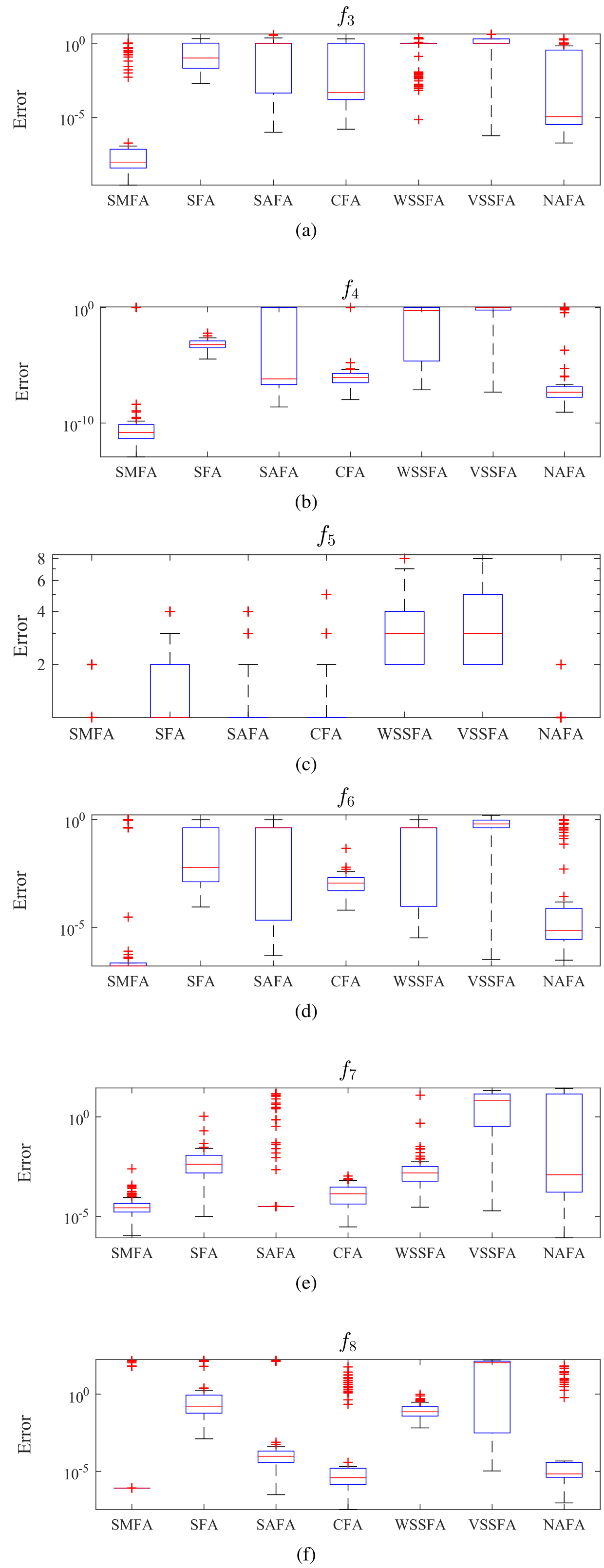


FIGURE 5. Errors in 100 runs of the six algorithms. (a) f_3; (b) f_4; (c) f_5; (d) f_6; (e) f_7; (f) f_8.

to switch the search mode automatically. In words, it switches to the exploitation mode when the exploration mode has not improved the best fitness value significantly for a certain period of time. A similar idea can be found in switching PSO algorithms (e.g. [29]) while they divided the search process into more modes and used more complex switching conditions.

B. EXPLOITATION WITH GRADUALLY DECREASING RANDOMIZATION PARAMETER

Randomization plays an important role in the SFA, because it increases the diversity of the population. During exploration, usually the randomization parameter α is set to a relative large value, which enables the SFA to explore the search space more quickly. However, a large α also results in slow convergence in exploitation. To cope with this problem, the exploitation mode of the SWFA uses a gradually decreasing random α:

$$\alpha = \frac{r\alpha_0}{1 + e^{-a(k-c)}} \tag{3}$$

where α_0 is the randomization parameter in exploration, r is a random number from a truncated normal distribution [30], [31] $N(0.5, 1)$ in (0, 1), a and c are parameters of the sigmoid denominator, and k is the current iteration number.

Examples of $\frac{1}{1+e^{-a(k-c)}}$, r and α are shown in Fig. 2. Observe that α is a gradually decreasing random number, which enables the SWFA to focus more and more on exploitation as k increases. First, the proposed adjustment strategy maintains a gradually decreasing trend in α which is crucial to the convergence of the firefly population. Second, the random α brings a certain degree of diversity which helps explore the search space more sufficiently and avoid local optima as much as possible. The product of a random number r (from a truncated normal distribution) and a monotonically decreasing function $\frac{\alpha_0}{1+e^{-a(k-c)}}$ is a random function (see the third subfigure in Fig. 2), whose mean equals half of the corresponding value of the original monotonically decreasing function and standard deviation gradually decreases with time. Experiments showed that r greatly improves the FA algorithm performance.

IV. EXPERIMENTS

In this section, the performance of SWFA is compared with another six approaches in the literature:

- The standard firefly algorithm (SFA) [14].
- The simulated annealing strategy firefly algorithm (SAFA) [14].
- The firefly algorithm with chaos (CFA) [32].
- The wise step strategy firefly algorithm (WSSFA) [25].
- The variable step size firefly algorithm (VSSFA) [26].
- The firefly algorithm with neighborhood attraction (NAFA) [27].

A. EXPERIMENT SETUP

Three types of benchmark functions were used: 1) multimodal functions to verify the ability of SWFA to find all global and local optima simultaneously; 2) low-dimensional maximization problems; and, 3) high-dimensional minimization problems. Table 1 summarizes the 15 benchmark functions.

We used $\gamma = 1$ and $\beta_0 = 1$ for all algorithms, and the population size $n = 35$ for f_1 and f_2, and $n = 20$ for all other functions. To validate that SMFA converges faster, or, in other words, SMFA can find a better solution when given a fixed amount of search time, each algorithm was run only T seconds, where $T = 0.03$ for f_1 and f_2, $T = 0.05$ for f_3 to f_8, and $T = 3$ for f_9 to f_{15}. Additional parameters of SFA and SAFA can be found in [14], WSSFA in [25], CFA in [32], and VSSFA in [26]. For SMFA, we chose $\eta = 0.1$. For (3), we set $a = -0.1$, $c = 15$, and α_0 was chosen according to the range of the corresponding search space (1 for $f_1 - f_3, f_{13}$ and f_{14}, 2 for $f_4 - f_7$ and f_{10}, 4 for f_8, 10 for f_9, 20 for f_{11} and f_{12}, and 0.2 for f_{15}).

We ran each algorithm for each benchmark function 100 times, and used boxplots to show the errors between the found optima and the true global optima. All simulations were performed using MATLAB 2016a on a Windows machine with an Intel core i3 2.00G HZ CPU and 4GB RAM.

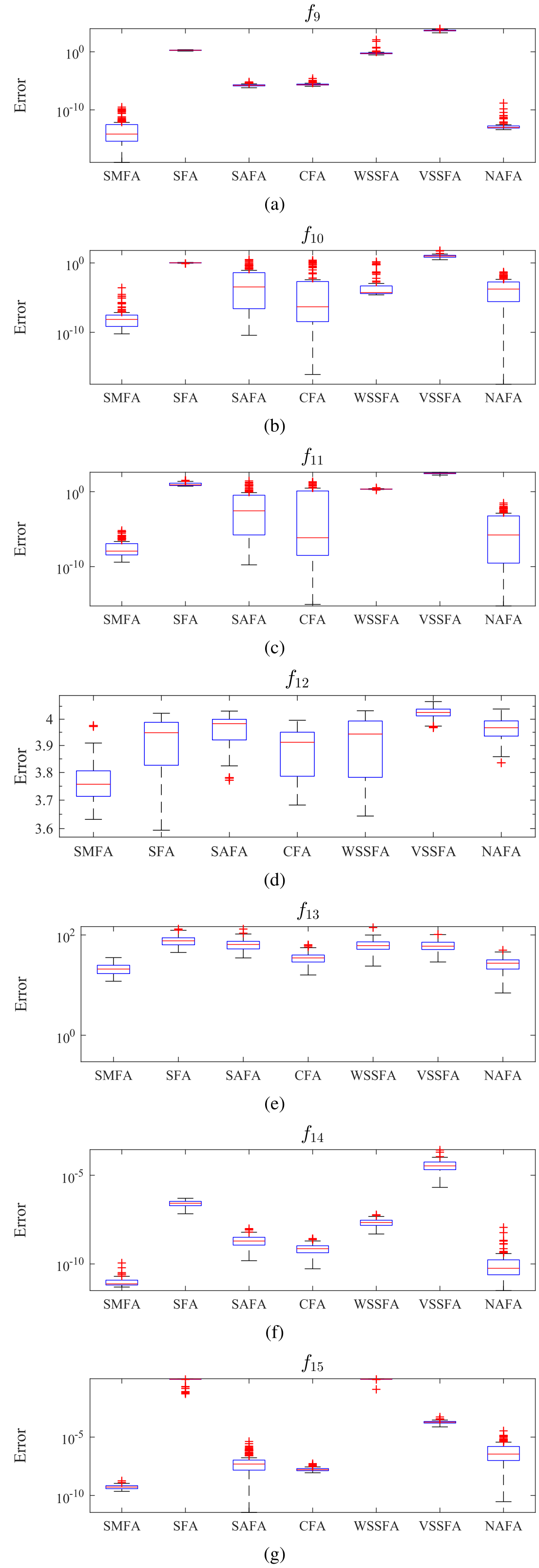


FIGURE 6. Errors in 100 runs of the six algorithms. (a) f_9; (b) f_{10}; (c) f_{11}; (d) f_{12}; (e) f_{13}; (f) f_{14}; (g) f_{15}.

TABLE 1. The 15 benchmark functions. "*" means a local optimum. $d = 30$ in $f_9 - f_{15}$.

Function	Formulation		Range	Optima
f_1	max	$(\lvert x_1\rvert + \lvert x_2\rvert)\cdot\exp[-0.0625(x_1^2+x_2^2)]$	$[-5,5]$	2.4261
f_2	max	$\exp[-(x_1-4)^2-(x_2-4)^2]+\exp[-(x_1+4)^2-(x_2-4)^2]$ $+2[\exp(-x_1^2-x_2^2)+\exp(-x_1^2-(x_2+4)^2)]$	$[-5,5]$	$2, 1^*$
f_3	max	$-[20+(x_1^2-10\cos 2\pi x_1)+(x_2^2-10\cos 2\pi x_2)]$	$[-2.048, 2.048]$	0
f_4	max	$\cos x_1\cdot\cos x_2\cdot\exp[-(x_1-\pi)^2-(x_2-\pi)^2]$	$[-20,20]$	1
f_5	max	$\sum_{i=1}^2 integer(x_i)$	$[-5.12, 5.12]$	10
f_6	max	$\ln\left[(\sin^2(\cos x_1+\cos x_2)^2-\cos^2(\sin x_1+\cos x_2)^2\right]$ $-0.1[(x_1-1)^2+(x_2-1)^2]$	$[-10,10]$	2.28395
f_7	max	$280-0.5(x_1^4-16x_1^2+5x_1+x_2^4-16x_2^2+5x_2)$	$[-5,5]$	358.3323
f_8	max	$-\left[\sum_{i=1}^5 i\cos\left((i+1)x_1+i\right)\right]\left[\sum_{i=1}^5 i\cos\left((i+1)x_2+i\right)\right]$	$[-10,10]$	186.7309
f_9	min	$\sum_{i=1}^d x_i^2$	$[-100,100]$	0
f_{10}	min	$\sum_{i=1}^d \lvert x_i\rvert+\prod_{i=1}^d x_i$	$[-10,10]$	0
f_{11}	min	$\sum_{i=1}^d \lvert x_i+0.5\rvert$	$[-10,10]$	0
f_{12}	min	$\sum_{i=1}^d -x_i\cdot\sin(\sqrt{\lvert x_i\rvert})+418.9829d$	$[-500,500]$	0
f_{13}	min	$\sum_{i=1}^d (x_i^2-10\cos 2\pi x_i+10)$	$[-5.12,5.12]$	0
f_{14}	min	$\sum_{i=1}^d \lvert x_i\rvert\exp(-\sum_{i=1}^d x_i^2)$	$[-1,1]$	0
f_{15}	min	$\sum_{i=1}^d -\exp[-0.5-\sum_{i=1}^d x_i^2]$	$[-1,1]$	-1

TABLE 2. Mean errors of different FAs on the 15 benchmark functions.

Function	SMFA	CFA	WSSFA	VSSFA	SAFA	SFA	NAFA
f_1 $GO1:$	$1.10E-01$	$2.06E-01$	$3.07E-01$	$\mathbf{8.02E-02}$	$1.04E-01$	$8.46E-02$	$1.74E-00$
f_1 $GO2:$	$4.50E-02$	$3.07E-01$	$3.43E-01$	$6.92E-02$	$\mathbf{1.67E-02}$	$7.17E-02$	$1.75E-00$
f_1 $GO3:$	$\mathbf{4.14E-02}$	$5.97E-02$	$2.93E-01$	$6.35E-02$	$4.29E-02$	$9.06E-02$	$1.72E-01$
f_1 $GO4:$	$1.44E-01$	$2.59E-01$	$2.94E-01$	$1.07E-01$	$\mathbf{7.59E-02}$	$9.07E-02$	$1.80E-00$
f_2 $GO1:$	$\mathbf{2.24E-07}$	$2.00E-02$	$1.76e-01$	$4.63E-02$	$1.19E-06$	$2.80E-02$	$1.00E+01$
f_2 $GO2:$	$\mathbf{2.25E-07}$	$6.20E-06$	$1.17E-01$	$4.50E-02$	$2.00E-02$	$5.50E-03$	$2.56E-04$
f_2 $LO1:$	$1.08E-01$	$\mathbf{5.62E-02}$	$8.61E-02$	$8.56E-02$	$1.00E+00$	$9.07E-02$	$1.00E+01$
f_2 $LO2:$	$8.39E-02$	$\mathbf{3.89E-02}$	$7.65E-02$	$1.10E-01$	$8.80E-01$	$1.40E-01$	$1.00E+01$
f_3	$\mathbf{1.09E-01}$	$4.18E-01$	$9.68E-01$	$1.20E+00$	$7.84E-01$	$4.97E-01$	$2.49E-01$
f_4	$7.00E-02$	$2.00E-02$	$5.01E-01$	$7.80E-01$	$3.55E-01$	$\mathbf{9.12E-04}$	$1.62E-01$
f_5	$\mathbf{5.00E-02}$	$6.30E-01$	$3.06E+00$	$3.45E+00$	$6.90E-01$	$1.43E+00$	$1.03E+00$
f_6	$1.49E-01$	$\mathbf{1.90E-03}$	$3.21E-01$	$6.45E-01$	$3.00E-01$	$2.37E-01$	$1.34E-01$
f_7	$\mathbf{7.27E-05}$	$2.00E-04$	$1.30E-01$	$7.30E+00$	$1.35E-00$	$1.94E-02$	$5.54E+00$
f_8	$9.85E-00$	$2.02E-00$	$\mathbf{1.28E-01}$	$7.63E+01$	$6.99E-00$	$2.81E+01$	$5.99E+00$
f_9	$\mathbf{9.75E-12}$	$3.06E-06$	$2.51E-00$	$4.80E+03$	$1.91E-06$	$1.87E+00$	$1.71E-11$
f_{10}	$\mathbf{3.51E-06}$	$1.39E-01$	$7.73E-02$	$1.05E+01$	$1.77E-01$	$1.06E+00$	$3.10E-03$
f_{11}	$\mathbf{3.60E-07}$	$1.39E-00$	$2.26E+00$	$2.83E+02$	$1.41E-00$	$1.10E+01$	$1.30E-03$
f_{12}	$\mathbf{5.85E+03}$	$7.61E+03$	$8.00E+03$	$1.06E+04$	$9.12E+03$	$8.24E+03$	$9.22E+03$
f_{13}	$\mathbf{2.14E+01}$	$3.57E+01$	$6.16E+01$	$6.13E+01$	$6.59E+01$	$7.83E+01$	$2.76E+01$
f_{14}	$\mathbf{1.14E-11}$	$8.08E-10$	$2.35E-08$	$4.36E-05$	$2.42E-09$	$2.69E-07$	$3.39E-10$
f_{15}	$\mathbf{5.62E-10}$	$1.79E-08$	$9.07E-01$	$1.98E-04$	$2.32E-07$	$8.47E-01$	$1.88E-06$

B. EXPERIMENTAL RESULTS

According to [14], FA can find all global and local optima simultaneously. Multimodal functions f_1 and f_2 were used to verify this. f_1 has four global optima, as shown in Fig. 3(a). f_2 has two global optima and two local optima, as shown in Fig. 3(b). Fig. 4 shows the errors between the true optima and the found optima for f_1 and f_2. To plot, e.g., the boxplot for Global Optimum 1 of f_1, we first find all fireflies within the bottom left quadrant of the search space, and then identify their maximum. Fig. 4 shows that generally SMFA gave solutions closer to the global and local optima than the other six algorithms. Furthermore, the outputs of SMFA were also more consistent, as the heights of its boxes were generally smaller.

Next, the six algorithms were compared on six low-dimensional maximization functions ($f_3 - f_8$). The results are shown in Fig. 5, and the mean and standard deviation of the errors are shown in Tables 2 and 3. Again, SMFA gave overall better and more consistent results.

Finally, the six algorithms were compared on seven high-dimensional minimization functions ($f_9 - f_{15}$). The results are shown in Fig. 6 and Tables 2 and 3. Once again, SMFA gave overall better and more consistent results.

In summary, extensive experiments showed that our proposed SMFA can indeed find multiple global and local optima simultaneously, and its solutions are better and more consistent than those from six other FA variants in the literature.

V. CONCLUSIONS

In this paper, we have proposed SMFA, which explicitly switches from exploration to exploitation in the search process to facilitate high-performance global

TABLE 3. Standard deviations of the errors of different FAs on the 15 benchmark functions.

Function	SMFA	CFA	WSSFA	VSSFA	SAFA	SFA	NAFA
f_1 $GO1:$	$4.55E-01$	$5.83E-01$	$5.23E-01$	$\mathbf{3.71E-01}$	$3.95E-01$	$3.14E-01$	$1.09E-00$
f_1 $GO2:$	$2.79E-01$	$7.14E-01$	$5.52E-01$	$3.03E-01$	$\mathbf{6.62E-02}$	$3.24E-01$	$1.09E-00$
f_1 $GO3:$	$2.64E-01$	$\mathbf{2.03E-01}$	$4.76E-01$	$2.92E-01$	$2.50E-01$	$2.79E-01$	$1.11E-00$
f_1 $GO4:$	$5.29E-01$	$6.43E-01$	$4.26E-01$	$4.33E-01$	$3.52E-01$	$\mathbf{3.38E-01}$	$1.07E+00$
f_2 $GO1:$	$2.77E-09$	$1.10E-01$	$4.14E-01$	$2.79E-01$	$1.82E-06$	$2.00E-01$	$\mathbf{0.00E-00}$
f_2 $GO2:$	$\mathbf{3.72E-10}$	$3.87E-05$	$2.58E-01$	$2.81E-01$	$1.41E-01$	$2.31E-02$	$2.50E-03$
f_2 $LO1:$	$2.96E-01$	$2.17E-01$	$2.38E-01$	$2.17E-01$	$2.05E-06$	$1.95E-01$	$\mathbf{0.00E-00}$
f_2 $LO2:$	$2.64E-01$	$1.63E-01$	$1.82E-01$	$2.61E-01$	$3.27E-01$	$2.72E-01$	$\mathbf{0.00E-00}$
f_3	$\mathbf{2.80E-01}$	$5.86E-01$	$6.38E-01$	$8.93E-01$	$7.89E-01$	$5.54E-01$	$4.39E-01$
f_4	$2.56E-01$	$1.41E-01$	$5.01E-01$	$3.59E-01$	$4.85E-01$	$\mathbf{8.80E-04}$	$3.55E-01$
f_5	$\mathbf{2.97E-01}$	$9.71E-01$	$1.55E+00$	$1.60E+00$	$9.50E-01$	$9.46E-01$	$1.10E+00$
f_6	$3.36E-01$	$\mathbf{4.65E-03}$	$2.58E-01$	$3.91E-01$	$3.01E-01$	$2.85E-01$	$2.97E-01$
f_7	$2.47E-04$	$\mathbf{2.12E-04}$	$1.22E+00$	$6.71E+00$	$3.77E+00$	$1.09E-01$	$7.07E+00$
f_8	$3.19E+01$	$6.97E-00$	$\mathbf{1.55E-01}$	$5.94E+01$	$3.07E+01$	$5.50E+01$	$1.54E+01$
f_9	$\mathbf{3.89E-11}$	$3.09E-06$	$1.36E+01$	$1.37E+03$	$9.67E-07$	$2.07E-01$	$1.48E-11$
f_{10}	$\mathbf{2.65E-05}$	$3.96E-01$	$2.77E-01$	$7.76E+00$	$5.17E-01$	$7.01E-02$	$7.97E-03$
f_{11}	$\mathbf{1.06E-06}$	$3.65E-00$	$1.83E-01$	$4.39E+01$	$3.95E-00$	$6.03E+00$	$4.00E-03$
f_{12}	$9.92E+02$	$1.46E+03$	$1.96E+03$	$\mathbf{5.23E+02}$	$1.23E+03$	$1.76E+03$	$8.68E+02$
f_{13}	$\mathbf{5.26E+00}$	$1.00E+01$	$1.84E+01$	$1.53E+01$	$1.84E+01$	$1.99E+01$	$7.88E+00$
f_{14}	$\mathbf{1.26E-11}$	$4.98E-10$	$1.11E-08$	$3.72E-05$	$1.85E-09$	$1.05E-07$	$1.30E-09$
f_{15}	$\mathbf{2.44E-10}$	$6.94E-09$	$8.09E-02$	$6.24E-05$	$5.62E-07$	$2.20E-01$	$4.33E-06$

optimization. A fixed randomization parameter is used in exploration, whereas a gradually decreasing random randomization parameter is used in exploitation. The condition for switching from exploration to exploitation is identified automatically. Extensive experiments on 15 benchmark functions verified the effectiveness of the proposed approach. Because of its reliable performance and easy implementation, SMFA is a promising approach in real-world optimization problems.

REFERENCES

[1] I. Fister, I. Fister, Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm Evol. Comput.*, vol. 13, pp. 34–46, Dec. 2013.

[2] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[3] M. H. Tayarani, X. Yao, and H. Xu, "Meta-heuristic algorithms in car engine design: A literature survey," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 609–629, Oct. 2015.

[4] D. Wu and W. W. Tan, "Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers," *Eng. Appl. Artif. Intell.*, vol. 19, no. 8, pp. 829–841, 2006.

[5] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[7] J. Huang, F. Ding, T. Fukuda, and T. Matsuno, "Modeling and velocity control for a novel narrow vehicle based on mobile wheeled inverted pendulum," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1607–1617, Sep. 2013.

[8] J. Huang, J. Qian, L. Liu, Y. Wang, C. Xiong, and S. Ri, "Echo state network based predictive control with particle swarm optimization for pneumatic muscle actuator," *J. Franklin Inst.*, vol. 353, no. 12, pp. 2761–2782, 2016.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Symp. Neural Netw.*, Dec. 1995, pp. 1942–1948.

[10] L. Liu, Y. Wang, F. Xie, and J. Gao, "Legendre cooperative PSO strategies for trajectory optimization," *Complexity*, vol. 2018, Apr. 2018, Art. no. 5036791.

[11] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[12] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.

[13] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, no. 3, 2013, Art. no. 35.

[14] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2010.

[15] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Proc. Int. Symp. Stochastic Algorithms*, vol. 5792, 2009, pp. 169–178.

[16] R. Arul, S. Velusami, and G. Ravi, "Chaotic firefly algorithm to solve economic load dispatch problems," in *Proc. Int. Conf. Green Comput., Commun. Conservation Energy (ICGCE)*, 2013, pp. 458–464.

[17] A. Rahmani and S. A. MirHassani, "A hybrid firefly-genetic algorithm for the capacitated facility location problem," *Inf. Sci.*, vol. 283, pp. 70–78, Nov. 2014.

[18] M.-H. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 1078–1091, 2012.

[19] T. Niknam, R. Azizipanah-Abarghooee, and A. Roosta, "Reserve constrained dynamic economic dispatch: A new fast self-adaptive modified firefly algorithm," *IEEE Syst. J.*, vol. 6, no. 4, pp. 635–646, Dec. 2012.

[20] A. Baykasoğlu and F. B. Ozsoydan, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3712–3725, 2014.

[21] X. Chen, M. Zhou, J. Huang, and Z. Luo, "Global path planning using modified firefly algorithm," in *Proc. Int. Symp. Micro-NanoMechatron. Hum. Sci. (MHS)*, Dec. 2017, pp. 1–7.

[22] C. Zhao *et al.*, "Decomposition-based multi-objective firefly algorithm for RFID network planning with uncertainty," *Appl. Soft Comput.*, vol. 55, pp. 549–564, Jun. 2017.

[23] H. Su, Y. Cai, and Q. Du, "Firefly-algorithm-inspired framework with band selection and extreme learning machine for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 1, pp. 309–320, Jan. 2017.

[24] I. Fister, X.-S. Yang, J. Brest, and I. Fister, Jr., "Modified firefly algorithm using quaternion representation," *Expert Syst. Appl.*, vol. 40, no. 12, pp. 7220–7230, 2013.

[25] S. Yu, S. Su, Q. Lu, and L. Huang, "A novel wise step strategy for firefly algorithm," *Int. J. Comput. Math.*, vol. 91, no. 12, pp. 2507–2513, 2014.

[26] S. Yu, S. Zhu, Y. Ma, and D. Mao, "A variable step size firefly algorithm for numerical optimization," *Appl. Math. Comput.*, vol. 263, pp. 214–220, Jul. 2015.

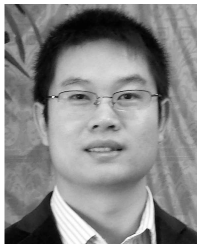
[27] H. Wang *et al.*, "Firefly algorithm with neighborhood attraction," *Inf. Sci.*, vols. 382–383, pp. 374–387, Mar. 2017.

[28] H. Wang *et al.*, "A hybrid multi-objective firefly algorithm for big data optimization," *Appl. Soft Comput.*, vol. 69, pp. 806–815, Aug. 2018.

[29] Y. Tang, Z. Wang, and J.-A. Fang, "Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2523–2535, 2011.
[30] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann. Math. Statist.*, vol. 2, no. 29, pp. 610–611, 1958.
[31] A. C. Cohen, Jr., "On estimating the mean and standard deviation of truncated normal distributions," *J. Amer. Stat. Assoc.*, vol. 44, no. 248, pp. 518–525, 1949.
[32] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, 2013.

JIAN HUANG (M'07–SM'17) received the B.E., M.Eng., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1997, 2000, and 2005, respectively. From 2006 to 2008, he was a Post-Doctoral Researcher with the Department of Micro-Nano System Engineering and the Department of Mechano-Informatics and Systems, Nagoya University, Japan. He is currently a Full Professor with the School of Automation, HUST. His main research interests include rehabilitation robot, robotic assembly, networked control systems, and bioinformatics.

XIAOCHAO CHEN (S'17) received the bachelor's degree from the School of Electrical Information, Sichuan University, Chengdu, China, in 2015. He is currently pursuing the master's degree with the School of Automation, Huazhong University of Science and Technology, Wuhan, China. His main research interest is swarm intelligence and robot path planning.



DONGRUI WU (S'05–M'09–SM'14) received the B.E. degree in automatic control from the University of Science and Technology of China in 2003, the M.E. degree in electrical engineering from the National University of Singapore in 2005, and the Ph.D. degree in electrical engineering from the University of Southern California in 2009. He is currently a Professor with the School of Automation, Huazhong University of Science and Technology, Wuhan, China. He has more than 100 publications, including a book entitled *Perceptual Computing* (Wiley–IEEE, 2010). His research interests include affective computing, brain–computer interface, computational intelligence, and machine learning.

Dr. Wu received the IEEE CIS Outstanding Ph.D. Dissertation Award in 2012, the IEEE Transactions on Fuzzy Systems Outstanding Paper Award in 2014, the NAFIPS Early Career Award in 2014, and the IEEE SMC Society Early Career Award in 2017. He is an Associate Editor of the IEEE Transactions on Fuzzy Systems, the IEEE Transactions on Human-Machine Systems, and the *IEEE Computational Intelligence Magazine*.