局所個体群のfitnessに基づくfitness景観の勾配推定を導入した花火アルゴリズム

余, 俊
九州大学大学院芸術工学府

高木, 英行
九州大学大学院芸術工学研究院

出版情報：進化計算シンポジウム. 2016, pp.131-137, 2016-12-10. 進化計算学会
バージョン：
権利関係：

局所個体群のfitnessに基づく fitness景観の勾配推定を導入した花火アルゴリズム

余俊[†]，高木英行[††]

九州大学大学院芸術工学府[†], 九州大学大学院芸術工学研究院[††]

1 はじめに

群知能・進化計算は，個体ベースの最適化探索手法で，単純，ロバスト，並列処理，処理能力などの特長から関心と応用が拡がっている．群知能は生物進化に由来する演算に拘ることなく，複雑な個体群の振る舞いを単純な個体間の相互協力で表現するよう模擬するものであり，particle swarm optimization [1]，artificial bee colony ant colony optimization [2] など多くのアルゴリズムが広く使われている．

2010年に提案された花火アルゴリズム (Fireworks Algorithm: FA) [3] もそのうちの一つである．花火が夜空に打ち上げられると炸裂点の周りにスパークが飛び散る．花火アルゴリズムは，この大小の複数の花火が夜空に広がる様子を，夜空（探索空間）に複数の花火炸裂点（探索点）があり，各炸裂点の周りにスパーク（局所探索点）を散らしながら徐々に大局的最適解に集中していく，と模擬したアルゴリズムである[3]．最近は様々な改良型花火アルゴリズムが提案されている．例えば，Enhanced Fireworks Algorithm (EFA) [4] はオリジナルの花火アルゴリズムの演算を改良し性能向上を図ったものであり，Two-stage Explosions Fireworks Algorithm (TsEFA) [7] は広域探索と局所探索のバランスを強調した改良である．

しかし，これらの改良アルゴリズムも含めてこれまでの花火アルゴリズムは，生成された多くの花火のスパーク（探索点）利用に焦点を当てておらず，次世代探索点決定以降はもう使われない．

本論文では，花火の炸裂点fitnessとスパークfitnessの差を考慮した，局所合成スパークの概念を導入する．さらに，オリジナルの花火アルゴリズムで使われている距離ベースの選択ではなく，個体群の多様性維持のために局所最適選択戦略を採用する．

以降では，第2節で花火アルゴリズムも枠組みを概観し，第3節で提案法を述べる．第4節では，3種類の次元数×20個のベンチマーク関数で提案手法を評価してオリジナルの花火アルゴリズムと比較する．最後に第5節で考察し第6節で結論を述べる．

2 花火アルゴリズム

本論文では花火の炸裂点とスパークという用語を多用する．炸裂点は前世代からコピーされた有力な子個体で探索空間に広く散らばり局所探索領域の中心になり，スパークはその炸裂点周辺に散らばるという違いがある．両者ともfitnessを求める探索点である．Fig. 1にこの様子を示す．炸裂点のfitness値で花火の善し悪し（fitnessが良い／悪いと期待される局所探索空間）を言うとすれば，良い花火ほど炸裂点近傍に多くのスパークを生成して近傍探索能力を上げ，悪い花火ほど少ない生成スパーク点を広い範囲内に生成して広域探索に重点を移す．

Algorithm 1に，主に炸裂，突然変異，選択の3つの演算からなる花火アルゴリズムの計算フローを示す．次に，各演算を簡単に説明する．

初めに記号の定義をする．xは探索空間上の探索点で，$x_0^{(i)}$をi番目花火の炸裂点，$x_j^{(i)}$はi番目花火の炸裂点周辺に散らばるj番目のスパークとする．共に探索点である．$f(x)$はfitness関数，nは各世代での花火炸裂点数，F_{max}とF_{min}は当該世代での全花火炸裂点のfitnessの最良fitness値と最悪fitness値，ξは0で割り算することを避けるためのコンピュータで表現できる最小定数，である．

Fireworks Algorithm with local fitness gradient estimated from fitness values of local individuals

† Jun Yu (yujun@kyudai.jp)

†† Hideyuki Takagi (http://www.design.kyushu-u.ac.jp/~takagi/)

Graduate School of Design, Kyushu University (†)

Faculty of Design, Kyushu University (†)

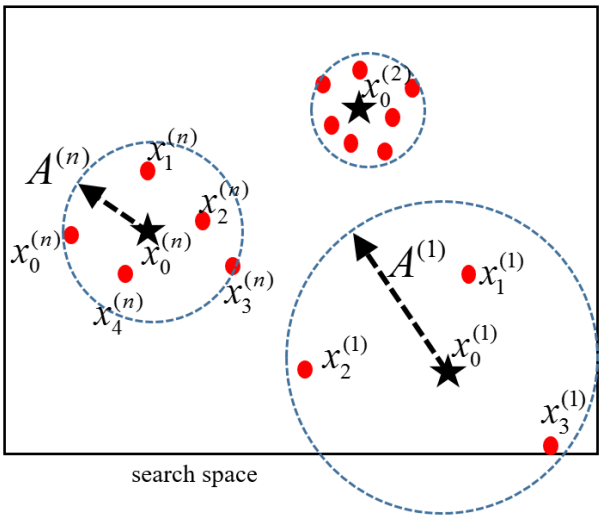


Fig. 1　花火アルゴリズムのよる探索．★（$x_0^{(i)}$）は花火炸裂点，●（$x_j^{(i)}$）は炸裂点の周りに散らばるスパークで，両者とも探索点．アルゴリズムでこれらの位置や花火の広がり（$A^{(i)}$）を決定する．

スパーク数

次式で，i番目花火の炸裂点周辺に生成するスパーク数を決める．

$$s_i = m * \frac{F_{max} - f(x_0^{(i)}) + \xi}{\sum_{k=1}^{n}\left\{F_{max} - f(x_0^{(k)})\right\} + \xi}$$

$$s_i^{'} = \begin{cases} \mathrm{round}(a * m) & if \quad s_i < a * m \\ \mathrm{round}(b * m) & else\ if \quad b * m < s_i \\ \mathrm{round}(s_i) & others \end{cases}$$

ここで，mは総数n個の花火炸裂点から生成するスパーク数を調整係数，aとbは定数パラメータ$(0 < a < b < 1)$，round()は丸め込み関数，$s_i^{'}$は第i番目の花火のスパーク数である．スパーク数が多過ぎたり少な過ぎたりしないよう，各花火炸裂点周辺に生成するスパーク数を$_i$から$s_i^{'}$に制限する（Fig. 2 ）．

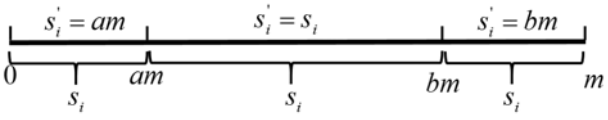


Fig. 2　スパーク数が多過ぎたり少な過ぎたりしないよう，スパーク数を$_i$から$s_i^{'}$に制限する．

Algorithm 1 花火アルゴリズムの枠組み

```
Initialization of n fireworks randomly.
Evaluate the fitness of each firework.
while termination condition is not satisfied do
    Generate explosion sparks for each firework.
    Use Gaussian mutation to obtain Gauss sparks.
    if sparks are generated outside search area then
        use a mapping rule for bringing back to the area.
    end if
    Evaluate the fitness of each spark.
    Select fireworks for the next generation.
end while
end of program.
```

花火の開花サイズ

次式で各花火の大きさを計算する．

$$A^{(i)} = A_{max} * \frac{f(x_0^{(i)}) - F_{min} + \xi}{\sum_{i=1}^{n}\left\{f(x_0^{(i)}) - F_{min}\right\} + \xi}$$

ここで，A_{max}は最大炸裂サイズを示す定数，$A^{(i)}$は第i番目の花火の炸裂サイズである．

スパークの生成

乱数Δを生成する（$-A_i \le \Delta \le A_i$）．次に$z = \mathrm{round}(d \times \mathrm{rand}(0,1))$で$z$値をランダムに決める（$d$は探索次元数，$\mathrm{rand}(0,1)$は一様乱数）．$d$次元の花火座標の中からランダムに選んだ$z$個の座標要素値にΔを加えた座標をスパーク座標とする．すなわち，d次元花火座標中のz座標のみに同じ変位量Δ加えた座標が生成スパーク座標になる．

Gaussian突然変異

スパークの多様性維持のためGaussian突然変異を導入する．

選択戦略

次世代の花火炸裂点の一つは，現世代の全個体（全花火炸裂点＋全スパーク）の中の最良個体を用いる．残りの$(n-1)$個の花火炸裂点は，個体間距離に応じた確率で選択した$(n-1)$個の個体を使う．具体的には，個体数popの中のある個体kから残りすべての個体への累積距離をd_kとすると，個体kの選択確率を正規化距離 $d_k / \sum_{l=1}^{pop} d_l$ で

与える．この選択確率で選ばれた$(n-1)$個の個体と先の最良個体が次世代の花火炸裂点になる．

3　提案の改良花火アルゴリズム

花火アルゴリズムでは，花火炸裂点のfitnessに応じて，花火開花の大きさ（花火炸裂点周辺の局所探索領域の大きさ）を決め，その探索領中に割り当てられた数のスパークをランダムに生成する（Fig. 1）．オリジナルの花火アルゴリズムでは，次世代に生き残る個体は少数のスパークのみで，大多数のスパークは次世代探索に使われずに消滅する．

第１の改善点は，少しの計算コスト増で局所領域に生成されたスパークをフル活用する局所合成スパークの導入である．良し悪しに関わらず花火炸裂点の周りにはスパークがあるので，花火炸裂点からスパーク点に向かう多くのベクトルが得られる．もし生成スパークfitnessが花火炸裂点fitnessより良ければこのベクトルの方向は探索を進める方向を向いていると言えるし，そうでなければ探索を進める逆方向を向いていると言える．またこのベクトルの始点と終点のfitness差は勾配の大きさを示していると言える．

そこで，この花火破裂点からその周辺の各スパークへのベクトルを考える．この始点（花火転破裂点）と終点（スパーク）のfitness差でこれらのベクトル長を正規化した合成ベクトルを求め，その始点を花火破裂点とする（Fig. 3，および，次式）．

$$v^{(i)} = \sum_{i=1}^{n} \frac{f(x_0^{(i)}) - f(x_j^{(i)})}{\sum_{j=1}^{s_i'} \left\{ f(x_0^{(i)}) - f(x_j^{(i)}) \right\}} * \left\{ x_j^{(i)} - x_0^{(i)} \right\} + x_0^{(i)}$$

ここで，$v^{(i)}$は第i番目の花火グループの合成スパーク，$x_0^{(i)}$は第i番目花火の炸裂点，$x_j^{(i)}$は第i番目花火のj番目スパークである．この全スパーク情報から得た合成ベクトルを次世代解候補に加える新たな方式を第1の改善として提案する．

第2の改良点は，次世代の花火炸裂点の選択戦略を，前節で述べた個体間距離に基づく選択確率で選択する方法（オリジナルの花火アルゴリズムの方式）から，局所領域での最適個体選択方式（提案方式）に変えるものである．個体間距離に基づく選択戦略は離れた位置の個体が選ばれて次世代の花火炸裂点になるため，exploitationよりもexplorationを優先しがちな方式と言える．この結果，収束に時間がかかりがちになる．

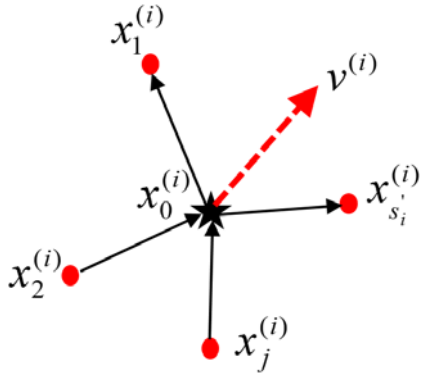


Fig. 3　花火アルゴリズムの第1の改良提案：複数スパークによる合成ベクトル．

各花火の探索範囲には，花火炸裂点，その周辺に生成されたスパーク群，および，第1改良提案の合成スパークが含まれる．これをまとめて局所サブ個体群と呼ぶとすれば，n個の所サブ個体群がある．その他に，Gaussian突然変異で生成された第$(n+1)$番目のサブ個体群もある．次世代にはn個の花火炸裂点が必要なので，最悪花火の局所サブ個体群とGaussian突然変異サブ個体群をまとめて一つのサブ個体群とすることで，n個になったサブ個体群から各々の最良個体を選択して次世代花火炸裂点とする（Fig. 4）．この第2の改良案でも結果的には全個体中の最良個体が選ばれることになるので，オリジナル花火アルゴリズムの選択戦略である全個体中の最良個体を次世代花火炸裂点の一つにするという考えは，包含されることに注意されたい．

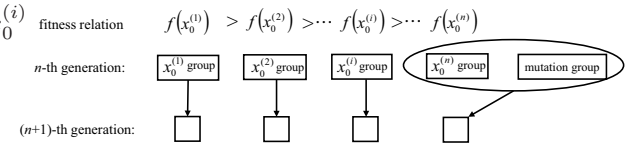


Fig. 4　花火アルゴリズムの第2の改良提案：各花火の探索範囲の最良個体を次世代花火炸裂点とする方式．最悪花火の場合は，探索範囲の全個体とGaussian突然変異で得られた全個体を合わせた中の最良個体を次世代花火炸裂点候補とする．

4　評価実験

CEC2013ベンチマーク関数群[8]から20関数を選び，2次元，10次元，30次元（$D = 2, 10, 30$）で提案手法とオリジナル花火アルゴリズムの収束性能比較に用いる．これらの関数の特性をTable 1に

示す．これらfitness景観特性には，シフト，回転，対称，単峰，多峰が含まれる． 実験パラメータはTable 2に示す．本論文での提案は新しい個体（合成スパーク）の追加と選択戦略の改良であり基本的に同じ花火アルゴリズムなので，両アルゴリズムのこれらパラメータは同じである．

Table 1　Benchmar関数: Uni=単峰性, Multi=多峰性.

No.	Types	Characteristics	Ranges	Optimum fitness
F_1		Sphere Function		−1400
F_2		Rotated High Conditioned Elliptic Function		−1300
F_3	Uni	Rotated Bent Cigar Function	$[-100, 100]$	−1200
F_4		Rotated Discus Function		−1100
F_5		Different Powers Function		−1000
F_6		Rotated Rosenbrock's Function		−900
F_7		Rotated Schaffers Function		−800
F_8		Rotated Ackley's Function		−700
F_9		Rotated Weierstrass Function		−600
F_{10}		Rotated Griewank's Function		−500
F_{11}		Rastrigin's Function		−400
F_{12}	Multi	Rotated Rastrigin's Function	$[-100, 100]$	−300
F_{13}		Non-Continuous Rotated Rastrigin's Function		−200
F_{14}		Schwefel's Function		−100
F_{15}		Rotated Schwefel's Function		100
F_{16}		Rotated Katsuura Function		200
F_{17}		Lunacek BiRastrigin Function		300
F_{18}		Rotated Lunacek BiRastrigin Function		400
F_{19}		Expanded Griewank's plus Rosenbrock's Function		500
F_{20}		Expanded Scaffer's F_6 Function		600

提案花火アルゴリズムでは計算コストが増えるので，公平な評価のために，世代数ではなくfitness計算回数に対する収束を評価する．実験試行回数は，各次元とも30試行である．Fig. 5に30次元ベンチマーク関数の30試行平均の収束特性を示す．その後，Table 2の最大fitness計算回数時点でのfitness値に有意な差があるかどうかをWilcoxonの符号検定で確認し，Table 3に示す．

Table 2　実験パラメータ条件.

# of fireworks for 2-D, 10-D and 30-D search	5
# of sparks m	50
# of Gauss mutation sparks,	5
constant parameters	$a = 0.04$ $b = 0.8$
Maximum amplitude A_{max}	40
stop condition; MAX_{NFC}, for 2-D, 10-D, and 30-D search	4,000, 40,000, 100,000
dimensions of benchmark functions, D	2, 10, and 30
# of trial runs	30

Table 3　FAとPFAは各々オリジナルの花火アルゴリズムと提案花火アルゴリズムを示す. A≪Bと A<B は各々，危険率1%と5%で有意にBの収束停止条件でのfitness値がAより有意に良い, A=Bは両者有意差なし，を示す.

	2-D	10-D	30-D
F_1	FA ≪ PFA	FA ≪ PFA	FA ≪ PFA
F_2	FA = PFA	FA ≪ PFA	FA ≪ PFA
F_3	FA ≪ PFA	FA ≪ PFA	FA ≪ PFA
F_4	FA < PFA	FA = PFA	FA ≪ PFA
F_5	FA ≪ PFA	FA ≪ PFA	FA ≪ PFA
F_6	FA = PFA	FA = PFA	FA ≪ PFA
F_7	FA ≪ PFA	FA = PFA	FA ≪ PFA
F_8	FA < PFA	FA ≪ PFA	FA = PFA
F_9	FA ≪ PFA	FA = PFA	FA ≪ PFA
F_{10}	FA ≪ PFA	FA ≪ PFA	FA ≪ PFA
F_{11}	FA ≪ PFA	FA ≪ PFA	PFA ≪ FA
F_{12}	FA = PFA	PFA ≪ FA	PFA < FA
F_{13}	FA = PFA	FA = PFA	FA = PFA
F_{14}	FA = PFA	FA = PFA	FA = PFA
F_{15}	FA = PFA	FA = PFA	FA ≪ PFA
F_{16}	FA ≪ PFA	FA ≪ PFA	FA ≪ PFA
F_{17}	FA = PFA	FA ≪ PFA	FA = PFA
F_{18}	FA < PFA	FA ≪ PFA	FA ≪ PFA
F_{19}	FA ≪ PFA	FA ≪ PFA	FA ≪ PFA
F_{20}	FA = PFA	FA = PFA	FA = PFA

統計検定結果から，20関数×3次元=60条件中，実験停止条件のfitness計算回数時点で提案手法がオリジナルアルゴリズムより有意に収束が遅い場合は3ケース，両者に有意差が見られなかった場合が21ケース，残りの26ケースで提案手法が有意に高速になった．提案手法が従来手法より有意な場合は各次元20関数中，12関数，11関数，13関数であり，次元数による性能効果はみられない．提案手法は有意に劣る30次元F_{11}, 10次元F_{12}, 30次元F_{12}はRastrigin関数系で，効果を発揮しない理由はまだ解析中である．

5　考察

本論文では，生成されたスパーク点からfitnessの良い勾配方向に合成スパークと呼ぶ新しい探索点を追加する提案をした．探索終了条件（2-D,

10-D, 30-Dでfitness計算回数が各々4千回，4万回，10万回に達した時点）でのfitness値比較という今回の実験評価では，提案手法が回数的にはオリジナル手法よりも収束が早いが，20関数中の11, 12, 13関数，および，60回数（=20関数関数中×3種の次元数）中の36回数という今回の実験結果は，符号検定的には有意差がみられない．

しかしながら，Fig. 5の収束特性を見て分かるように，提案手法が早く収束している場合が多いことが見て取れるので，複数のfitness計算回数時点での有意差検定，および，大局的最適解の極近傍へ到着するfitness計算回数での検定など，収束特性の全体像を見渡す評価をする必要がある．この全体評価なく上記の一点のみで優劣を議論することは誤った結倫を導く恐れがある．この評価がまだ完全ではない本論文は，この研究の第一報と位置付けている．

今回の評価実験では，同じ花火内の炸裂点とスパーク点のみを使い，距離情報とfitness差を計算して合成スパークを算出した．オリジナル花火アルゴリズムでは，各個体から他のすべての個体への距離を利用しているし，正確に勾配方向を計算するのであれば，直交ベクトルを挿入すべきである[5, 6]．しかし，これらは計算コストが増加するため本実験では上記の簡易方法を選択した．この計算精度と計算コストのtrade off 関係を明らかにして，最良の費用対効果になるよう実験的に明らかにすべきであり，今後の課題とした．

合成スパークは局所探索領域の複数の探索点情報から良い探索方向を推定して作成されるので，当該局所探索領域で最良個体になる可能性が高く，その場合は次世代の局所探索領域の中心になるので，探索性能向上に寄与することが期待できる．さらに，局所領域毎のサブ個体群の最良個体が次世代の探索領域の核（花火炸裂点）になることから，オリジナルアルゴリズムのように離れた個体周辺に探索バイアスがかかることなく，explorationからexploitationへ移行していくことが期待できる．

このように本提案は可能性を秘めているものの，第一報の現時点ではまだまだ改良の余地があると思われる．理論的に期待できる長所を結果に表れるよう，実験結果の解析と評価をさらに進める必要がある．

6 結論

本論文では花火アルゴリズムの性能改善のために二つの改良を提案した．第1の提案は局所探索点全体からよりfitness勾配の良い点に新しい探索点（合成スパーク）を求めて探索候補に加えるものである．第2の提案は，各花火の局所最適解を次世代の花火炸裂点として新しい局所探索をする選択戦略である．

今後は，考察で述べた実験結果の収束特性の全体的な評価，より良い局所勾配方向決定のための効果的な個体情報利用の考案を行い有意な効果を得るとともに，提案手法の他の最適化アルゴリズムへの応用展開を考えていきたい．

参考文献

1) Kennedy, J. and Eberhart, R. C., “Particle swarm optimization,” IEEE Int. Conf. on Neural Networks, vol. 4, pp. 1942–1948 (1995).

2) Dorigo, M., Maniezzo, V., and Colorni, A., “Ant system: optimization by a colony of cooperating agents, ” IEEE Tran. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 26, no. 1, pp. 29–41 (1996).

3) Tan, Y., and Zhu, Y., “Fireworks algorithm for optimization,” Advances in Swarm Intelligence, vol. 6145 of the series Lecture Notes in Computer Science, pp. 355–364 (2010).

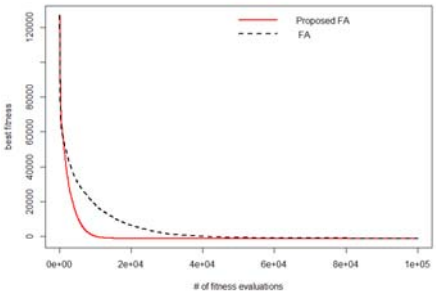
4) Zheng, S.q., Janecek A., and Tan, Y., “Enhanced Fireworks Algorithm” IEEE Int. Conf. on Evolutionary Computation, pp. 2069–2077 (2013).

5) 余俊，高木英行「多峰性最適化問題での局所最適解推定高度化のための補正法 – 局所最適解が2個の場合 –」第9回進化計算研究会，pp.92–97, 神戸 (2015年9月7-8日).

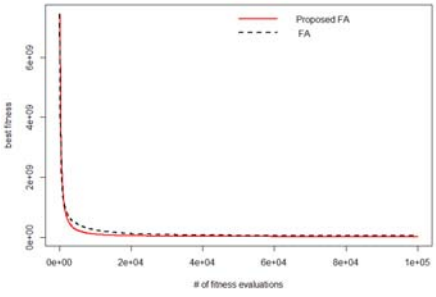
6) Yu,J. and Takagi,Takagi, ”Clustering of Moving Vectors for Evolutionary Computation,” 7th Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR2015), pp.169–174, Fukuoka, Japan (Nov. 13-15, 2015).

7) 余，高木, “八重芯型花火アルゴリズム,” 第10回進化計算研究会, pp.19–23, 川崎 (2016).

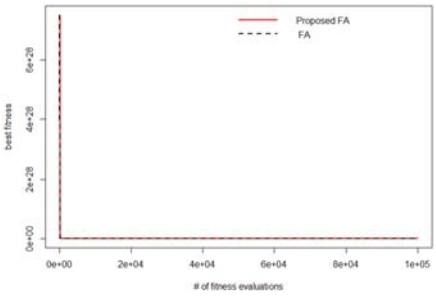
8) Liang, J., Qu, B., Suganthan P. and H., A., G., ”Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” Technical Report (2013). http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf



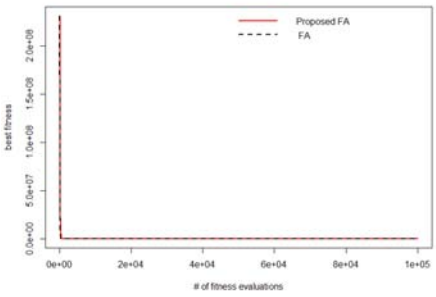
(a) F_1



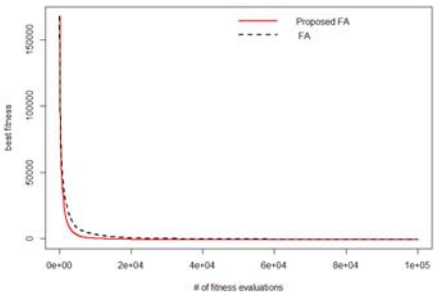
(b) F_2



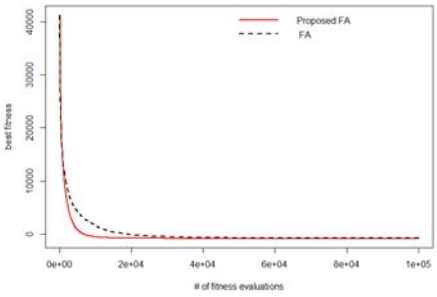
(c) F_3



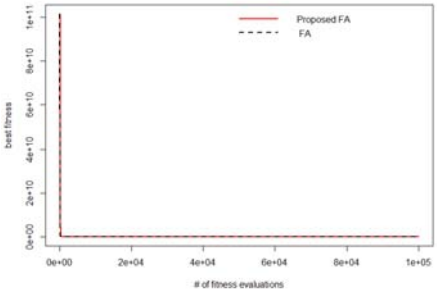
(d) F_4



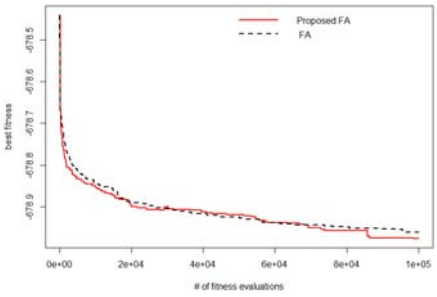
(e) F_5



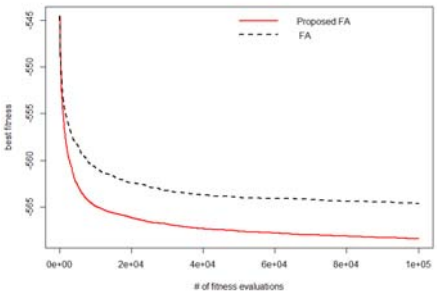
(f) F_6



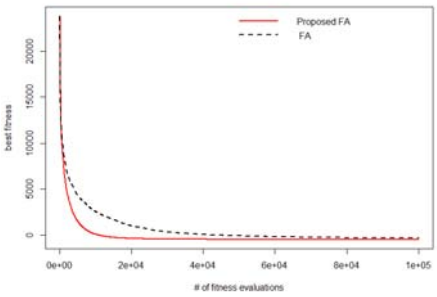
(g) F_7



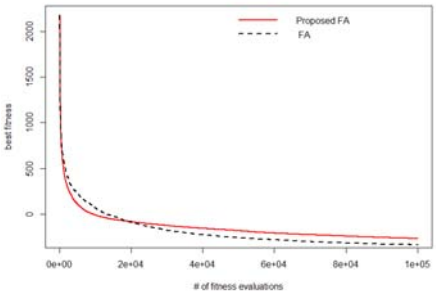
(h) F_8



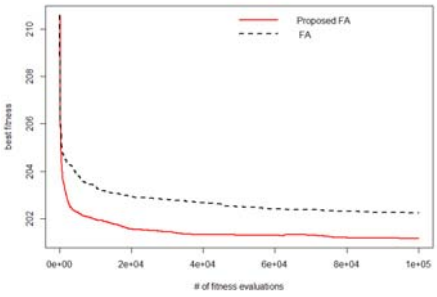
(i) F_9



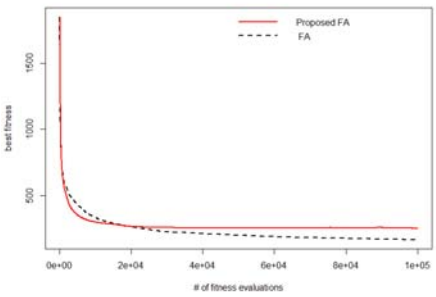
(j) F_{10}



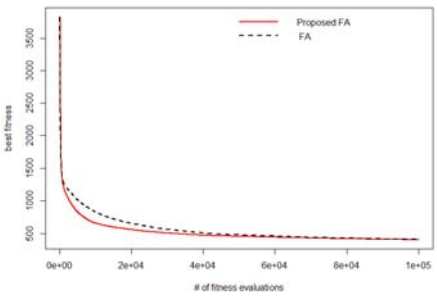
(h) F_{11}



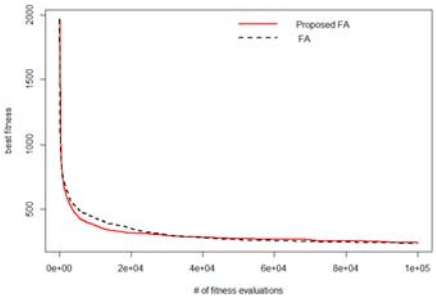
(m) F_{16}



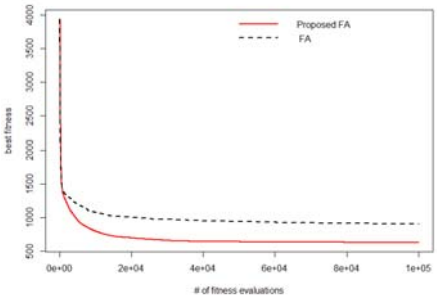
(i) F_{12}



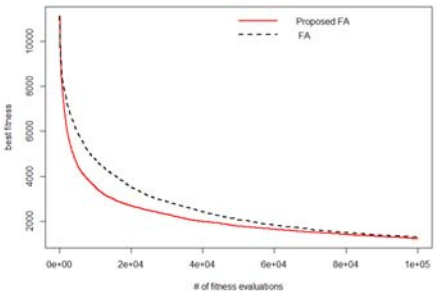
(n) F_{17}



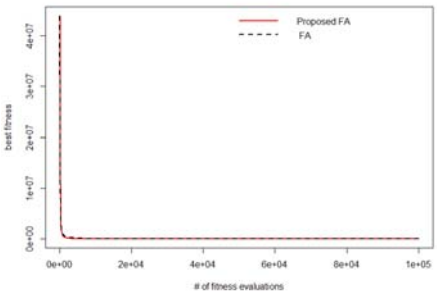
(j) F_{13}



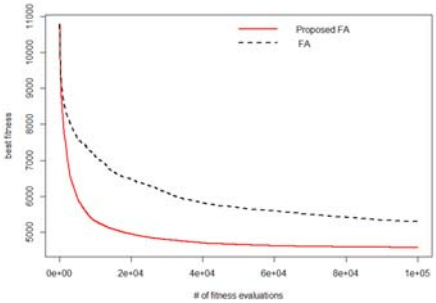
(o) F_{18}



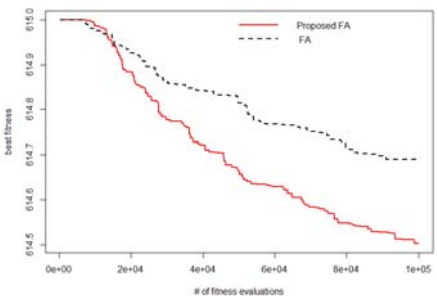
(k) F_{14}



(p) F_{19}



(l) F_{15}



(q) F_{20}

Fig. 5　30次元関数F_1–F_{20}を用いたオリジナル花火アルゴリズム（FA）と提案手法（proposed FA）の30試行平均の収束特性.