

多目的最適化アルゴリズム NSGA-II における指向性を持つ並列分散処理方式 The parallel distributed processing method using objective space partitioning by directivity for the multi-objective optimization algorithm NSGA-II.

小林良輔^{†*}
Ryosuke Kobayashi

佐藤未来子[†]
Mikiko Sato

佐藤 裕二[‡]
Yuji Sato

1. はじめに

近年では、新幹線の先端形状やジェット旅客機の翼といった形状設計で多目的最適化アルゴリズムを適用する事例がある。このような事例では多数の制約や複雑な目的関数を持つ問題が多く、有用な解を求めるためには解候補となる個体を多く用いて探索する必要があり、最適化に膨大な時間がかかる。多目的最適化アルゴリズムの解探索精度向上と高速化を実現することは、実問題を扱う現場において設計開発時間の短縮や、効率向上による良解の探索や計算資源の効率的な利用によるコスト削減につながる。本稿では多目的最適化問題の解探索精度向上のための指向性探索手法について述べる。

2. 関連研究

2.1 NSGA-II

Deb らが提案する NSGA-II[1]は、単一目的最適化アルゴリズムである遺伝的アルゴリズムを多目的最適化問題向けに拡張したアルゴリズムである。図 1 に NSGA-II の概要を示す。NSGA-II は、高速非優劣ソートによるランキングや混雑距離による評価、混雑度トーナメントによる選択といった特徴を持つ。

2.2 分散 NSGA-II

佐藤らが提案している分散 NSGA-II[2]は、NSGA-II の探索高速化および解探索精度の向上を実現するための分散システム向け並列化方式である。複数のコア上で NSGA-II を並列実行させ、各コアで求めた全ての最適解集合を集め、再評価を行うマイグレーション処理を行い最終的な最適解集合を求める。解を単に分散して並列化する方式では、総個体集団を各コアに分割することにより、各コアで扱う個体数が減り、従来の単一 CPU で全個体を最適化する時よりも多様性が低下するという課題がある。この問題に対して、分散 NSGA-II では、解探索途中に各コアの非劣解集合を集め、再度非優越ソートを行い、rank1 の解集合を各コアに対し再分配する。これと同時に、個体間距離が最大である rank1 の両端の解個体を全てのコア間で共有することでパレート最適解の解探索精度の向上を図っている。

しかし、分散 NSGA-II では、各サブグループが解探索領域全体を探索する。そのため、より並列度を上げた場合に速度向上は図れるが、各サブグループの個体数が減少し、収束性が弱まるという傾向にある。したがって、並列度を高めて解探索精度を向上させるためには、総個体数を増やす必要があり、並列度に応じたメモリリソース量を要する

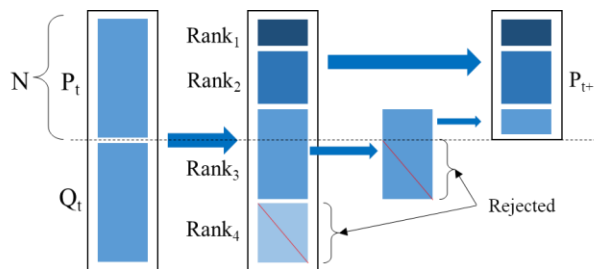


図 1 NSGA-II のアルゴリズムの概念図

こととなる。総個体数を増やさずに可能な限り解探索性能を向上させることが分散 NSGA-II の課題となっている。

3. 指向性をもたせる分散処理方式の提案

分散 NSGA-II[2]における、並列度を上げる場合の探索性能の向上に関する課題を解決するために、本研究では分散 NSGA-II の各サブグループの探索に指向性を持たせる方式を提案する。図 2 に提案方式の概要を示す。提案方式では、分散 NSGA-II での個体のマイグレーション処理の際に、目的空間上で距離が近い個体同士が同一サブグループとなるように全個体を分割する。これにより、各サブグループが、解探索領域の一部を集中的に探索する傾向を強める。各サブグループが指向性に即した探索を行うことにより、収束性を高める効果が期待できる。

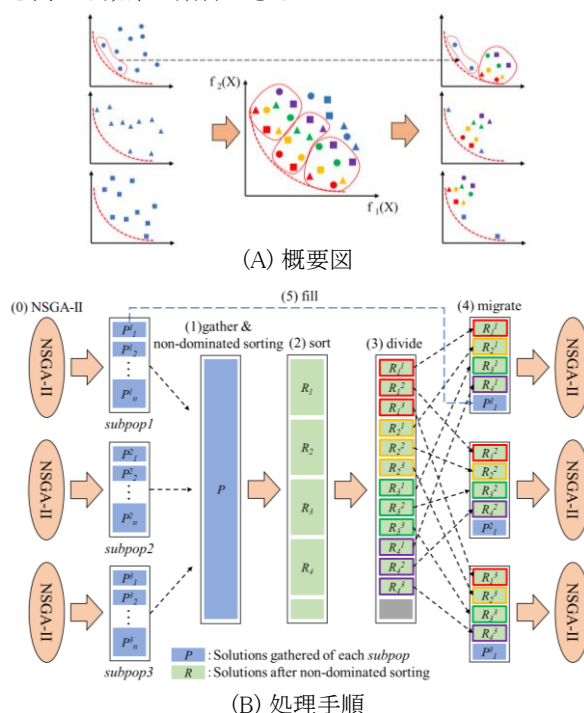


図 2 指向性を持つ分散 NSGA-II の概念図

[†] 東海大学 Tokai University

[‡] 法政大学 Hosei University

*現在、富士通株式会社に所属

4. 指向性をもたせる分散処理方式の実装と評価

4.1 実装

本提案方式の実装には、多目的進化アルゴリズムを評価するためのフレームワークの一つである pymoo[3]を活用した。pymoo は、NSGA-II などの最適化アルゴリズムと解析ツールを有している。本研究では pymoo に実装されている NSGA-II[1]を利用して、分散 NSGA-II[2]および指向性を持たせる分散処理方式の両方を実装した。

4.2 評価実験

本実験では、単一 CPU で実行する従来方式 (NSGA-II)、マイグレーションの工夫をせず単純に複数の NSGA-II を並列実行して解を求める方式 (Simple parallel NSGA-II)、2.2 節で述べた分散 NSGA-II (Distributed NSGA-II)、3 節で述べた提案方式 (Proposed method) において、pymoo で提供されているベンチマーク問題を用いて、一定の時間で探索した解探索精度の推移を比較し、提案方式の有効性を評価する。本実験では総個体数 40 とし試行回数 31 回の平均値を算出する。単純並列化方式、分散 NSGA-II、提案方式の並列度を 4 とし、単一 CPU で実行する NSGA-II と比較する。解探索精度の評価尺度には Hypervolume[4]を用いる。

図 3(A)に、最適化に時間がかかる実問題系ベンチマーク問題である MODAct の CS1 の結果を、図 3(B)(C)に、制約数の多い標準ベンチマーク問題である MW5 および DASCOP1 の結果を示す。CS1 では、初期の解探索時点ではどの並列化方式も単一 CPU の NSGA-II より探索精度が向上し、中盤以降では指向性を持つ提案方式の効果が最も高かった。一方、MW5 および DASCOP1 では、精度向上のための工夫をした提案方式や分散 NSGA-II が、NSGA-II や単純並列化方式よりも高い解探索精度を示した。

指向性方式の効果について考察する。図 4 に MODAct の CS1 において、顕著に差が生じた探索時間 10 秒と 30 秒時の NSGA-II と本提案方式の解分布を示す。CS1 は Pareto Front で示すように目的のパレート解集合には 30 秒でも到達できない問題である。本提案方式は NSGA-II と比較して 10 秒付近から多様性があり、30 秒付近でも明らかに多様性が増している。CS1 のように探索に時間を要する実問題系の問題に対しては、指向性を考慮しながら領域ごとに集中的に並列探索する本提案が有効であったと考える。一方、MW5 や DASCOP1 は図 3(B)(C)の結果に示すとおり、少ない個体数 40 であっても、単一 CPU で実行する NSGA-II の結果が各種並列化方式と同等の解探索精度を示している。すなわち、並列化で高速化せず短時間で解探索できる、比較的簡易なベンチマーク問題だといえる。並列化方式で

は、分散 NSGA-II (Distributed NSGA-II) が高い値を示している。分散 NSGA-II は各コアが目的空間全体を広く探索する方式であり、パレートフロントの特徴に応じて並列化方式を使い分けることが重要であるという知見を示している。

本提案方式は、各サブグループが解探索領域の一部を集中的に探索することを特徴としており、解探索に時間のかかる実問題系ベンチマーク問題において特にこの方式の効果が現れることを示した。

5. おわりに

本研究では、先行研究である分散 NSGA-II のさらなる精度向上のために、指向性を考慮した解探索法を提案した。Python の多目的最適化ライブラリ pymoo へ構築して評価し解探索に時間のかかる実問題系ベンチマーク問題に有効であることを示した。今後は、さらに並列度を高めた場合の実験や、マイグレーションのタイミングで分割対象とするランク数を変化させた場合の評価、最適化問題における実行可能領域に注目した方式などを追加実験し、本提案方式の有効性を確かめる。

謝辞

本研究は文部科学省科学研究費補助金 (課題番号 JP19K12162, JP22K12185) の助成により行われた。

参考文献

- [1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197 (2002).
- [2] Y. Sato, M. Sato and M. Miyakawa, "Distributed NSGA-II Sharing Extreme Non-dominated Solutions for Improving Accuracy and Achieving Speed-up," 2019 IEEE Congress on Evolutionary Computation (CEC), pp. 3086-3093 (2019).
- [3] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," in IEEE Access, vol. 8, pp. 89497-89509 (2020).
- [4] N. Beume, C. M. Fonseca, M. Lopez-Ibanez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 1075-1082 (2009).

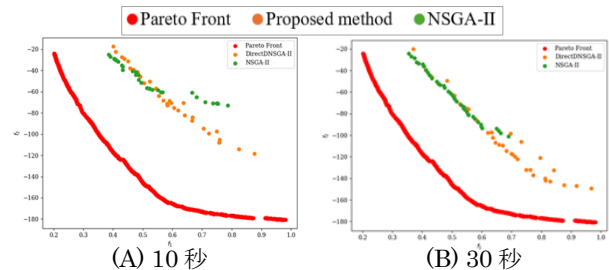


図 4 解探索精度の時間推移の結果

(Proposed method の解集合の表示には劣解も含まれている)

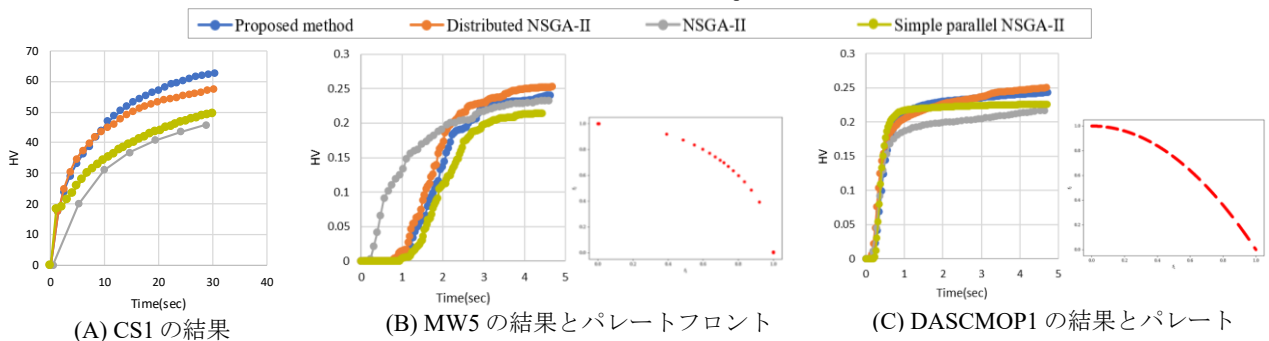


図 3 解探索精度の時間推移の結果