

# 改良型偵察戦略：花火アルゴリズムへの応用

余俊<sup>†</sup>, 高木英行<sup>††</sup>

新潟大学自然科学系<sup>†</sup>, 九州大学大学院芸術工学研究院<sup>††</sup>

## 1 はじめに

ビッグデータの時代になり、パーソナライズ化した要求も増えるようになった。従来の多くの最適化アルゴリズムだけでは、一般に多変数、高コスト、微分不可の特性を持つこれら多くの問題を完全に解くことは困難である。そのため自然選択理論から様々な個体ベースの進化計算、例えば、遺伝的アルゴリズム<sup>2)</sup>、PSO<sup>5)</sup>、差分進化<sup>6)</sup>、花火アルゴリズム (FWA)<sup>7)</sup>などを考案してきた。多くの研究者は、これらの進化計算アルゴリズムに新しい探索戦略を組み込みことで最適化特性の向上を図ってきた。

進化計算分野の中で、花火アルゴリズムは提案以来多くの関心を集め、色々な改良版が提案されてきた。例えば、Zhengらは基本花火アルゴリズムに五つの改良を加えて残されていた花火アルゴリズムの欠点の克服を図った<sup>12)</sup>。Yuらは花火アルゴリズム演算を改良し性能向上を図るために、花火炸裂の大きさ（探索範囲）を縮約する戦略<sup>8)</sup>と、多層炸裂戦略<sup>9)</sup>を提案した。花火アルゴリズムは色々な最適化問題に応用されている。例えば、多目的最適化への応用<sup>4)</sup>、多峰性最適化への応用<sup>10)</sup>、制約付最適化への応用<sup>1)</sup>などである。花火アルゴリズムは多くの複雑な実問題へも応用されている。まだ提案から10年と若い技術でありながら、大きな可能性を見せてきた。

本論文の目的は、筆者らが提案した偵察戦略<sup>11)</sup>に二つの改良を加えた適応偵察戦略を提案し、より花火アルゴリズムが強力になることを示すことである。第1の改良は、子スパーク個体を探索する炸裂範囲の中心位置を適応的に変化させることであり、第2の改良は、現在の探索結果に基づいて炸裂範囲の大きさを適応的に変え

ることである。第2の目的は、提案手法の動作メカニズムを解析しその適用について議論することである。最後に、今後のための議論点を示す。

## 2 花火アルゴリズム

花火大会を想像してもらえると分かるように、花火が炸裂する周辺に多くのスパークが散らばる美しいシーンを楽しむことができる。Tan等は、花火の炸裂過程が局所探索に相当し、花火炸裂の大きさを制御することで局所探索や大域的探索のように異なる探索特性を自由に切り替えることができることに着目した。更に分析から、花火から生成されるスパーク数を適応的に決定できることも明らかにした。こうして、複数の花火個体が協調する炸裂モデルに基づく花火アルゴリズムを提案した<sup>7)</sup>。主要4演算（初期化、炸裂、突然変異、選択）からなる花火アルゴリズムの基本形をFig. 1に示す。

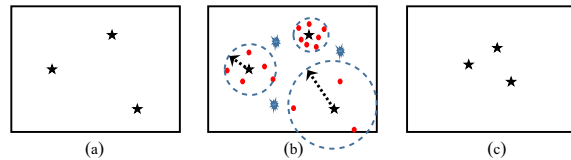


Fig. 1 基本的な花火アルゴリズムの最適化過程。(a) 複数の花火個体（黒★）がランダムに生成され初期個体群を形成。(b) 炸裂演算による花火個体の周辺のスパーク個体（赤●）の生成と、突然変異によるスパーク個体（青?）の生成。(c) 現世代の花火個体と全スパーク個体から次世代に残る新しい花火個体の選択。終了条件に達するまでステップ(b)と(c)の繰り返し。

花火アルゴリズムの特徴は、局所探索と広域探索のバランスを取るために、fitness情報を用いて炸裂演算パラメータを適応的に調整する点にある。fitnessが良い個体程狭い炸裂範囲に多くのスパーク個体を生成する。言い換えると、良い花火個体程局所探索に性能を発揮し、悪い個体

Improved Scouting Strategy: Application to Fireworks Algorithm

<sup>†</sup> Jun Yu (yujun@ie.niigata-u.ac.jp)

<sup>††</sup> Hideyuki Takagi (h.takagi.457@m.kyushu-u.ac.jp)

Institute of Science and Technology, Niigata University (<sup>†</sup>)  
Faculty of Design, Kyushu University (<sup>††</sup>)

程多様性維持のための広域探索を行う。花火個体の協調動作によって花火アルゴリズムは際立った最適化能力とロバスト性を示す。

### 3 適応型偵察戦略

#### 3.1 偵察戦略

実際の花火は、菊、牡丹、冠、千輪菊などの古典的名称が示すように、カスタマイズされた色々な形状を形成することができる。これらの現象観察から、徐々に炸裂してスパーク個体を生成していく偵察戦略、すなわち、一気にスパーク個体を生成するのではなくスパーク個体を一つずつ生成していく方法を提案した<sup>11)</sup>。この生成戦略は、未知の土地で安全なルートを探す斥候のような働きをする。

偵察戦略の特徴の一つは、初期化、突然変異、選択等の他の演算を変更することなくスパーク個体生成モードを変えることができる点である。偵察戦略は花火アルゴリズム基本形の炸裂演算と同じ方法を採用し、花火個体のfitness値に応じて炸裂の大きさと生成スパーク個体数を決める。唯一異なる点は、Fig. 2に示す偵察戦略がスパーク個体の一つずつ生成する点である。

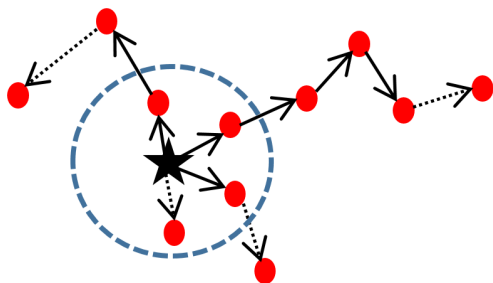


Fig. 2 花火個体周辺の偵察戦略例。黒★と赤●は各々花火個体とスパーク個体。実線矢印は現在の探索方向がうまくいっていることを示し、破線矢印は偵察を止め、最初（の花火個体）からランダムに偵察を再開することを示している。

最初に提案した偵察戦略<sup>11)</sup>では、各花火個体の周辺にスパーク個体を1個体だけ生成する。新規生成スパーク個体が元の花火個体よりも悪ければ、良いスパーク個体が得られるまで、花火個体周辺にランダムに新規スパーク個体を生成し続ける。花火個体よりも良いスパーク個体を得られた場合は、そのスパーク個体周辺に次のスパーク個体を生成する。第 $j$ スパーク個体周辺に生成した第 $(j+1)$ スパーク個体の方が良い限

り、この偵察戦略に基づく新規スパーク個体生成を継続するが、第 $(j+1)$ スパーク個体の方が劣る場合は、最初の花火個体に戻って、同様の処理を繰り返す。花火個体のfitnessで決められたスパーク個体数に達するまで、この逐次スパーク個体生成を繰り返す。最後に、全花火個体がこの偵察戦略で子スパーク個体を生成し終えたら、偵察戦略はうまく行ったと言える。

#### 3.2 適応偵察戦略

基本花火アルゴリズムで使われているランダム炸裂探索に比べると、偵察戦略は局所fitness景観情報を使う事で有望領域を早く見つけ、探索速度を速めることができる。このことは、多くの収集情報を効率的に使う事で花火アルゴリズムを一層改善でき得ることを示唆している。

分析から、過去提案の偵察戦略には改善すべき課題があることも分かっている。一旦親スパーク個体よりも劣る子スパーク個体を生成したら、それまで苦労して探し出した現在の有望領域を捨て、最初の花火個体まで戻って一から探索し直す。それまでの探索を打ち切ってそれまでの探索履歴情報を使わないとなると、全体の収束速度が悪くなるかもしれない。特に計算コストの高い複雑なタスクの場合には受け入れ難い。劣ったスパーク個体を生成したとしても一からやり直すのではなく、その時の探索範囲を変更することでそれまでの探索結果をある程度活かしたい。

これらより最初の提案の偵察戦略に二つの改良を加えた「適応偵察戦略」を提案する。第1の改良点は、劣ったスパーク個体を生成してしまった時に、最初の花火個体 (Fig. 2の5個の黒星) へ戻り、ランダムにスパーク個体を生成し直すのではなく、それまでの探索情報を有効に用いることである。一つの方法は、次の子スパーク個体生成範囲の中心点を初めの花火個体に戻さず、一つ前の親スパーク個体にするのである。この方法だとそれまでの探索結果をすべて捨てることなく、それまでの探索で得ている有望領域を維持しやすい。Fig. 3はこの提案戦略の炸裂効果の例を示す。

第2の改良点は、劣ったスパーク個体を生成してしまった時に探索範囲を調整し直して局所探索と大域探索のバランスを変え、最適化問題の特性により適応させることである。子個体が親個体よりも良い時は、その探索領域まだ有望であると言えるので、次のスパーク個体を探索す

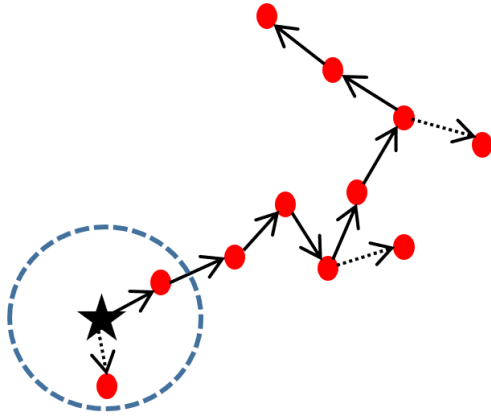


Fig. 3 花火個体周辺の適応偵察戦略の例. 現在の偵察方向がうまくいかなかった場合には, 最初の花火個体 (黒★) に戻って偵察を再開するのではなく, 一つ前の親スパーク個体に戻って再開する. この図の記号はFig. 2と同じである.

る炸裂範囲を小さくし, 局所探索を重視して収束精度を向上させる. そうでない場合は局所領域からの脱出を優先して, 炸裂範囲を大きくし他の有望な領域に早く移るようにする. Fig. 4にそれまでの探索結果に応じて炸裂範囲を調整する両者の例を示す. 基本的な花火アルゴリズムにこの新偵察戦略を組み込んだ一般的な最適化の枠組みをAlgorithm 1に示す.

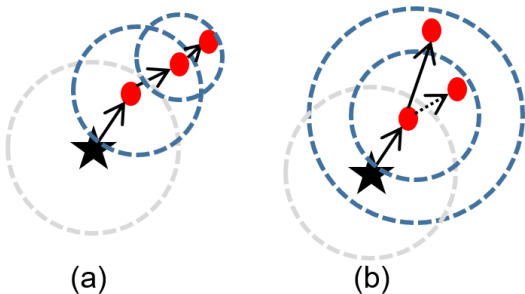


Fig. 4 炸裂の大きさ (探索範囲) は現在の探索結果に応じて適応的に調整される. (a)良い子個体が生成される程, 炸裂の大きさが徐々に小さくなる. (b) 劣ったスパーク個体が増える程, 炸裂範囲が大きくなる. 記号はFig. 2と同じである.

#### 4 評価実験

初期提案の偵察戦略であっても, 探索性能向上が明らかであったので<sup>11)</sup>, ここでは新規提案の適応偵察戦略の有無の効果を比較するのではなく, 初期提案の偵察戦略と改良型の適応偵察戦略の性

**Algorithm 1** 花火アルゴリズムの組み込んだ提案の一般的な枠組み. Step 5-11が提案戦略.

- 1:  $n$ 個の花火個体をランダムに配置して初期化
- 2: 各花火個体のfitness評価
- 3: **while** 終了条件が満たされていない間 **do**
- 4:   **for**  $i = 0; i < n; i++$  **do**
- 5:     提案偵察戦略でスパーク個体を1個生成
- 6:     そのスパーク個体のfitnessを計算
- 7:     **if** 子個体が親個体よりも良いならば **then**
- 8:       炸裂の大きさを小さくする
- 9:     **else**
- 10:      炸裂の大きさを大きくする
- 11:     **end if**
- 12:   **end for**
- 13:   ガウス突然変異で突然変異スパーク個体を生成しfitnessを評価.
- 14:   次世代用に $n$ 個の新花火個体を選択
- 15: **end while**
- 16: end of program.

能比較を行う. 公平な比較のために, enhanced FWA (EFWA)をベンチマークアルゴリズムにして以前の比較実験とまったく同じパラメータ設定にする. 適応偵察戦略で炸裂範囲を拡張する場合は1.5倍に, 縮小する場合は0.8倍にする.

CEC2013ベンチマークセット<sup>3)</sup>の28関数×3種類の次元数 (2-D, 10-D, 30-D) に対して, (EFWA + 偵察戦略) vs. (EFWA + 適応偵察戦略)を30試行行う. 終了条件は世代数ではなくfitness評価回数で行い, 終了時での両戦略で得られた最良fitness値の30平均の差の検定をWilcoxonの符号検定で行う. この結果をTable 1に示す.

#### 5 考察

まず, 性能価格比の点から提案の適応偵察戦略について考察する. 従来の偵察戦略では親スパーク個体と子スパーク個体を比較しながら有望領域を進み性能向上を目指したが, 探索過程での情報を次の探索に活かし切れていなかった. 具体的には一旦親よりも劣る子スパーク個体が得られると, それまでの情報を捨てて最初の探索開始点 (花火個体) に戻ってしまう. これはこれで局所解領域に捕まるリスクを軽減できるが, それまでうまく探索できていた過去探索情報を捨てるということは収束速度の低下を招き, 探索効

率を低下させることにもなる．そこで本論文では二つの改良を行い従来の偵察戦略を改良した．

Table 1 終了時点での2偵察戦略の30試行平均fitness値の差のWilcoxon符号検定結果． $A \gg B$ と $A > B$ は危険率1%と5%で $A$ が $B$ より有意に性能がよく， $A \approx B$ は数値的には $A$ が $B$ よりも良いが，統計的には有意な差は見られないことを意味する．表中のScoutingFWAは(EFWA + 偵察戦略)を，Proposalは(EFWA + 適応偵察戦略)を示す．

	2D	10D	30D
F1	Proposal $\approx$ ScoutingFWA	ScoutingFWA $\gg$ Proposal	ScoutingFWA $\gg$ Proposal
F2	Proposal $\approx$ ScoutingFWA	Proposal $\approx$ ScoutingFWA	ScoutingFWA $\gg$ Proposal
F3	Proposal $\approx$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	ScoutingFWA $\approx$ Proposal
F4	ScoutingFWA $\approx$ Proposal	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F5	Proposal $\gg$ ScoutingFWA	Proposal $\approx$ ScoutingFWA	ScoutingFWA $\gg$ Proposal
F6	Proposal $\gg$ ScoutingFWA	ScoutingFWA $\approx$ Proposal	Proposal $\approx$ ScoutingFWA
F7	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F8	Proposal $\gg$ ScoutingFWA	ScoutingFWA $\approx$ Proposal	ScoutingFWA $\gg$ Proposal
F9	Proposal $>$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F10	Proposal $\gg$ ScoutingFWA	Proposal $\approx$ ScoutingFWA	ScoutingFWA $>$ Proposal
F11	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F12	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F13	Proposal $>$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F14	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F15	Proposal $\gg$ ScoutingFWA	Proposal $>$ ScoutingFWA	Proposal $>$ ScoutingFWA
F16	ScoutingFWA $\gg$ Proposal	ScoutingFWA $>$ Proposal	ScoutingFWA $\gg$ Proposal
F17	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F18	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F19	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F20	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F21	Proposal $\gg$ ScoutingFWA	ScoutingFWA $>$ Proposal	Proposal $\approx$ ScoutingFWA
F22	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F23	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F24	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F25	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F26	Proposal $>$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F27	Proposal $\gg$ ScoutingFWA	Proposal $>$ ScoutingFWA	Proposal $\gg$ ScoutingFWA
F28	Proposal $>$ ScoutingFWA	Proposal $\gg$ ScoutingFWA	Proposal $\gg$ ScoutingFWA

第1の改良は，効果的に収束へ導きランダム探索を減らすためにいかにそれまで収集した探索情報を使うかに注力した改良である．一旦親個体よりも劣る子スパーク個体があったとしても子個体探索の炸裂の中心を少し過去に戻してやり直すことで，有望な探索領域に留まったままfitness計算を大きく増やすことなく良い探索方向を探し直すようにした．しかし長所短所は裏腹で，この改良方法では，局所領域に捕まり脱出が困難になるリスクも抱える．

第2の改良はこの状況打破のための補完方法である．核となる考えは，子スパーク個体の探索範囲を決める炸裂の大きさを適応的に変え，局所探索と大域的探索のバランスを調整することで現在の最適化状況に合わせようとするものである．生成した子スパーク個体の方が劣っている場合には炸裂の大きさを大きくして現状の領域からの脱出を図り，そうでない場合は炸裂範囲を小さくして収束先の精度向上に重点を置く．

提案の適応偵察戦略は，これら二つの改良を協力させることで，探索の効果を維持しながら局所最適領域からの脱出も図る．さらに，この提案では新たなfitness評価を増やすわけではないので，性能向上のための操作のCPUタイムはほとんど無視できる．提案法は，シンプルでかつ効果的であると言える．

第2に，新戦略の拡張性について述べる．今回の実験のベースラインアルゴリズムとして用いたEFWAに限らず，どのような花火アルゴリズムの改良版とも組み合わせが可能で，単に炸裂演算を提案戦略に置き換えるだけで使うことができる．提案の二つの改良は独立しているので，いずれかの改良法を花火アルゴリズムに組み込むことも可能である．更には，他の進化計算手法との組み合わせの可能性もある．例えば，PSOの個体更新時に適応偵察探索の考えを利用できるかもしれない．

まだ確認しているわけではないが，提案の更なる改良ができるのではないかとこの憶測を述べたい．劣った子個体が生成された時に，どのように探索の炸裂範囲の中心を決めるか，という点は議論に値する．本論文では，一つ前の親スパーク個体を探索のための炸裂範囲の中心にするとした．が，その他の妥当な探索範囲の決定方法でもよい．例えば，数個前の親スパーク個体まで戻ってもよいし，過去探索履歴からの重み付けで探索方向を修正をしてもよい．こうすることで，有望な探索領域を維持しつつ個体の多様性を増やすことができる可能性がある．

他の議論点として，長い間進化しない個体がある場合に局所領域からいかに脱出するか，という問題がある．万能な方法があるわけではないが，できる限り探索を効率的になるよう改良し，無駄な計算コストを掛けないようにする必要がある．例えば，局所領域から脱出するための新しい突然変異を導入し，進化しない個体の増加に応じて突然変異の確率を上げることも一つの手である．そのような他の研究成果や考えを我々の提案に組み込みことも可能である．今回いくつかの話題と可能な解決策について述べたが，まだまだ多くの議論すべき点もあり，多くの研究者に示唆を与えられたらと期待する．

Wilcoxonの符号検定を適用し新提案の適応偵察戦略とオリジナルの偵察戦略の有意差を調べた．検定の結果，低次元でも高次元の複雑な問

題でも適応偵察戦略にすることで花火アルゴリズムの収束を一層加速できることを示した。

しかし、単峰性の問題では期待した程の性能差はなく、一部では次元数が上がるにつれて改善効果が弱くなったり逆に悪くなったりすることすらあった。これは、単峰性の問題は複雑な特性を持たず、すぐ最適解に行き着いてしまうからかもしれないし、逆に、ベスト調整をしても収束速度を低下させてしまうかもしれない。このことから、異なる特性の問題毎に適切な探索戦略を取る必要性を示唆される。提案法はどの次元の $F_{16}$ に対しても効果がなかった。多くの似た局所最適解が密にあるため、色々な局所領域を容易に飛び越え、精度向上は難しかったからではないかと推察する。もちろん、実際のところは今後の分析を待つ必要があり、その結果、より妥当な改善版を提案できるようになることを期待している。

## 6 結論

花火アルゴリズムの性能を更に改良するために、以前提案した偵察戦略に二つの改良を加えた適応偵察戦略を提案した。第1改良点は劣った子スパーク個体が生成された時に最初の花火個体に戻って一から探索をするのではなく、過去探索中の良い親スパーク個体まで戻って探索をし直すことであり、第2改良点は現在の探索結果に基づいて子スパーク個体探索の範囲を適応的に変えることである。実験結果から新提案戦略の有効性が確認でき、特に複雑な多峰性問題で有効であることを明らかにした。

今後も花火アルゴリズム探索の収束速度のために情報利用について改良を重ね、異なる特性の問題も高速に最適化できるインテリジェントな適応戦略を提案していきたい。

## 謝辞

本研究はJSPS科学研究費（課題番号 18K11470）の助成を受けたものである。

## 参考文献

- 1) N. Bacanin and M. Tuba, “Fireworks algorithm applied to constrained portfolio optimization problem,” IEEE Congress on Evolutionary Computation, pp. 1242–1249 (2015).
- 2) J. H. Holland, “Genetic algorithms,” Scientific American, vol. 267, no. 1, pp. 66–72 (1992).
- 3) J. Liang, B. Qu, P. N. Suganthan, and G. H. Alfredo, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization.” [http://al-roomi.org/multimedia/CEC\\_Database/CEC2013/RealParameterOptimization/CEC2013\\_RealParameterOptimization\\_TechnicalReport.pdf](http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf) (2013).
- 4) L. Liu, S. Zheng, and Y. Tan, “S-metric based multi-objective fireworks algorithm,” IEEE Congress on Evolutionary Computation, pp. 1257–1264 (2015)..
- 5) J. Kennedy and R. Eberhart, “Particle swarm optimization,” IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (2015).
- 6) R. Storn and K. Price, “Differential evolution? a simple and efficient heuristic for global optimization over continuous spaces,” Journal of global optimization, vol. 11, no. 4, pp. 341–359 (1997).
- 7) Y. Tan and Y. Zhu, “Fireworks algorithm for optimization,” The First International Conference on Swarm Intelligence, pp. 355–364 (2010).
- 8) J. Yu and H. Takagi, “Acceleration for fireworks algorithm based on amplitude reduction strategy and local optima-based selection strategy,” 8th International Conference on Swarm Intelligence, pp. 477–484 (2017).
- 9) J. Yu, H. Takagi, and Y. Tan, “Multi-layer explosion based fireworks algorithm,” International Journal of Swarm Intelligence and Evolutionary Computation, vol. 7, no. 3, pp. 1–9 (2018).
- 10) J. Yu, H. Takagi, and Y. Tan, “Fireworks algorithm for multimodal optimization using a distance-based exclusive strategy,” IEEE Congress on Evolutionary Computation, pp. 2215–2220 (2019).
- 11) J. Yu, Y. Tan, and H. Takagi, “Scouting strategy for biasing fireworks algorithm search to promising directions,” The Genetic and Evolutionary Computation Conference Companion, pp. 99–100. (2018).
- 12) S. Zheng, A. Janecek, and Y. Tan, “Enhanced fireworks algorithm,” IEEE Congress on Evolutionary Computation, pp. 2069–2077 (2013).