

多目的最適化問題のための階層型粒子群最適化

室伏 紀幸¹ 中野 秀洋¹ 宮内 新¹

概要: 本論文では、多目的最適化問題のための分散型粒子群最適化手法を提案する。提案手法では、粒子群を複数のサブ粒子群に分けて二層の階層構造を構築する。下位層のサブ粒子群では一部の目的関数に特化した解の探索を行い、上位層のサブ粒子群ではすべての目的関数に対するパレート最適解の探索を行う。また、いくつかの粒子は一定の間隔で上位層と下位層の間を移住する。提案するアルゴリズムは単純で、計算コストも少ない。いくつかのシミュレーション実験の結果を示す。

1. はじめに

多目的最適化問題では、与えられたすべての目的関数値を最適化する必要がある。しかし、一般的に、各目的関数は互いにトレードオフの関係にあることが多い。このため様々な最適解の候補を実現可能な範囲で得ることが重要となる。具体的には、パレート最適フロントと呼ばれる最適解の集合を得る必要がある。多目的最適化問題の解法として多目的遺伝的アルゴリズムや多目的粒子群最適化 [1]- [4] のような多くの方法が提案されている。得られた解の評価にはパレートランキング法 [2] が使用される。また解の多様性を保つための効果的な方法としてシェアリング処理 [3] があり、これにより多様な解を得ることが可能である。しかし、通常シェアリング処理は高い計算コストを要するという問題点がある。

本論文では、粒子群最適化 (Particle Swarm Optimization: PSO [5]) に基づく多目的最適化問題のための簡素なアルゴリズムを提案する。PSO では対象となる問題空間において粒子が探索を行う。各粒子は速度と位置情報を持ち、探索を行った中で各粒子が発見した最良の位置である pbest と群れ全体における最良の位置である gbest の情報をもつ。PSO は単純な四則演算を使って計算を行うことで、様々な最適化問題を高速に解くことができる。提案するアルゴリズムでは、二層のサブ粒子群に分けた階層構造を導入する [4]。下位層は、目的関数の数に対応する複数のサブ粒子群で構成される。各サブ粒子群は、各目的関数に対応した解を探索する。上位層は、パレート最適フロントの解を見つける一つのサブ粒子群で構成される。各サブ粒子群は、パレートフロントの解を格納し、これらを gbest

として使用するためのアーカイブおよびグリッドを使用する。また、上位層と下位層のサブ粒子群の間で、いくつかの粒子は定期的に移住を行う。提案するアルゴリズムは簡素であり、従来手法よりも計算コストを削減できる。ベンチマーク問題を対象とした数値シミュレーションを行い、提案アルゴリズムの有効性の検証を行う。

2. 従来手法のアルゴリズム

PSO は魚や鳥の群としての行動に着想を得た最適化手法である。各粒子はこれまで探索した自身の最良の位置 (pbest) と群全体で共有する最良の位置 (gbest) 及び前回の移動方向を合わせた 3 つのベクトルによって空間内を探索するように速度と位置ベクトルを更新する。以下に更新式を示す。

$$v_{ij}^{k+1} = wv_{ij}^k + c_1rand_1(pbest_{ij}^k - x_{ij}^k) \quad (1)$$

$$+ c_2rand_2(gbest_j^k - x_{ij}^k)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2)$$

ここで、 x, v は粒子の位置と速度、 i は粒子番号、 j は変数ベクトルの第 j 成分、 w は速度に対する重み、 $rand_1, rand_2$ は 0~1 の範囲の一様乱数である。一般的な多目的 PSO (MOPSO [1]) では、 $gbest^k$ はランダムに $gbest$ のストレージから選択される。ここで、他の解と比べて少なくとも 1 つは劣らない目的関数値を持つ解をパレート最適解と呼ぶ。通常多目的最適化問題ではパレート最適解は集合として与えられる。粒子の適応度はパレートランキングによって与えられる。各粒子のランクは、その粒子に対して全ての目的関数値が優れている粒子の数に対応しており、パレート最適解のランクは 1 となる。より多様なパレート解 (パレート最適フロント) を発見するために、解の評価に混雑度を取り入れることによって、粒子が密集している領

¹ 東京都市大学
Tokyo City University

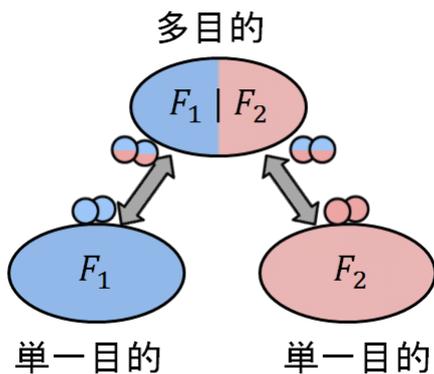


図 1 二層のツリー構造を有する島モデル

Fig. 1 An island model with a two-layered tree structure

域の適応度を低くすることができる。最も一般的な混雑度の計算方法にシェアリング処理がある。適応度 F_i をシェアリング関数によって修正した適応度 F'_i は、以下の式で与えられる。

$$F'_i = \frac{F_i}{\sum_{j=1}^n sh(d(i,j))} \quad (3)$$

ここで、 i と j は粒子番号、 n は粒子の数であり、 $sh(\cdot)$ は粒子の密集度を計るシェアリング関数である。シェアリング関数の式を以下に示す。

$$sh(d(i,j)) = \begin{cases} 1 - \frac{d(i,j)}{\sigma_{share}} & \text{if } d(i,j) < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

ここで、 $d(i,j)$ は粒子 i, j 間のユークリッド距離、 σ_{share} はシェアリング半径のパラメータである。粒子間の距離がシェアリング半径よりも小さい時、評価値は低くなる。しかし、シェアリング処理は全ての粒子間の距離を計算する必要があり、計算コストが高くなることが問題点として挙げられる。

3. 提案手法のアルゴリズム

提案手法では、二層のツリー構造を有する島モデルを使用する。図 1 に示すように、単目的の各島を下位層に配置し、多目的の島は上位層に配置する。下位層の島ではそれぞれ単一の目的関数を評価し、それ以外の目的関数の評価は行わない。上位層の島ではすべての目的関数についてパレートランキングを使って評価を行う。また、いくつかの粒子は一定の間隔でこれらの層の間を移住する。このような階層構造は等価な多目的の島によるネットワーク構造よりも大域的な探索と局所的な探索の両方の面で利点があると考えられる。

さらに、解の多様性を得るために以下の機構を導入する。まず、図 2 に示されるように解空間をグリッドに分割する。各目的関数の最大値と最小値を粒子が存在する範囲から計算し、解空間の i 番目の目的関数 F_i の領域を、 N_i 分

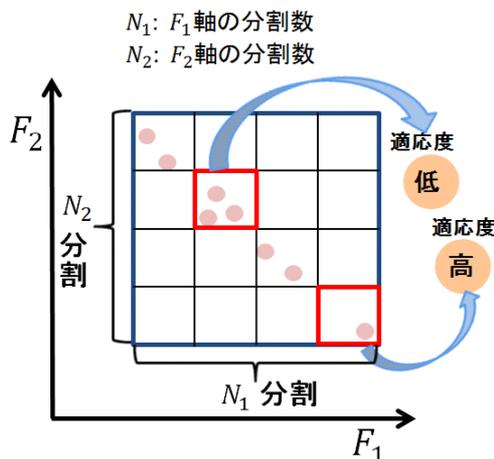


図 2 グリッドとアーカイブの機構

Fig. 2 Grid and archive schemes

割する。探索によって見つかった全てのパレート最適解を $gbest$ として保存すると多くのメモリ容量を必要とするため、保存できる粒子数を事前に決定しておく。新たに $gbest$ の候補として粒子が選択された場合、すでに $gbest$ の集合に格納されている粒子と比較し、パレートランクが 1 の粒子を保存する。もし保存できる許容数を超えた場合は各グリッドで密集度を用いた評価を行い、密集部分の解を消去する。式 (1) における各粒子の $gbest^k$ は一般的な MOPSO と同様に $gbest$ の集合の中からランダムに一つを選んで割り当てる。

4. シミュレーション実験

提案手法と従来手法 (シェアリング処理を用いる一般的な MOPSO) を比較するために、ZDT セット [6] から選択したテスト問題を使用する。各問題の目的関数は、いずれも 2 つである。各手法の性能を測定するために以下の 2 つの測定基準を使用する。

- Generational Distance (GD) [7]

$$GD = \frac{1}{n} \sqrt{\sum_{i=1}^n d_i^2} \quad (5)$$

ここで、 n は解集合に含まれる解の数であり、 d_i は解 i から真のパレート最適解集合のうち一番近い解までのユークリッド距離である。この測定基準によって、見つけられたパレートフロントと真のパレート最適フロントの間の平均距離を評価する。

- Spacing (S) [7]

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2 / \bar{d}} \quad (6)$$

ここで、 $\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$ である。また、 n は解集合に含まれ

表 1 共通の実験パラメータ

Table 1 Common parameters in the simulation experiments.

	提案手法	従来手法
w	0.9	
c_1, c_2	1.0	
島数	3	1
移住間隔	20	-
移住粒子数	1	-
設計変数 (x_i) の数	10	

表 2 テスト問題別の実験パラメータ

Table 2 Case parameters in the simulation experiments.

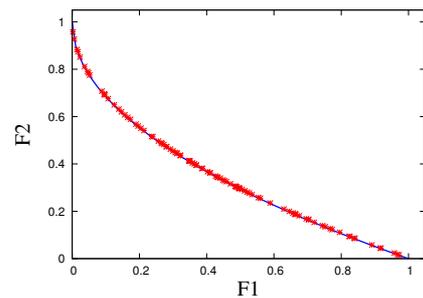
	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
粒子数	102 (3×34)	102 (3×34)	300 (3×100)	501 (3×167)	102 (3×34)
世代数	1×10^4	2×10^3	1×10^4	1×10^5	1×10^4
分割数	$N_1=15,$ $N_2=15$	$N_1=10,$ $N_2=10$	$N_1=10,$ $N_2=18$	$N_1=17,$ $N_2=17$	$N_1=17,$ $N_2=10$
σ_{share}	0.003	0.12	0.002	0.1	0.2

表 3 テスト問題の実験結果

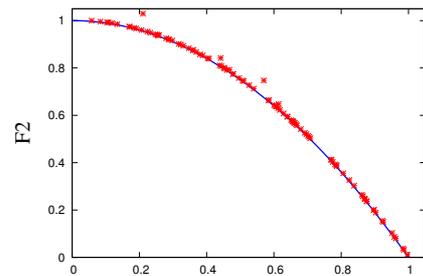
Table 3 Comparison of performances on test problems

	手法	GD	S
ZDT1	従来手法	0.002561	1.305792
	提案手法	0.000198	0.566345
ZDT2	従来手法	0.009106	1.620482
	提案手法	0.005435	1.243906
ZDT3	従来手法	0.014512	2.048453
	提案手法	0.014549	1.413845
ZDT4	従来手法	0.077915	5.912123
	提案手法	0.309881	2.610747
ZDT6	従来手法	0.053571	5.081610
	提案手法	0.002070	0.716897

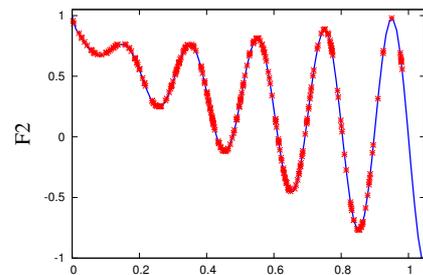
る解の数であり、 d_i は解集合に含まれる解 i と解集合のうち一番近い解までのユークリッド距離である。この測定基準によって、見つけられたパレートフロントがどれだけ均一に広がっているかを評価する。GD と S の数値は、いずれもより小さいほど優れている。実験を行い提案手法と従来手法 (シェアリング処理を用いた一般的な MOPSO) の性能を比較する。実験で使用したパラメータを表 1, 2 に示す。全てのテスト問題において、10 回ずつ試行を行った。各試行のうち最良であった結果を表 3 に示す。また、各手法によって得られた最終世代のパレートフロントを図 3, 図 4 に示す。提案手法はほとんどのテスト問題において従来手法と同等かそれを上回る結果となった。特に設計変数と目的関数の間に偏りが存在する ZDT6 において良い結果を確認することができる。この問題では、従来手法は図 4(e) に示すようにパレート最適解を探索することが困難であったが、提案手法は図 3(e) に示すように広く均一にパレート最適解付近を探索することができた。一方、



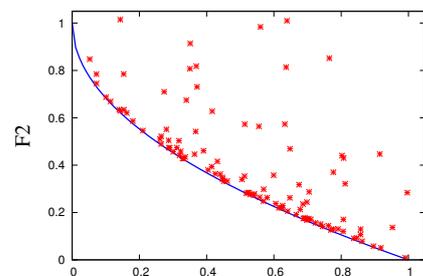
(a) ZDT1



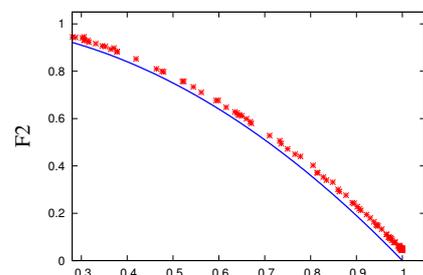
(b) ZDT2



(c) ZDT3



(d) ZDT4



(e) ZDT6

図 3 実験結果 (提案手法)

Fig. 3 Simulation results (proposed method)

ZDT4 では従来手法の方が提案手法よりも GD の値が良好であった。しかしながら、提案手法では図 3(d) に示すようにいくつかの粒子は真のパレート最適フロントに到達している。一方、従来手法では図 4(d) に示すように全ての粒子が真のパレート最適フロントに到達できなかった。つまり、この問題において提案手法は必ずしも従来手法よりも劣っているとはいえない。以上の結果より提案アルゴリズムは単純で計算コストも少なく、多目的問題に対する探索の有効性が確認された。

5. 結論

本論文では、二層のツリー構造を有する島モデル MOPSO を提案した。提案手法を一般的なテスト問題を使って評価した結果、シェアリング処理を用いずにより分散したパレート最適解を発見することができた。すなわち、提案手法は高い計算コスト必要とせずパレート最適フロントの解を効率的に探索できた。

今後の課題としてより幅広いテスト問題に適用していくことが挙げられる。特に目的関数の多い問題に適用する場合、単目的と多目的の島をどのように配置するかを考える必要がある。

参考文献

- [1] M. Reyes-Sierra and C. A. Coello Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *Int. J. Computational Intelligence Research*, vol. 2, no. 3, pp. 287-308 (2006).
- [2] C.M.Fonseca and P.J.Fleming, "Genetic algorithms for multiobjective optimization," *Proc. of the Fifth International Conference on Genetic Algorithms*, pp. 416-423 (1993).
- [3] D.E.Goldberg and J.Richardson, "Genetic algorithms with sharing for multi-modal function optimization," *Proc. of the Second International Conference on Genetic Algorithms*, pp. 41-49 (1987).
- [4] K.Shibata, H.Nakano, and A.Miyauchi, "A learning method for dynamic Bayesian network structures using a multi-objective particle swarm optimizer," *Artificial Life and Robotics*, vol.16, No.3, pp. 329-332 (2011).
- [5] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942-1948 (1995).
- [6] K.Deb, L.Thiele, M.Laumanns, and E.Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," *Evolutionary multi-objective Optimization: Theoretical Advances and Applications*, A.Abraham, R.jain, and R.Goldberg, Eds., pp. 105-145 (2005).
- [7] K.C.Tan, T.H.Lee, Y.J.Yang, and D.S.Liu, "A Cooperative Coevolutionary Algorithm for Multiobjective Optimization," *IEEE international Conference on Systems, Man and Cybernetics*, pp. 1926-1931 (2004).

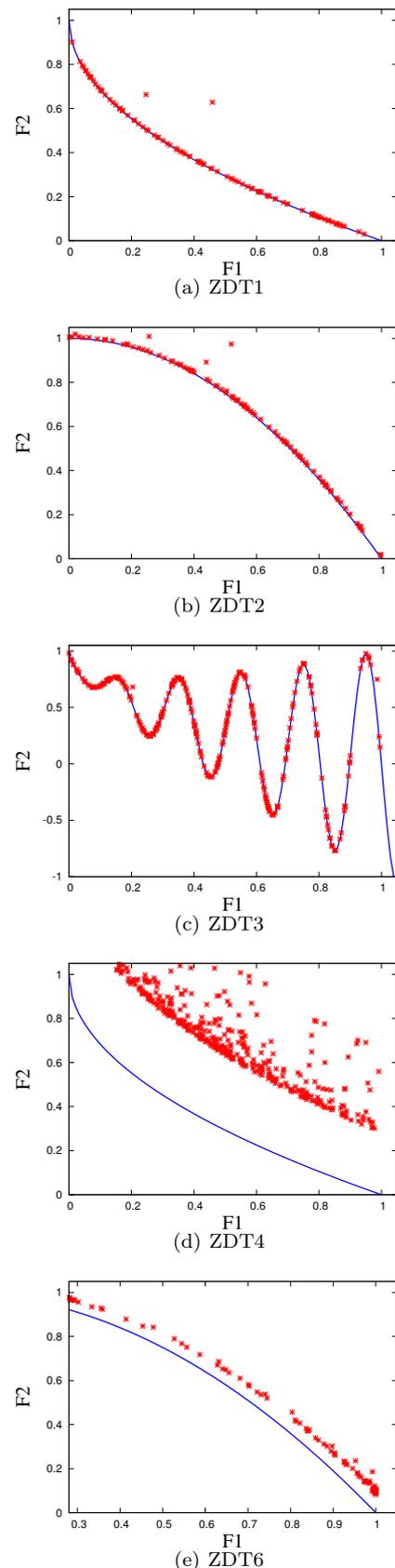


図 4 実験結果 (従来手法)

Fig. 4 Simulation results(conventional method)