Dynamic Search in Fireworks Algorithm

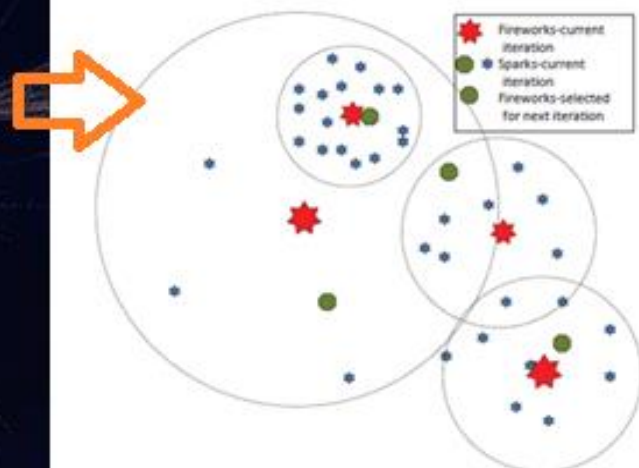
Shaoqiu Zheng, Andreas Janecek, Junzhi Li, Ying Tan
zhengshaoqiu@pku.edu.cn
Schol of EECS, Peking University
2014/7/3

Outline

- Introduction
- The Framework of (Enhanced Fireworks Algorithm)EFWA
- Properties of EFWA's Minimal Explosion Amplitude Check Strategy
- The Proposed dynFWA
- Experiments
- Conclusion

Introduction

- Motivation

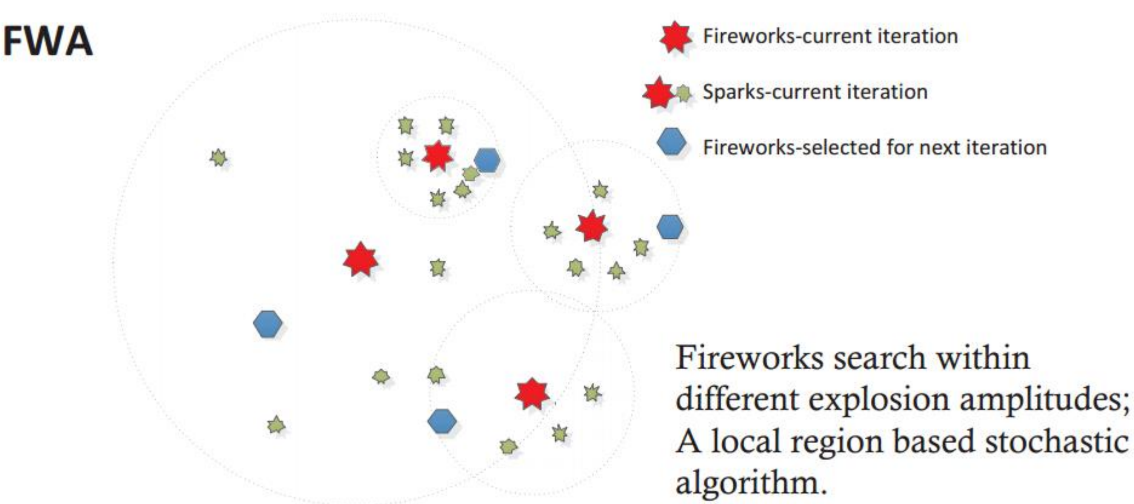


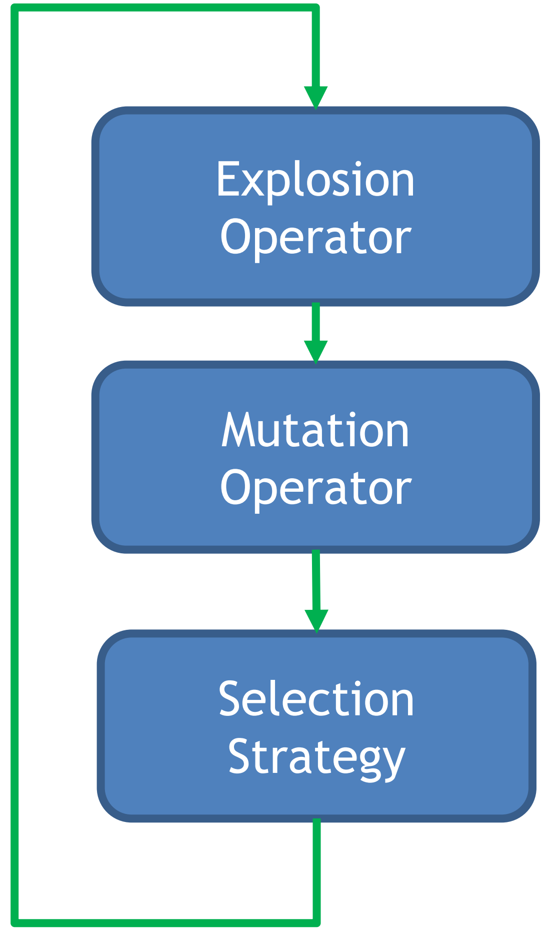
Fireworks Explosion

Search in the solution space

Introduction

- Framework of FWA





Introduction

- History

2010

1. FWA

2011

2. Digital filter Design
3. NMF
4. 0/1 problem
5. CA-FWA

2012

6. AcFWA
7. FWA-DE

2013

8. EFWA
9. IFWA
10. GPU-FWA
11. Swarm Robots
12. Equations Problems
13. MOFWA
14. Spam Detection
15. Image Recognition

2014

16. dynFWA
17. AFWA
18. FWA-DM
19. Convergence Analysis
20. BBO-FWA

The Framework of EFWA

$$A_i = \hat{A} \cdot \frac{f(X_i) - y_{min} + \varepsilon}{\sum_{i=1}^{N} (f(X_i) - y_{min}) + \varepsilon}$$

$$s_i = M_e \cdot \frac{y_{max} - f(X_i) + \varepsilon}{\sum_{i=1}^{N} (y_{max} - f(X_i)) + \varepsilon}$$

where $y_{max} = max(f(X_i))$, $y_{min} = min(f(X_i))$, and $\hat{A}$ and M_e are two constants to control the explosion amplitude and the number of explosion sparks, respectively, and ε is the machine epsilon. Additionally, the number of sparks s_i that can be generated by each firework is limited by an upper bound.

Algorithm 1 – General structure of EFWA.

Initialize N fireworks X_i and constant parameters
repeat
$\boxed{\textit{Explosion sparks search}}$
 (i) Calculate explosion amplitude and explosion sparks number
 (ii) Generate the explosion sparks
 (iii) Map back if out of bounds for newly created explosion sparks
 (iv) Evaluate fitness of newly created explosion sparks
$\boxed{\textit{Gaussian sparks search}}$
 (i) Generate the Gaussian sparks
 (ii) Map back if out of bounds for each created Gaussian sparks
 (iii) Evaluate fitness of newly created Gaussian sparks
$\boxed{\textit{Survival of the brightest (fittest)}}$
 (i) Select fireworks for next iteration (Section III-D)
until termination (time, max. #evals, convergence, ...)

The Framework of EFWA

- MEAC

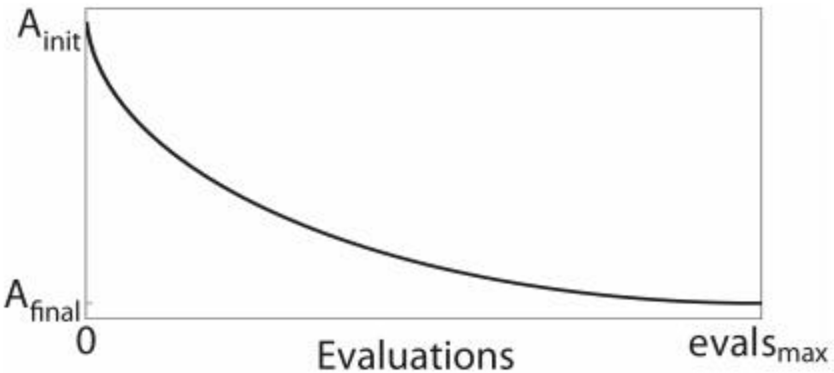


Fig. 1. Non-linearly decreasing minimal explosion amplitude

$$A_i^k = \begin{cases} A_{\min}^k & \text{if } A_i^k < A_{\min}^k, \\ A_i^k & otherwise, \end{cases}$$

$$A_{\min}^k = A_{init} - \frac{A_{init} - A_{final}}{evals_{max}} \sqrt{(2 * evals_{max} - t)\, t},$$

Properties of EFWA's Minimal Explosion Amplitude Check Strategy

Definition 1

Core Firework: In each iteration, the firework at the currently best location is marked as core firework (CF). Thus, for minimization problems, among the set C of all fireworks the firework X_{CF} is selected as CF iff

$$\forall X_i \in C: f(X_{CF}) \leq f(X_i)$$

Definition 2

Local Minimum Space and Local Minimum Point: Given an objective function f, in a continuous space Ψ ⊆ Ω, there ∃ only one point x, ∃ε, and f(xi) − f(x) > 0, for ∀xi ,|xi − x| <= ε, then x is a local minimum point. For region S, if these is only one local minimal point in it, then S is a local minimum space.

北京大学 计算智能实验室

Computational Intelligence Laboratory, Peking University

Properties of EFWA's Minimal Explosion Amplitude Check Strategy

- MEAC $$A_{\min}^{k} = A_{init} - \frac{A_{init} - A_{final}}{evals_{max}} \sqrt{(2 * evals_{max} - t\,)\,t},$$

the early phase of the algorithm, larger Akmin ⇒ global search,
The final phase of the algorithm, smaller Akmin ⇒ local search.

Problems:

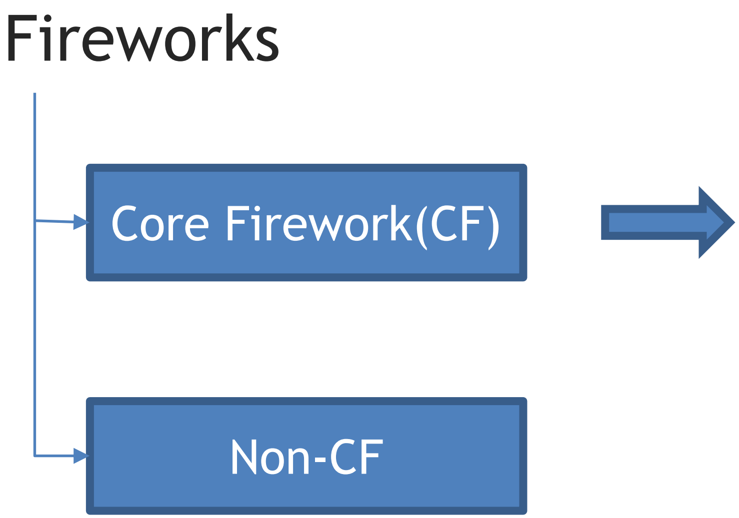
this procedure decreases the explosion amplitude threshold solely with the current number of function evaluations which heavily depends on the pre-defined number of iterations for the algorithm.

The Proposed dynFWA

Fireworks

- Core Firework(CF) → **Smaller Explosion Amplitude**
  Local Search/ Global Search
  CF is always selected
- nonCF → **Bigger Explosion Amplitude**
  Global Search

The Proposed dynFWA



Algorithm 2 Dynamic explosion amplitude update for CF

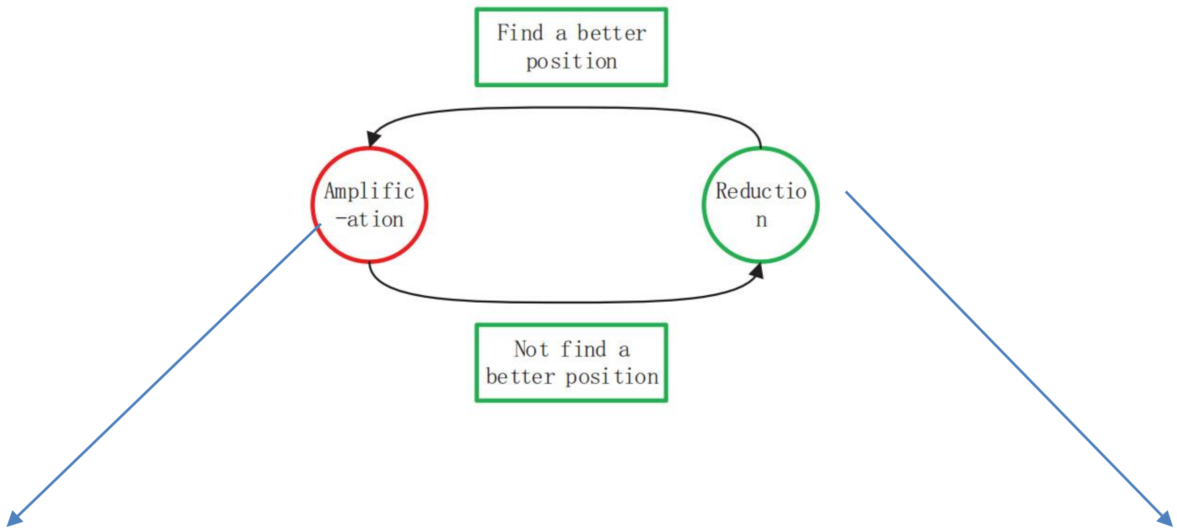
Initialization: Define:

X_{CF} is the current location of the CF;
$\hat{X}_b$ is the best location among all explosion sparks;
A_{CF} is the current explosion amplitude of the CF;
C_a is the amplification coefficient;
C_r is the reduction coefficient;

Iteration:

1: **if** $f(\hat{X}_b) - f(X_{CF}) < 0$ **then**
2: $\quad A_{CF} \leftarrow A_{CF} * C_a$;
3: **else**
4: $\quad A_{CF} \leftarrow A_{CF} * C_r$;
5: **end if**

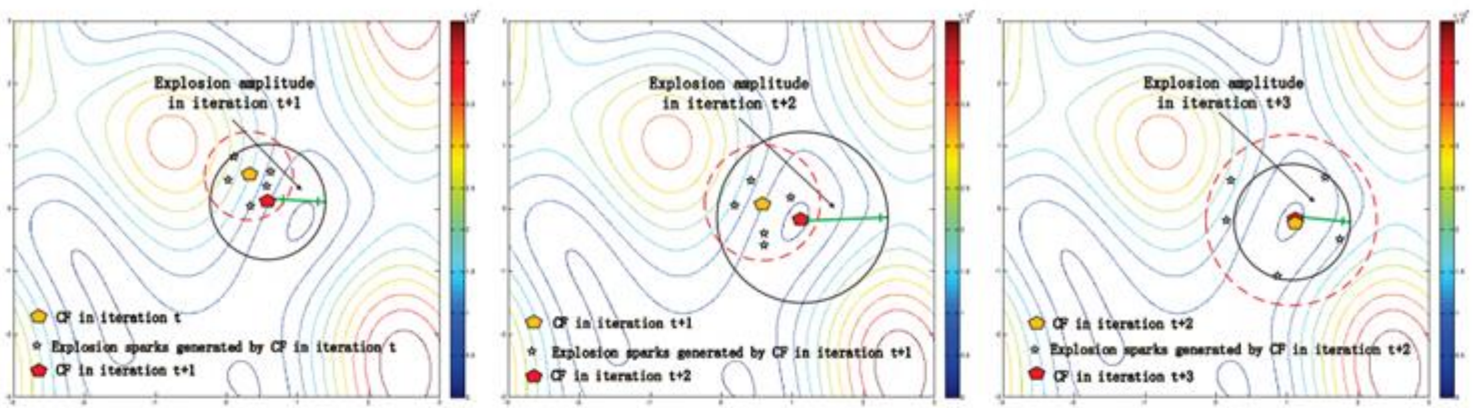
The Proposed dynFWA



Accelerate the convergence speed

Move towards to global optimum,
The fireworks swarm can get a better Position.

The Proposed dynFWA



(a) Iteration $t+1$ (b) Iteration $t+2$ (c) Iteration $t+3$

Fig 2. Illustration of the amplification/reduction of the CF's explosion amplitude. In Figure 1(a), the radius of the circle with dashed red line indicates the explosion amplitude of the CF in iteration t, while the cirlce with solid black line indicates the explosion amplitude in iteration $t+1$; the increased explosoin amplitude indicates that in this situation, a better position has been found by the explosion sparks. In iteration $t+2$ (cf. Figure 1(b)), the CF is able to further improve its location, and, as a result, the explosion amplitude of the CF is further increased. Figure 1(c) shows an example when the fitness of the CF could not be improved. In this case, the CF's explosion amplitude is decreased in iteration $t+3$.

The Proposed dynFWA

- Convergence

A Taylor series is used to represent the properties of the local region around the CF.

Assume a continuously differentiable second-order optimization function f with k dimensions: if the position of the CF is not a local/global minimal point, and ACF is the current explosion amplitude, then

$$g(\mathbf{x}) - g(X_{CF}) = \nabla g(X_{CF})^T(\mathbf{x} - X_{CF}) + \frac{1}{2}(\mathbf{x} - X_{CF})H(\mathbf{x})(\mathbf{x} - X_{CF}),$$

where

$$H(\mathbf{x}) = [\frac{\partial^2 g}{\partial x_i \partial x_j}]_{k\times k}.$$

According to the definition of “local/global minimal point”, there ∃ε, ∀x in

$$S = \{\mathbf{x} | |\mathbf{x} - X_{CF}| \leqslant \varepsilon\},$$

and

$$g(\mathbf{x}) - g(X_{CF}) = \nabla g(X_{CF})^T(\mathbf{x} - X_{CF}) + o(\nabla g(X_{CF})^T(\mathbf{x} - X_{CF})).$$

北京大学 计算智能实验室

Computational Intelligence Laboratory, Peking University

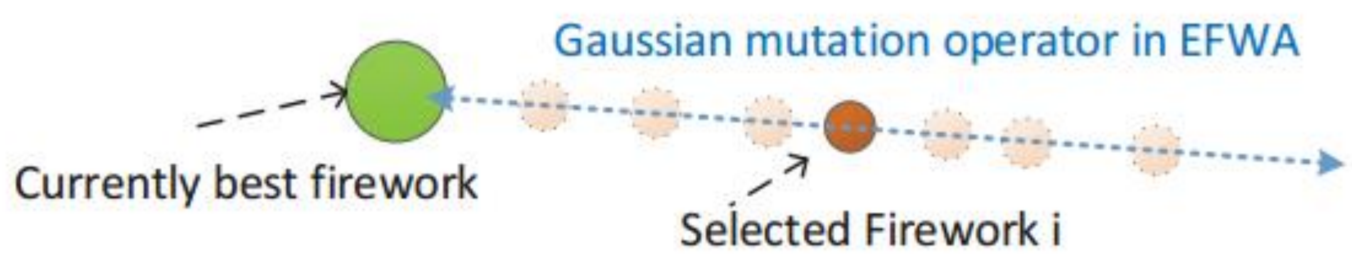
The Proposed dynFWA

- Convergence

From the Taylor series, if $\varepsilon \to 0$, then in region S, if there exists a point $\mathbf{x}_1$ and $\mathbf{x}_1 - X_{CF} = \Delta\mathbf{x}$, then there exists a point $\mathbf{x}_2$ and $\mathbf{x}_2 - X_{CF} = -\Delta\mathbf{x}$. Under this circumstance, the probability of generating a spark with smaller fitness than the CF is very high (i.e. $(g(\mathbf{x}_1) - g(X_{CF})) * (g(\mathbf{x}_2) - g(X_{CF})) < 0$). In case the CF does not find a better position while generating a number of sparks, it is likely that $A_{CF} \geqslant \varepsilon$. We cannot expect the property of region $T = \{\mathbf{x} | \varepsilon \leqslant |\mathbf{x} - X_{CF}| \leqslant A_{CF}\}$ that in region T whether there exists a position with better fitness compared to the CF, thus, if the CF generates sparks with uniform distribution in each dimension, the probability p' that a spark is located in S is $p' = \frac{\|S\|}{\|S\|+\|T\|}$, where $\|\|$ denotes the hypervolume of this region. If the CF does not find a better position, the explosion amplitude A_{CF} is reduced in order to increase the probability p' that the CF can generate a spark in region S thus to increase the probability that finding a point with smaller fitness than CF.

The Proposed dynFWA

- Elimination of the Gaussian Sparks Operator



Not effective as they are designed to be!

The Proposed dynFWA

Algorithm 3 Framework of dynFWA

```
Initialize N fireworks and evaluate the quality
Initialize the explosion amplitude for CF
while termination criteria are not met do
    Calculate number of explosion sparks
    Calculate explosion amplitude for non-CF
    for each firework do
        Generate explosion sparks
        Map sparks at invalid locations back to search space
        Evaluate quality of explosion sparks
    end for
    Calculate explosion amplitude of CF
    Select N fireworks for next iteration
end while
```

Experiments

Design

Algorithm

- EFWA – the baseline algorithm;
- EFWA-NG – in this algorithm, the Gaussian sparks operator has been removed from EFWA;
- dynFWA-G – this algorithm implements the dynFWA algorithm as described in Section IV including the Gaussian mutation operator.
- dynFWA – similar as dynFWA-G but without Gaussian mutation operator.
- SPSO2011 – the most recent SPSO variant.

Benchmark

- CEC2013 Competition Problems

Setup

The reduction and amplification factors Cr and Ca of dynFWA are empirically set to 0.9 and 1.2

the number of fireworks in dynFWA
is set to 5, but in dynFWA, the maximum number of explosion sparks (Me in Eq. 1) in each iteration is set to 150

Experiments

Evaluation of Gaussian sparks operator

For EFWA these results suggest that the Gaussian sparks operator should not be removed；

dynFWA without the Gaussian sparks opera slightly better results than dynFWA-G.

TABLE I

WILCOXON SIGNED-RANK TEST RESULTS FOR EFWA *vs* EFWA-NG AND DYNFWA-G *vs* DYNFWA (BOLD VALUES INDICATE THE PERFORMANCE DIFFERENCE IS SIGNIFICANT).

F.	EFWA *vs* EFWA-NG			dynFWA-G *vs* dynFWA		
	EFWA	EFWA-NG	p-value	dynFWA-G	dynFWA	p-value
f_1	-1.3999E+03	-1.3999E+03	**2.316E-03**	-1.4000E+03	-1.4000E+03	1.000E+00
f_2	6.8926E+05	6.5258E+05	4.256E-01	7.6981E+05	8.6937E+05	1.801E-01
f_3	7.7586E+07	6.4974E+07	8.956E-01	1.2007E+08	1.2317E+08	6.393E-01
f_4	-1.0989E+03	-1.0989E+03	7.858E-01	-1.0863E+03	-1.0896E+03	**3.183E-02**
f_5	-9.9992E+02	-9.9992E+02	**4.290E-02**	-1.0000E+03	-1.0000E+03	1.463E-01
f_6	-8.5073E+02	-8.4462E+02	1.654E-01	-8.6524E+02	-8.6995E+02	9.156E-02
f_7	-6.2634E+02	-6.2991E+02	9.552E-01	-6.9946E+02	-7.0010E+02	6.663E-01
f_8	-6.7907E+02	-6.7906E+02	9.776E-01	-6.7909E+02	-6.7910E+02	4.997E-01
f_9	-5.6846E+02	-5.6889E+02	5.178E-01	-5.7435E+02	-5.7587E+02	1.711E-01
f_{10}	-4.9916E+02	-4.9918E+02	3.732E-01	-4.9994E+02	-4.9995E+02	3.591E-01
f_{11}	5.8198E+00	3.5430E+01	5.830E-02	-2.9978E+02	-2.9589E+02	6.127E-01
f_{12}	3.9944E+02	4.1107E+02	6.193E-01	-1.4993E+02	-1.4222E+02	4.762E-01
f_{13}	2.9857E+02	2.8909E+02	8.220E-01	5.4523E+01	5.3830E+01	8.513E-01
f_{14}	2.7240E+03	2.9344E+03	**4.101E-02**	2.8909E+03	2.9180E+03	8.147E-01
f_{15}	4.4595E+03	4.5515E+03	6.869E-01	3.9186E+03	4.0227E+03	4.879E-01
f_{16}	2.0063E+02	2.0056E+02	2.811E-01	2.0056E+02	2.0058E+02	7.358E-01
f_{17}	6.2461E+02	6.3152E+02	9.179E-01	4.5397E+02	4.4261E+02	1.197E-01
f_{18}	5.7361E+02	5.6953E+02	6.938E-01	5.8801E+02	5.8782E+02	8.660E-01
f_{19}	5.1022E+02	5.1012E+02	9.402E-01	5.0750E+02	5.0726E+02	6.193E-01
f_{20}	6.1466E+02	6.1457E+02	**1.559E-02**	6.1309E+02	6.1328E+02	3.632E-01
f_{21}	1.1178E+03	1.1362E+03	**6.910E-04**	9.9532E+02	1.0102E+03	6.431E-01
f_{22}	6.3181E+03	6.3674E+03	9.776E-01	4.1463E+03	4.1262E+03	9.402E-01
f_{23}	7.5809E+03	7.5707E+03	7.217E-01	5.6661E+03	5.6526E+03	9.402E-01
f_{24}	1.3452E+03	1.3611E+03	**1.079E-02**	1.2738E+03	1.2729E+03	8.586E-01
f_{25}	1.4426E+03	1.4435E+03	8.734E-01	1.3964E+03	1.3970E+03	8.882E-01
f_{26}	1.5461E+03	1.5400E+03	2.687E-01	1.4744E+03	1.4607E+03	1.337E-01
f_{27}	2.6210E+03	2.5780E+03	3.534E-01	2.2721E+03	2.2804E+03	8.147E-01
f_{28}	4.7651E+03	4.9949E+03	6.460E-01	1.7686E+03	1.6961E+03	3.555E-01

Experiments

Comparison of dynFWA and EFWA

dynFWA achieves better mean fitness results than EFWA on 23 functions.

The test results indicate that the improvement of dynFWA is significant compared to EFWA for 22 benchmark functions.

TABLE II

MEAN FITNESS ON THE BENCHMARK FUNCTIONS AND MEAN FITNE

RANK OF SPSO2011, EFWA AND DYNFWA.

F.	SPSO2011	Rank	EFWA	Rank	dynFWA	Rank
f_1	**-1.4000E+03**	**1**	-1.3999E+03	3	**-1.4000E+03**	**1**
f_2	**3.3719E+05**	**1**	6.8926E+05	2	8.6937E+05	3
f_3	2.8841E+08	3	**7.7586E+07**	**1**	1.2317E+08	2
f_4	3.7543E+04	3	**-1.0989E+03**	**1**	-1.0896E+03	2
f_5	**-1.0000E+03**	**1**	-9.9992E+02	3	-1.0000E+03	2
f_6	-8.6210E+02	2	-8.5073E+02	3	**-8.6995E+02**	**1**
f_7	**-7.1208E+02**	**1**	-6.2634E+02	3	-7.0010E+02	2
f_8	-6.7908E+02	2	-6.7907E+02	3	**-6.7910E+02**	**1**
f_9	-5.7123E+02	2	-5.6846E+02	3	**-5.7587E+02**	**1**
f_{10}	-4.9966E+02	2	-4.9916E+02	3	**-4.9995E+02**	**1**
f_{11}	-2.9504E+02	2	5.8198E+00	3	**-2.9589E+02**	**1**
f_{12}	**-1.9604E+02**	**1**	3.9944E+02	3	-1.4222E+02	2
f_{13}	**-6.1406E+00**	**1**	2.9857E+02	3	5.3830E+01	2
f_{14}	3.8910E+03	3	**2.7240E+03**	**1**	2.9180E+03	2
f_{15}	**3.9093E+03**	**1**	4.4595E+03	3	4.0227E+03	2
f_{16}	2.0131E+02	3	2.0063E+02	2	**2.0058E+02**	**1**
f_{17}	**4.1626E+02**	**1**	6.2461E+02	3	4.4261E+02	2
f_{18}	**5.2063E+02**	**1**	5.7361E+02	2	5.8782E+02	3
f_{19}	5.0951E+02	2	5.1022E+02	3	**5.0726E+02**	**1**
f_{20}	6.1346E+02	2	6.1466E+02	3	**6.1328E+02**	**1**
f_{21}	**1.0088E+03**	**1**	1.1178E+03	3	1.0102E+03	2
f_{22}	5.0988E+03	2	6.3181E+03	3	**4.1262E+03**	**1**
f_{23}	5.7313E+03	2	7.5809E+03	3	**5.6526E+03**	**1**
f_{24}	**1.2667E+03**	**1**	1.3452E+03	3	1.2729E+03	2
f_{25}	1.3993E+03	2	1.4426E+03	3	**1.3970E+03**	**1**
f_{26}	1.4861E+03	2	1.5461E+03	3	**1.4607E+03**	**1**
f_{27}	2.3046E+03	2	2.6210E+03	3	**2.2804E+03**	**1**
f_{28}	1.8013E+03	2	4.7651E+03	3	**1.6961E+03**	**1**
			Mean Rank			
	SPSO2011	1.75	EFWA	2.68	dynFWA	**1.54**

TABLE III

WILCOXON SIGNED-RANK TEST RESULTS FOR DYNFWA *vs.* EFWA

(BOLD VALUES INDICATE THE SIGNIFICANT IMPROVEMENT).

F.	f_1	f_2	f_3	f_4	f_5	f_6	f_7
p-value	**0.00E+00**	6.94E-03	9.90E-02	0.00E+00	**0.00E+00**	**1.58E-03**	**0.00E+00**
F.	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
p-value	**1.73E-02**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.41E-01
F.	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}
p-value	**5.10E-05**	3.20E-01	**0.00E+00**	6.35E-02	**1.41E-04**	**0.00E+00**	**0.00E+00**
F.	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}	f_{28}
p-value	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Experiments

Comparison of dynFWA and SPSO2011

In total, dynFWA achieves better results (smaller mean fitness) than SPSO2011 on 17 functions, while SPSO2011 is better than dynFWA on 10 functions. For one function the results are identical.

Experiments

- **Runtime**

TABLE IV

RUNTIME COMPARISON.

	SPSO2007	1.01	EFWA	1.30	dynFWA-G	1.17
	SPSO2011	–	EFWA-NG	1.03	dynFWA	**1**

Conclusion

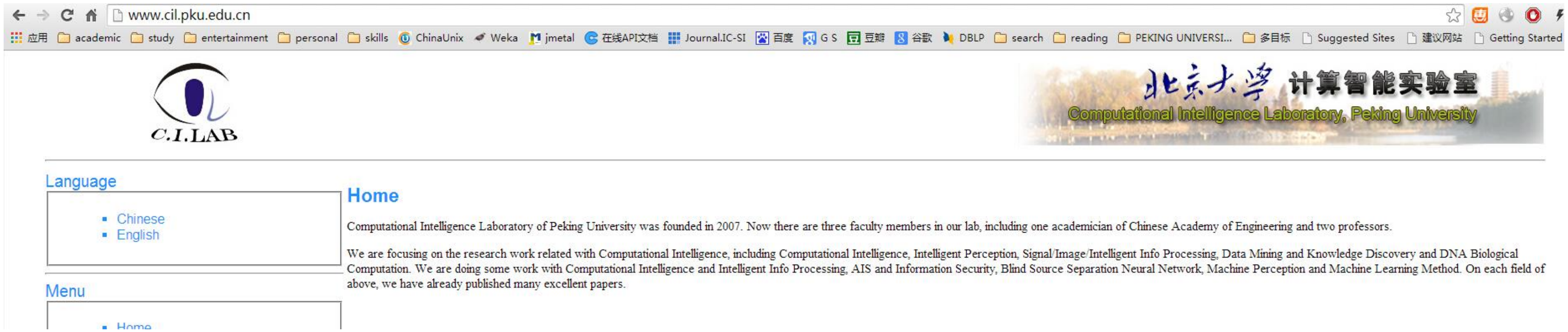
- The proposed dynFWA algorithm significantly improves the results of EFWA and also reduces the runtime by more than 20%.

- Compared with SPSO2011, dynFWA achieves a better mean rank among 28 benchmark functions with similar computational cost.

- The Gaussian sparks operator of EFWA should not be removed in EFWA. However, removing this operator in dynFWA significantly reduces the runtime of dynFWA without loss in optimization accuracy.

CIL@PKU

www.cil.pku.edu.cn



www.cil.pku.edu.cn

C.I.LAB

北京大學 计算智能实验室

Computational Intelligence Laboratory, Peking University

Language

- Chinese
- English

Menu

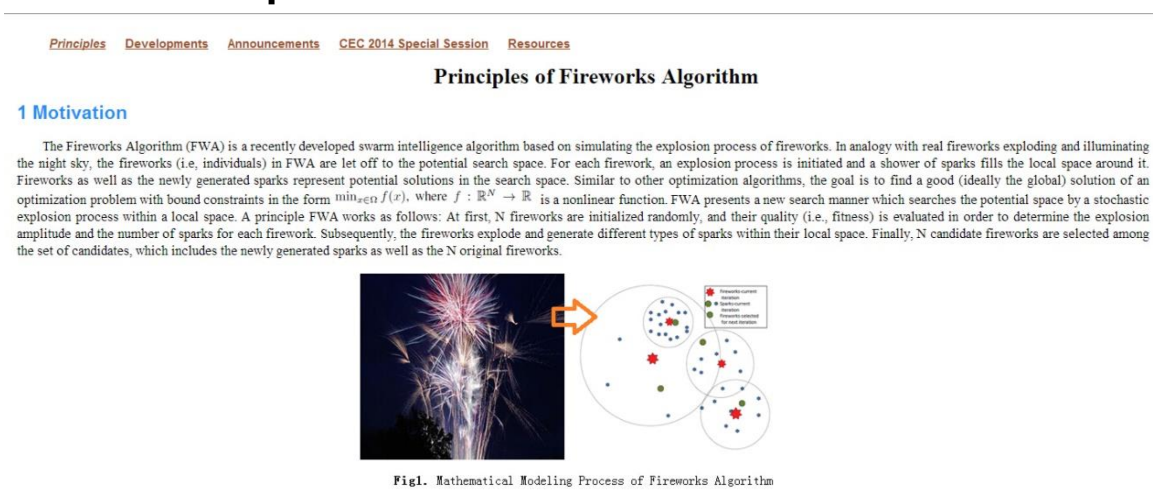
- Home

Home

Computational Intelligence Laboratory of Peking University was founded in 2007. Now there are three faculty members in our lab, including one academician of Chinese Academy of Engineering and two professors.

We are focusing on the research work related with Computational Intelligence, including Computational Intelligence, Intelligent Perception, Signal/Image/Intelligent Info Processing, Data Mining and Knowledge Discovery and DNA Biological Computation. We are doing some work with Computational Intelligence and Intelligent Info Processing, AIS and Information Security, Blind Source Separation Neural Network, Machine Perception and Machine Learning Method. On each field of above, we have already published many excellent papers.

http://www.cil.pku.edu.cn/research/FWA/index.html



Principles Developments Announcements CEC 2014 Special Session Resources

Principles of Fireworks Algorithm

1 Motivation

The Fireworks Algorithm (FWA) is a recently developed swarm intelligence algorithm based on simulating the explosion process of fireworks. In analogy with real fireworks exploding and illuminating the night sky, the fireworks (i.e, individuals) in FWA are let off to the potential search space. For each firework, an explosion process is initiated and a shower of sparks fills the local space around it. Fireworks as well as the newly generated sparks represent potential solutions in the search space. Similar to other optimization algorithms, the goal is to find a good (ideally the global) solution of an optimization problem with bound constraints in the form $\min_{x\in\Omega} f(x)$, where $f : \mathbb{R}^N \to \mathbb{R}$ is a nonlinear function. FWA presents a new search manner which searches the potential space by a stochastic explosion process within a local space. A principle FWA works as follows: At first, N fireworks are initialized randomly, and their quality (i.e., fitness) is evaluated in order to determine the explosion amplitude and the number of sparks for each firework. Subsequently, the fireworks explode and generate different types of sparks within their local space. Finally, N candidate fireworks are selected among the set of candidates, which includes the newly generated sparks as well as the N original fireworks.

Fig1. Mathematical Modeling Process of Fireworks Algorithm

计算智能实验室

Computational Intelligence Laboratory, Peking University

Reference

[1] Y. Tan and Y. Zhu (2010). Fireworks algorithm for optimization. ICSI 2010, Part I, LNCS 6145, pp. 355-364 [pdf]

[2] Shaoqiu Zheng, Andreas Janecek and Ying Tan, "Enhanced Fireworks Algorithm " 2013 IEEE Congress on Evolutionary Computation, (IEEE CEC 2013) , June 20-23, Fiesta Americana Grand Coral Beach Hotel, Cancun, Mexico, pp. 2069-2077. [pdf]

[3] Shaoqiu Zheng, Andreas Janecek, Junzhi Li and Ying Tan, "Dynamic Search in Fireworks Algorithm," 2014 IEEE Congress on Evolutionary Computation, (IEEE CEC 2014).

[4] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, “Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements,” in Evolutionary Computation (CEC), 2013 IEEE Congress on, 2013, pp. 2337–2344.