

進化型多目的最適化における探索履歴を活用した局所解脱出と集中探索メカニズム

Escaping from Local Optima and Convergence Mechanisms Based on Search History in Evolutionary Multi-criterion Optimization

左文字 響
Hibiki Samonji

室蘭工業大学大学院 情報電子工学系専攻
Division of Information and Electronic Engineering, Muroran Institute of Technology
15043026 @ mmm.muroran-it.ac.jp

渡邊 真也
Shinya Watanabe

室蘭工業大学大学院 しくみ情報系領域
College of Information and Systems, Muroran Institute of Technology
sin@csse.muroran-it.ac.jp

keywords: evolutionary multi-criterion optimization, local search, search history

Summary

In this paper, a new local search approach using a search history in evolutionary multi-criterion optimization (EMO) is proposed. This approach was designed by two opposite mechanisms (escaping from local optima and convergence search) and assumed to incorporate these into an usual EMO algorithm for strengthening its search ability. The main feature of this approach is to perform a high efficient search by changing these mechanisms according to the search condition. If the search situation seems to be stagnated, escape mechanism would be applied for shifting search point from this one to another one. On the other hand, if it observes no sign of the improvement of solutions after repeating this escape mechanism for a fixed period, convergence mechanism is applied to improve the quality of solution through an intensive local search. This paper presents a new approach, called “escaping from local optima and convergence mechanisms based on search history - SPLASH -”.

Experimental results showed the effectiveness of SPLASH and the workings of SPLASH’s two mechanisms using WFG test suites.

1. はじめに

近年、進化型多目的最適化 (Evolutionary Multi-criterion Optimization, EMO) の関連研究が活発に行われており、大きな進歩を見せている [Coello 04, Deb 01]. 中でも、多目的進化型アルゴリズム (Multi-objective Evolutionary Algorithm, MOEA) に局所探索手法を組み合わせ、解の収束性能の向上を図るアプローチは数多く提案され、大きな成果を挙げている [Ishibuchi 98, Knowles 00, Martínez 13].

一般にこれらのアプローチでは、遺伝的アルゴリズム (Genetic Algorithm, GA) などの進化型アルゴリズムでグローバルな探索を行い、何らかの近傍探索操作により局所探索を行っている場合が多い [Ishibuchi 98, Knowles 00, Martínez 13]. 一方、局所解に陥った場合に対する明示的な解決策をメカニズムに組み込んでいるものはごく一部に限定されている。

そこで本稿では、MOEA の探索性能を強化させることを目的とした、新たな局所探索メカニズム SPLASH(eScaPing from Local optima and convergence mechanisms based

on Search History) を提案する。提案手法は、探索履歴を活用した局所解脱出及び集中探索という相異なる 2 つのメカニズムに基づくものであり、未探索領域への探索と有望領域への集中的な探索を組み合わせたものとなっている。

本手法では、探索過程で得られた実数情報を離散化し保存することで探索履歴情報の効率的な利用を実現している。局所解脱出メカニズムでは、探索履歴情報全体を基に大まかな探索の様子を読み取り、未探索の領域を推定し、局所解から脱出することを試みる。一方、集中探索メカニズムでは、探索履歴情報の一部を用いて、有望な探索領域を推定し集中的な探索を行う。

それぞれのメカニズムにおける計算量は、通常の遺伝的操作において費やすものと等しく設定しており、通常の操作の代替として用いる場合、世代あたりの計算量に変化は生じない。

提案手法の有効性の検証のため、代表的な MOEA の一つである MOEA/D-DE(MOEA based on Decomposition with Differential Evolution)[Li 09] に提案手法を組み込

んだアルゴリズムを利用し、代表的なテスト問題である WFG Test Suites [Huband 05] に対して代表的な解評価指標を用いた数値実験及び探索の推移情報から提案する 2 つのメカニズムの有効性・妥当性の検証を行った。

以下、本稿における構成について述べる。まず 2 章では、多目的最適化、多目的最適化問題並びに多目的進化型アルゴリズムの一つである MOEA/D-DE について概説する。3 章ではこれまでに行われてきた関連研究に関して述べ、4 章では提案する 2 つの局所探索メカニズムの詳細について説明する。5 章において提案手法の検証実験を行い、6 章にまとめを述べる。

2. 多目的最適化

ここでは多目的最適化問題とパレート最適解の一般的な定義について述べた後、代表的な多目的進化型アルゴリズムについて、本論文の数値実験において使用した MOEA/D-DE [Li 09] について述べる。

2.1 多目的最適化問題

多目的最適化問題とは、複数個の目的関数を与えられた制約条件のもとで最小化する問題である。多目的最適化問題は以下のような式で表される [Deb 01]。

$$\begin{aligned} \min \quad & f_m(\mathbf{x}) \quad (m = 1, 2, \dots, M) \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0 \quad (j = 1, 2, \dots, J) \\ & h_k(\mathbf{x}) = 0 \quad (k = 1, 2, \dots, K) \\ & \mathbf{x} \in S \end{aligned} \quad (1)$$

解空間 S において、 J 個の不等式制約条件 g_j 、 K 個の等式制約条件 h_k によって定義される実行可能領域 $F \subseteq S$ のもとで、 M 種類の目的関数 f_m を最小化する設計変数 \mathbf{x} (\mathbf{x} は n 次元ベクトル) を求める。

一般に多目的最適化問題では、目的関数間にトレードオフの関係が存在するため、ある目的関数の値を改善するためには、少なくとも他の 1 つの目的関数の値を改悪せざるを得ないような解を求めていく。このような解のことをパレート最適解 (Pareto Optimal Solutions) と呼ぶ。全ての目的が最小化であると仮定して、 $x, y \in F$ に対して

$$\forall i \in \{1, 2, \dots, M\} : f_m(\mathbf{x}) \leq f_m(\mathbf{y}) \wedge \exists m \in \{1, 2, \dots, M\} : f_m(\mathbf{x}) < f_m(\mathbf{y}) \quad (2)$$

を満たすとき、 \mathbf{x} は \mathbf{y} を支配するといい、 \mathbf{x} が \mathbf{y} を優越していることを示す。また \mathbf{x} を支配する他の解が存在しないとき、 \mathbf{x} をパレート最適解という。多目的最適化問題では、全ての目的関数 $f_m(\mathbf{x})$ において最良の値となるような唯一の完全最適解が存在しないため、パレート最適解集合全体を求めることが第 1 の目標となっている。ただし、対象問題が設計変数が実数値などの連続値で構成される関数最適化問題の場合、パレート解は無限に存在

し、パレート最適解を特定することは原理的に不可能である。そのため、パレート最適解全体をより高精度に近似できる集合を求めることがその目的となる [金田 13]。

2.2 多目的進化型アルゴリズム

多目的最適化に進化計算の概念を応用した多目的進化型アルゴリズムは、複数のパレート解を同時に導出することが可能であり、多目的最適化問題に非常に有効な手法である [Deb 01]。以下では、代表的な多目的進化型アルゴリズムの一つである、MOEA/D-DE について説明する。

§1 MOEA/D-DE

MOEA/D-DE は、交叉法に SBX [Deb 95] (Simulated Binary Crossover) を用いる MOEA/D [Zhang 07] に対して差分進化 (Differential Evolution, DE) の遺伝的操作を用いるなど、いくつかの改良を加えたアルゴリズムである [Li 09]。探索は一様な重みベクトルに基づくスカラー化関数によって複数の部分問題に分割して行われ、本稿では、文献 [Zhang 07] に示されるスカラー化関数のうち、式 (3) で定義される Tchebycheff 関数を採用した。

$$g^{te}(\mathbf{x}|\lambda, \mathbf{z}^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(\mathbf{x}) - z_i^*|\} \quad (3)$$

ここで、 λ はスカラー化関数 $g(\mathbf{x})$ に対応する重みベクトルであり、目的関数空間における探索方向を決定する。 $\lambda = (\lambda_1, \dots, \lambda_M)^T$ は $\lambda_i \geq 0 (i = 1, \dots, M)$ 、 $\sum_{i=1}^M \lambda_i = 1$ で定義される。例えば、2 目的の目的空間を 11 方向に分割する場合、 λ は $\lambda_1 = \{0, 1\}$, $\lambda_2 = \{0.1, 0.9\}$, ..., $\lambda_{10} = \{0.9, 0.1\}$, $\lambda_{11} = \{1, 0\}$ となる。また、 \mathbf{z}^* は獲得した理想点を意味しており、最小化問題の場合には探索点 $\mathbf{x} \in \Omega$ のうち各目的における最小値 $z_i^* = \min\{f_i(\mathbf{x}) | \mathbf{x} \in \Omega\} (i = 1, 2, \dots, m)$ とするのが一般的である。ここで、 Ω は最適化中に生成した解集合である。

MOEA/D-DE のアルゴリズムを以下に示す。以下、対象問題は最小化問題と仮定する。

step1. 初期化

step1-1. 近傍の決定

重みベクトル $\lambda^1, \dots, \lambda^N$ に対して、全ての組合せでユークリッド距離を求め、各重みベクトルに対して T 個の近傍 $\mathbf{B}(i) = \{i_1, \dots, i_T\}$ を定義。

step1-2. 初期個体の生成

初期個体 $\mathbf{x}^1, \dots, \mathbf{x}^N$ を生成し、評価。

step1-3. 初期理想点の決定

獲得した理想点 $\mathbf{z}^* = (z_1^*, \dots, z_m^*)^T (z_i^* = \min\{f_i\})$ を設定。

step2. 更新

$i = 1, \dots, N$ の範囲で行う。

step2-1. 更新・選択範囲の決定

$[0, 1]$ の一様乱数を用いて、以下の式で更新・選

択範囲を決定.

$$P = \begin{cases} B(i) & \text{if } \text{rand} < \delta \\ \{1, \dots, N\} & \text{otherwise} \end{cases}$$

step2-2. 交叉

$r_1 = i$ とし, P からランダムで重複しない r_2, r_3 を選択. $\mathbf{x}^{r_1}, \mathbf{x}^{r_2}, \mathbf{x}^{r_3}$ から DE の遺伝的操作により子個体 $\bar{\mathbf{y}}$ を生成. $\bar{\mathbf{y}}$ が設計変数の領域外にあった場合, 該当の設計変数をランダムな値で置き換える.

step2-3. 突然変異

$\bar{\mathbf{y}}$ に Polynomial Mutation [Deb 96] を適用し, \mathbf{y} を生成.

step2-4. 理想点の更新

$z_j^* < f_j(\mathbf{y})$ の場合, $z_j^* = f_j(\mathbf{y}) (j = 1, \dots, m)$ とする.

step2-5. 解の更新

更新カウンタ $c = 0$ とし, 下記の 1)-3) を実行する.

- 1) $c = n_r$ もしくは P が空の場合, step3 へ進む. そうでなければ, P からランダムで j を選択.
- 2) $g(\mathbf{y}|\lambda^j, \mathbf{z}) \leq g(\mathbf{x}^j|\lambda^j, \mathbf{z})$ なら, $\mathbf{x}^j = \mathbf{y}$ とおき, $c = c + 1$ とする.
- 3) P から j を削除し, 1) に戻る.

step3. 終了判定

終了条件を満たしていれば終了. そうでなければ, step2 へ戻る.

step2-2 の DE の遺伝的操作によって得られる子個体 $\bar{\mathbf{y}}$ は, (4) によって計算される. ここで CR は交叉率で, F はスケーリングパラメータである.

$$\bar{y}_k = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}) & \text{if } \text{rand} < CR \\ x_k^{r_1} & \text{otherwise} \end{cases} \quad (4)$$

また, step2-3 の突然変異によって得られる子個体 \mathbf{y} は, 式 (5), (6) によって計算される. 式 (5) 中の b_k, a_k はそれぞれ設計変数の上限値と下限値を, p_m は突然変異率を表す. また, (6) 中の rand は $[0, 1]$ の一様乱数で, η は分布指数となっている.

$$y_k = \begin{cases} \bar{y}_k + \sigma_k \times (b_k - a_k) & \text{if } \text{rand} < p_m \\ \bar{y}_k & \text{otherwise} \end{cases} \quad (5)$$

$$\sigma_k = \begin{cases} (2 \times \text{rand})^{\frac{1}{\eta+1}} - 1 & \text{if } \text{rand} < 0.5 \\ 1 - (2 - 2 \times \text{rand})^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \quad (6)$$

以上のように MOEA/D-DE は従来の MOEA/D を改良したのとなっており, 先行研究により従来の MOEA/D に比べ多くの問題で優れた性能を示すことが分かっている.

3. 先行研究

これまで, 進化計算に局所探索を組み合わせたアルゴリズムは Memetic Algorithm (MA) [Moscato 99] と呼ばれ, 広く研究されてきた. 多目的最適化の枠組みにおいても MA を多目的に拡張した MOMA (Multi-Objective Memetic Algorithm) は多くの手法が提案されている. 多目的最適化の場合, 複数の基準を同時に考慮しなければならないため, 局所探索で生成した近傍解と元の解の優劣が付きにくいという問題点を持つ. MOGLS [Ishibuchi 98] をはじめとしたいくつかの手法は Weighted Sum 関数や Tchebycheff といったスカラー化関数を利用した単目的化を行うことでこの問題に対応している.

これらの手法における局所探索の主な役割として, 解探索の促進が考えられる. 一方で局所的な解生成は局所解への収束の恐れがあり, 何らかの局所解脱出メカニズムが必要である. 局所解脱出としてタブー探索法を組み込んだ MOEA/D [Alhindi 14] などが提案されているものの, 多目的の連続問題に対して局所解脱出メカニズムを明示的に有する MOMA に関する研究事例は非常に限られている.

また, 擬似焼きなまし法 (Simulated Annealing, SA) やタブー探索法 (Tabu Search, TS) といった, 局所解脱出機構が暗に組み込まれている単目的最適化手法を多目的に応用した研究 [Bandyopadhyay 08, Jaeggi 08] が提案されているものの, これらの手法は局所解からの脱出を実現するために解の改悪を一時的に許容するアルゴリズムとなっている.

単目的の局所解脱出法としては, いくつかの興味深い研究が存在する. Hornby による ALPS (The Age-Layered Population Structure) [Hornby 06] では, 各個体毎に, 親として使用された世代数の概念 (Age) を導入し, 初期解あるいは優秀でないと考えられる解である Age0 の個体をランダムに再生成することで局所収束の回避及び局所解からの脱出を図っている.

また, Barbulescu らは, 解の表現型によって局所解が異なることに着目し, グレイコードによる解表現を「Shifting」によって適宜変更させることで局所解の脱出を図る手法を提案している [Barbulescu 00]. この手法は前述の ALPS などと比較して, 探索の改悪を許容しないという点で異なる.

探索履歴を利用した手法としては, いくつかの有望な個体をアーカイブに保持しておき, 交叉相手として用いる手法は広く行われているほか, これまでに変異した遺伝子 (設計変数) の回数を保持し, その値から変異させるべき変数を推定する手法 [堀野 12] などが存在する.

一方, 本論文で提案する SPLASH では, 既存の MOEA に組み込んで利用する実装を想定しており, 近傍探索と局所解からの脱出の両方を適応的に切り替える仕様となっている点が既存のものとは大きく異なっている. また探

探索履歴情報から解の探索情報を近似的に推定し、次の探索戦略を決定している点でもこれらの手法とは異なっている。

4. 提案手法

本論文で提案する提案手法 (SPLASH) は、探索履歴を利用した 2 つのメカニズムで構成される。ここでは、提案手法の全体像について述べた後、各メカニズムで使用する探索履歴の保存方法について説明し、提案する局所解脱出メカニズム及び集中探索メカニズムについて詳細な説明を行う。

4.1 概要

SPLASH は、探索履歴に基づく局所解脱出メカニズム及び集中探索メカニズムで構成される。探索履歴には最適化の過程で得られた子個体の設計変数情報を離散化した情報が保存されており、各メカニズムはこの情報を活用し未探索領域もしくは有望領域への探索を実現している。

局所解脱出メカニズムは、停滞している個体の設計変数の一部を強制的に未探索の領域に置き換えており、局所解に陥りにくい探索の実現を目指すものである。

一方、集中探索メカニズムでは、局所解脱出メカニズムの適用後も停滞が続く個体に対して、既に良好な個体であると判断し、設計変数の一部を有望領域に置き換え、解の洗練化を行う。

これらのメカニズムは、MOEA と組み合わせて実装することを想定しており、アルゴリズムの高品質化、効率化の手助けとなることを期待している。提案手法を MOEA に組み込んだ場合のフローチャートを図 1 に示す。図中の状態判定では、探索の状態に基づき遺伝的操作、局所解脱出メカニズム、集中探索メカニズムのどれを実行するかを判定している。提案手法を組み込む MOEA として MOEA/D を用いた場合、各部分問題に属する個体が K 世代連続で更新されていなかった場合に局所解脱出メカニズムに入り、さらに K 世代連続で更新されていなかった場合 (つまり、 $2K$ 世代連続で更新されていない場合)、集中探索メカニズムに入る。そこからさらに K 世代連続で更新されていなかった場合 (つまり、 $3K$ 世代連続で更新されていない場合)、遺伝的操作に戻る。なお、提案する 2 つのメカニズムは遺伝的操作の代わりとして適用するメカニズムであり、2.2 節における step2-1, step2-2 にあたる。

SPLASH は、実数値問題に対して適用することを想定しており、実数値情報を離散化したものを履歴情報として保存している。以下、履歴情報の保存方法及び SPLASH のメカニズムの詳細を述べる。

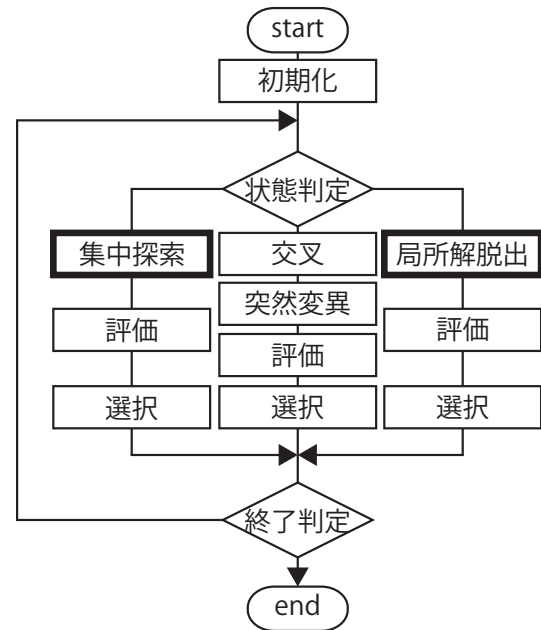


図 1 フローチャート

4.2 探索履歴の保存

2 つのメカニズムが使用する探索履歴の保存方法について説明する。本稿では文献 [Jaeggi 08] で紹介されている Long Term Memory (LTM) の概念を用いた近似的な探索履歴の保存を行う。具体的には、各設計変数を D 個の領域に等分割し、それらの領域に含まれる子個体の数をカウントした値を履歴情報として保存している。

図 2 に、履歴保存の概念図を示す。図 2 の例では、設計変数の上限 1.0, 下限 0.0 の問題に対し、 $D = 5$ として変数領域を 5 つに分割しており、領域 1 は 1.0 から 0.8, 領域 2 は 0.8 から 0.6, 領域 3 は 0.6 から 0.4, 領域 4 は 0.4 から 0.2, 領域 5 は 0.2 から 0.0 の範囲の設計変数がカウントされる。よって、 $\{0.53, 0.02, 0.9, 1.0, 0.31\}$ の変数値をもつ個体のそれぞれの属する領域は 3, 5, 1, 1, 4 となる。このカウント操作を繰り返すことで、探索の中で集中的に探索された部分は数値が大きくなり、未探索の領域は数値が小さいままとなる。本稿では、メモリを MOEA/D-DE における部分問題毎に定義し、それぞれ近傍個体のカウントを行うよう設定した。

4.3 局所解脱出メカニズム

本メカニズムは、探索履歴に基づき未探索の領域を探索させることで、局所解から脱出しやすい探索の実現を図る。本手法では、MOEA の探索過程で個体毎に停滞時間 k を記憶し、探索の停滞度合いの基準とした。 k は世代毎に解が更新されなかった場合に +1 ずつ増加させ、探索過程において一定世代 (K 世代) 更新されない個体 (つまり $k > K$ となる個体) に対して本メカニズムを適用する。なお、更新が行われた場合は $k = 0$ にリセットするものとする。

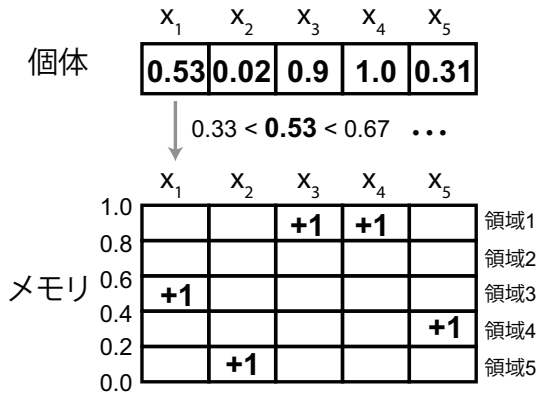


図2 履歴保存の概念図

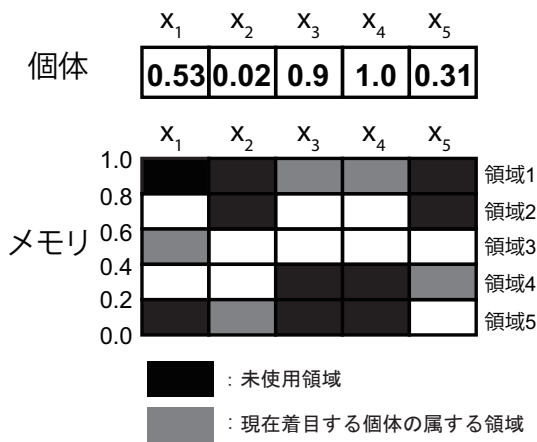


図3 集中探索メカニズムで使用するメモリ領域

以下にメカニズムの流れを示す。ここでは、前述のメモリを M とし、その要素を $m_{ij} (i = 1, \dots, D, j = 1, \dots, n)$ とした (D はメモリの離散化数で、 n は設計変数数)。

step1. 更新範囲の決定

$[0, 1]$ の一様乱数を用いて、以下の式で更新範囲を決定。

$$P = \begin{cases} B(i) & \text{if } rand < \delta \\ \{1, \dots, N\} & \text{otherwise} \end{cases}$$

step2. 変異させる設計変数の決定

着目している個体 x の設計変数のうち、ランダムに重複しない任意の複数個を選択。

step3. メモリの値の反転

$M = m_j^{worst} - m_{ij} (i = 1, \dots, D, j = 1, \dots, n)$ により求める。ここで、 m_j^{worst} は M の j 列目における最大値である。

step4. 変化量の決定

M の各列にルーレット選択を適用し、決定された要素の範囲からランダムな値を決定する。ここで得られた乱数ベクトルを y とおく。

step5. 新個体の生成

x のうち、step2 で求めた設計変数を y の要素で置き換え、新個体 x' を生成する。

step6. 更新

更新カウンタ $c = 0$ とし、下記の 1)-3) を実行する。

- 1) $c = n_r$ もしくは P が空の場合、終了。そうでなければ、 P からランダムでインデックス p を選択。
- 2) $g(x'|\lambda^p, z) \leq g(x^p|\lambda^p, z)$ なら、 $x^p = y$ とおき、 $c = c + 1$ とする。
- 3) P から p を削除し、1) に戻る。

なお、本論文における実験では step2 における変数の選択方法を変数毎に $1/\text{設計変数数}$ の確率で選択する方法を用いた。ただし、変数が 1 つも選択されなかった場合には、ランダムに 1 つの変数を強制的に選択する。

4.4 集中探索メカニズム

本メカニズムは、探索履歴に基づき有望と見込まれる領域を徹底探索させることで、解の洗練を図る。基本的には局所解脱出メカニズムと同じだが、探索履歴情報は現在着目している個体を中心とした限られた情報のみを用いる。図3に、集中探索に使用するメモリ領域の概念図を示す。図3では、5分割のメモリに対して、現在着目する個体の属する領域を含んだ3つの連続した領域のみを使用する。限定された領域のみを利用することで、「これまでの全ての探索済み領域で良好な領域」ではなく、「現個体に近い良好な近傍解」の探索を図る。

以下にメカニズムの流れを示す。ここでは、集中探索で使用するメモリ領域(図3のメモリにおける白及びグレーの領域)を M' とし、その要素を $m_{ij} (i = 1, \dots, D, j = 1, \dots, n)$ とした (D は使用するメモリ領域数であり、 n は設計変数の数を表す)。

step1. 更新・選択範囲の決定

$[0, 1]$ の一様乱数を用いて、以下の式で更新・選択範囲を決定。

$$P = \begin{cases} B(i) & \text{if } rand < \delta \\ \{1, \dots, N\} & \text{otherwise} \end{cases}$$

step2. 変異させる設計変数の決定

着目している個体 x の設計変数のうち、ランダムに重複しない任意の複数個を選択。

step3. 変化量の決定

M' の各列にルーレット選択を適用し、決定された要素の範囲からランダムな値を決定する。得られた乱数ベクトルを y とおく。

step4. 新個体の生成

x のうち、step2 で求めた設計変数を y の要素で置き換え、新個体 x' を生成する。

step5. 更新

更新カウンタ $c = 0$ とし、下記の 1)-3) を実行する。

- 1) $c = n_r$ もしくは P が空の場合、終了。そうでな

- ければ, P からランダムでインデックス p を選択.
 2) $g(\mathbf{x}'|\lambda^p, \mathbf{z}) \leq g(\mathbf{x}^p|\lambda^p, \mathbf{z})$ なら, $\mathbf{x}^p = \mathbf{y}$ とおき,
 $c = c + 1$ とする.
 3) P から p を削除し, 1) に戻る.

step2 における変数の選択方法は局所解脱出メカニズムと同様である.

5. 数値実験

提案手法の有効性の検証のため, WFG Test Suites[Huband 05]を用いた3つの数値実験を行った. 実験1では, 提案する SPLASH を MOEA/D-DE に組み込んだ場合の性能の検証を行った. 実験2では, 提案する脱出メカニズムと集中メカニズムそれぞれの有効性の検証を行った. 実験3では, SPLASH のパラメータである D 及び K を変更した場合の性能の変化について検証した.

本章では, はじめに実験1, 2, 3 で共通となるアルゴリズム及び問題設定, 評価指標, 対象問題について説明した後, それぞれの実験結果について述べる.

5.1 設定パラメータ

本実験で使用した設定パラメータを示す.

MOEA/D-DE に関するパラメータは, 近傍サイズ $T = 20$, メイティング及び選択を近傍内で行う確率 $\delta = 0.9$, 個体の更新数 $n_r = 2$, 重みベクトルの分割数 $H = 199(M = 2)$, $6(M = 5)$, $4(M = 8)$ と設定し, H の値に対応して個体数は $N = 200(M = 2)$, $210(M = 5)$, $330(M = 8)$ となる.

また, 遺伝的操作に関するパラメータは, 交叉に用いられるスケーリング $F = 0.5$, 交叉率 $CR = 1.0$, 突然変異率 $MR = 1/\text{設計変数数}$, Polynomial Mutation における分布指数 $\eta = 20$ である.

SPLASH に関するパラメータは, メモリの領域分割数 $D = 50$ (要素数は Dn 個となる) で, 集中探索ではそのうち 10 個の領域 (要素数は 10n 個となる) を用い, 変異率は $1/\text{設計変数数}$ (ただし, 最低でも 1 つの変数を変異) とした. 局所解脱出, 集中探索の適用世代数 $K = 5$ とした.

本実験では終了条件は 100,000 関数評価とし, 30 試行の平均値を比較する.

5.2 評価指標

解評価には Zitzler らによる Hypervolume (HV) [Zitzler 98] を用いた. Hypervolume は, 図4に示すようにある設定された参照点から, 得られた解集合が支配する目的関数空間上での体積を評価値とするものであり, HV 値が大きいほど優れた解集合であることを示す. 解集合の支配する空間を評価値としているため, 解集合の収束性, 多様性及び分散具合を総合的に評価する評価手法と言える.

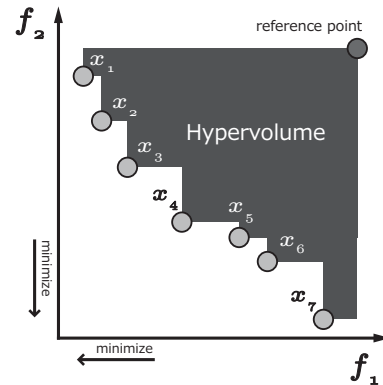


図4 Hypervolume の概念図

本実験では $f'_m(\mathbf{x}) = f_m(\mathbf{x})/2m$ ($m = 0, \dots, M$) のように各目的のスケールを正規化した解集合に対して適用した. また, 本実験において基準となる参照点 \mathbf{r} , $\mathbf{r} = \{1.5, \dots, 1.5\}$ と設定した.

5.3 対象問題

対象問題は WFG Test Suites を用いた. WFG は 9 個のベンチマーク関数から構成されている, 目的数, 設計変数数を任意に変更可能なテスト問題である. 関数の詳細については, 文献 [Huband 05] に示されている通りである. 本実験では, 設計変数数 $n = 20$, 位置変数数 $k = 2(M - 1)$, 距離変数数 $l = n - k$ とした (M : 目的数). ただし, WFG3 においては, 3 目的以上の場合に縮退した 1 次元の解集合の他に想定外の解の導出が確認されており, その対策のために文献 [Ishibuchi 16] に示す制約を付与している. WFG の特徴を表1に示す.

5.4 実験 1

実験1では, 提案手法全体としての性能評価を行う. 具体的には, WFG Test Suites を対象とした数値実験により提案するアルゴリズムの得意・不得意な問題の調査を行った. 以下, MOEA/D-DE に SPLASH を組み込んだ手法を提案手法と呼ぶ.

2, 5, 8 目的の結果を表2から表4に示す. まず, WFG1 では, 全ての場合において提案手法の HV 値が勝っている. WFG1 は単峰性だが, 偏りのある探索空間が探索の困難性を引き起こす問題と考えられる [Huband 06]. このことから, 提案手法は Tchebycheff 関数を採用した MOEA/D-DE における偏りのある探索の弱点を緩和できていると考えられる.

WFG4 は全ての場合において従来手法と比較して高い性能を誇る. WFG4 は多峰性 (ただし, 大域的には単峰性) であり, 局所解脱出メカニズムによって収束性能が向上していると考えられる. 一方で WFG9 も多峰性問題だが, 2 目的の場合において提案手法の性能は決して良好でない. WFG9 は設計変数間に依存関係を持つ非分離

表1 WFG Test Suite の特性 [Huband 05]

問題	目的	分離可能性	景観	特徴	パレートフロントの形状
WFG1	$f_{1:M}$	分離可能	単峰性	偏り, フラットな領域	凸, 複合 凸, 離散
WFG2	$f_{1:M-1}$ f_M	非分離 非分離	単峰性 多峰性		
WFG3	$f_{1:M}$	非分離	単峰性		線形, 縮退
WFG4	$f_{1:M}$	分離可能	多峰性		非凸
WFG5	$f_{1:M}$	分離可能	騙し		非凸
WFG6	$f_{1:M}$	非分離	単峰性		非凸
WFG7	$f_{1:M}$	分離可能	単峰性	変数依存	非凸
WFG8	$f_{1:M}$	非分離	単峰性	変数依存	非凸
WFG9	$f_{1:M}$	非分離	多峰性, 騙し	変数依存	非凸

表2 2目的, 20変数のHV

	従来手法	提案手法
WFG1	0.960	1.003
WFG2	1.803	1.801
WFG3	1.743	1.741
WFG4	1.423	1.453
WFG5	1.367	1.367
WFG6	1.361	1.367
WFG7	1.460	1.460
WFG8	1.381	1.371
WFG9	1.402	1.397

表3 5目的, 20変数のHV($\times 10^1$)

	従来手法	提案手法
WFG1	0.347	0.371
WFG2	0.748	0.756
WFG3	0.599	0.600
WFG4	0.641	0.693
WFG5	0.636	0.661
WFG6	0.625	0.634
WFG7	0.662	0.704
WFG8	0.611	0.655
WFG9	0.566	0.583

表4 8目的, 20変数のHV($\times 10^2$)

	従来手法	提案手法
WFG1	0.131	0.153
WFG2	0.255	0.256
WFG3	0.201	0.201
WFG4	0.214	0.228
WFG5	0.209	0.219
WFG6	0.226	0.232
WFG7	0.223	0.230
WFG8	0.209	0.218
WFG9	0.209	0.216

問題であり, WFG8も同様の傾向を示す.

全体を通してみると, 目的数が多くなるにつれて提案手法のHV値が高い傾向を示している. このことから, 提案手法は膨大な探索空間から探索履歴による有望領域の推定によって, ある程度の探索空間の削減を行った効果であると考えられる. また縮退したパレートフロントを持つWFG3においては, 多数目的の場合でも従来手法とほぼ同様の結果を示しており, 実質的な探索空間の少ない問題で提案手法が有効に働かない可能性が考えられる.

結論として, 目的数が少ないなど, 探索空間が広くない問題や変数依存性のある非分離問題に対しては提案手法は有効に働かない可能性を有することが分かった.

5.5 実験 2

実験2では, 各メカニズムの探索へ与える効果, 影響について考察する. 具体的には, 局所解脱出のみを組み込んだMOEA/D-DE (以下, 脱出のみ), 集中探索のみを組み込んだMOEA/D-DE (以下, 集中のみ)を用いて, 単峰性問題であるWFG1, 多峰性問題であるWFG4, 変数依存性を持ち非分離問題であるWFG8に対して各メカニズムの使用割合, HVの推移及び各メカニズムでの更新率から有効性の検証を行った. なお, 脱出のみは提案手法の集中探索部メカニズム部分を遺伝的操作で置換したバージョンであり, 集中のみは提案手法の局所解脱

出メカニズム部分を遺伝的操作で置換したバージョンである.

図5に脱出のみの各メカニズムの推移を, 図6に集中のみの各メカニズムの推移を示す. 図中の左縦軸が個体であり, 横軸が世代数, 右縦軸がHV値となっている. 赤, 緑, 黄の各グラフは, 各世代においてそれぞれのメカニズムを適用した個体の数を表しており, 左縦軸を参照する. 青のグラフはHVの推移を示し, 右縦軸を参照する.

例えば, 図5(a)における横軸の最終世代に着目すると, 約130個体が遺伝的操作を, 約70個体が局所解脱出を行っており, HVは1付近の値であることが読み取れる.

図5より, 序盤(60世代付近)に局所解脱出メカニズムの割合がピークを迎え, 以降は減少傾向にあることが分かる. また, 序盤の局所解脱出メカニズムの割合増加と同時に, HV値も大きく増加していることから, 局所解脱出メカニズムは序盤に有効な手法であると推測される. 次に図6より, 8目的WFG1, WFG4においては最終世代においても集中探索メカニズムの割合が多いことが分かる. また序盤に集中探索メカニズムの割合がピークを迎える傾向は局所解脱出メカニズムと同様だが, それ以降の減少が緩やかなことから, 集中探索メカニズムは中盤から終盤にかけて有効な手法であると推測される.

図7に提案手法の各メカニズムの推移を示す. また, 表5に各メカニズムの使用数及び更新率(30試行平均)を

示す。まず WFG1 の場合 (図 7(a), (b), (c)), 2 目的の場合では探索中盤から終盤にかけて遺伝的操作と局所解脱出メカニズムの割合が同程度であるが, 目的数が増えると局所解脱出メカニズムの割合が遺伝的操作を上回っていることが分かる。表 5 の解更新率からも, 2 目的の場合に低かった局所解脱出メカニズムの更新率が 8 目的では最も高くなっている。

次に WFG4 の場合 (図 7(d), (e), (f)), 探索中盤から終盤にかけて遺伝的操作の割合が最も低く, 次いで局所解脱出メカニズム, 集中探索メカニズムの順となっている。この傾向は目的数が増えるに従い顕著になっており, 実際表 5 では, 2 目的の場合には大きな差がなかった更新率が 5 目的, 8 目的では局所解脱出メカニズム及び集中探索メカニズムでの更新率が遺伝的操作による更新率を大幅に上回っている。

最後に WFG8 の場合 (図 7(g), (h), (i)), 5, 8 目的の場合でも探索中盤から終盤にかけての各メカニズムの使用割合はほぼ同じになっていることが分かる。表 5 によると, 局所解脱出メカニズムによる解更新率が全ての目的で最も低く, その傾向は 8 目的の場合でも変化がない。

以上より, 局所解脱出メカニズムは目的数の多い問題に対して有効だが, 依存関係を持つ問題に対しては必ずしも有効でないことが分かった。また集中探索メカニズムは, 全ての問題に対してある程度の解更新率を期待できることが確認できた。

5.6 実験 3

実験 3 では, SPLASH のパラメータである D 及び K を変更した場合の性能の変化について WFG1, WFG4, WFG8 を用いて検証する。

はじめに, 表 6 に $D = 2, 5, 25, 50$ の場合の HV を示す (集中探索に使用する領域数は $\lceil D/5 \rceil$ とした)。全体的な傾向として, $D = 50$ の場合に探索性能の向上が見て取れる。一方で, 8 目的 WFG1, 5 目的 WFG8 のように $D=2$ とした場合が最も有効な場合も存在することが分かった。しかし, D を大きくするにつれ, 必要なメモリ領域が膨大になる点, また計算時間も増加することから, D は 25 から 50 の範囲で設定することを推奨する。

次に, 表 7 に $K = 2, 5, 10, 20$ の場合の HV を示す。問題や目的によって適切な K が異なるものの, それらの差異は非常に小さい場合が多く, K の値による探索性能への影響は限定的であることが分かる。

6. おわりに

本稿では, MOEA に組み込むアルゴリズムとして, 探索履歴を利用した局所解脱出メカニズムと集中探索メカニズム SPLASH を提案した。SPLASH を組み込んだ MOEA/D-DE と, 組み込んでいない MOEA/D-DE との比較実験により, 提案手法が多峰性問題や目的数の多い

問題に対して有効であることが分かった。一方で, 変数依存性のある非分離問題や目的数の少ない問題に対しては有効でない場合もある。また局所解脱出メカニズムは中盤の探索の停滞時に探索の収束に貢献し, 集中探索メカニズムは探索の終盤に効果的であることが分かった。

また, 良好な領域が変数空間上のごく一部に限られている問題に対しては, 等分割による離散化手法が有効に働かない可能性があるため, 今後はより効果的な離散化手法の検討及び不得意な問題に対して効果的なメカニズムの検討を行う予定である。

謝 辞

本研究は, 北海道大学情報基盤センター共同研究採択課題であり, JSPS 科研費 26330269, 26350431, 16K00312 の助成を受けたものである。

◇ 参 考 文 献 ◇

- [Alhindi 14] Alhindi, A. and Zhang, Q.: MOEA/D with Tabu Search for multiobjective permutation flow shop scheduling problems, *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1155–1164 (2014)
- [Bandyopadhyay 08] Bandyopadhyay, S., Saha, S., Maulik, U., and Deb, K.: A simulated annealing-based multiobjective optimization algorithm: AMOSA, *IEEE Trans. on Evolutionary Computation*, Vol. 12, No. 3, pp. 269–283 (2008)
- [Barbulescu 00] Barbulescu, L., Watson, Paul J., and Whitley, L. D.: Dynamic representations and escaping local optima: Improving genetic algorithms and local search, *AAAI/IAAI*, pp. 879–884 (2000)
- [Coello 04] Coello, C. A. C. and Lamont, G. B.: *Applications of Multi-Objective Evolutionary Algorithms*, World Scientific, Singapore (2004)
- [Deb 95] Deb, K. and Agrawal, R.: Simulated binary crossover for continuous search space, *Complex Systems*, Vol. 9, No. 2, pp. 115–148 (1995)
- [Deb 96] Deb, K. and Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design, *Computer Science and Informatics*, Vol. 26, pp. 30–45 (1996)
- [Deb 01] Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*, Chichester, UK: Wiley (2001)
- [堀野 12] 堀野将晴, 佐藤寛之: 集約関数の探索履歴を用いた交叉法による進化型多数目的最適化, 進化計算学会 進化計算シンポジウム 2012 講演論文集, pp. 424–430 (2012)
- [Hornby 06] Hornby, G. S.: ALPS: The age-layered population structure for reducing the problem of premature convergence, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*, pp. 815–822 (2006)
- [Huband 05] Huband, S., Barone, L., While, L., and Hingston, P.: A scalable multi-objective test problem toolkit, *Evolutionary Multi-Criterion Optimization (EMO 2005)*, pp. 280–295 (2005)
- [Huband 06] Huband, S., Hingston, P., Barone, L., and While, L.: A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Trans. on Evolutionary Computation*, Vol. 10, No. 5, pp. 477–506 (2006)
- [Ishibuchi 98] Ishibuchi, H. and Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 28, No. 3, pp. 392–403 (1998)
- [Ishibuchi 16] Ishibuchi, H., Masuda, H., and Nojima, Y.: Pareto fronts of many-objective degenerate test problems, *IEEE Trans. on Evolutionary Computation*, Vol. 20, No. 5, pp. 807–813 (2016)
- [Jaeggi 08] Jaeggi, D., Parks, G., Kipouros, T., and Clarkson, P.: The development of a multi-objective Tabu Search algorithm for continuous optimisation problems, *European Journal of Operational Re-*

search, Vol. 185, pp. 1192–1212 (2008)

- [金田 13] 金田行雄：超多自由度系の最適化, *Optimization of Systems with Ultra Many Degree of Freedom*, 共立出版, 東京, Japan (2013)
- [Knowles 00] Knowles, J. D. and Corne, D. W.: M-PAES: A memetic algorithm for multiobjective optimization, *Proc. of 2000 IEEE Congress on Evolutionary Computation (CEC)*, Vol. 1, pp. 325–332 vol.1 (2000)
- [Li 09] Li, H. and Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans. on Evolutionary Computation*, Vol. 12, No. 2, pp. 284–302 (2009)
- [Martínez 13] Martínez, S. Z. and Coello, C. A. C.: A hybridization of MOEA/D with the nonlinear simplex search algorithm, *2013 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2013)*, pp. 48–55 (2013)
- [Moscato 99] Moscato, P.: *New ideas in optimization*, chapter Memetic Algorithms: A Short Introduction, pp. 219–234, McGraw-Hill Ltd., UK, Maidenhead, UK, England (1999)
- [Zhang 07] Zhang, Q. and Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evolutionary Computation*, Vol. 11, No. 6, pp. 712–731 (2007)
- [Zitzler 98] Zitzler, E. and Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study, *Parallel Problem Solving from Nature - PPSN-V*, pp. 292–301 (1998)

[担当委員：小林 亮太]

2016年11月01日 受理

著者紹介



左文字 響

2015年室蘭工業大学情報電子工学系学科卒。現在、室蘭工業大学大学院工学研究科 博士前期課程在学中。進化型多目的最適化、進化計算等の研究に従事。情報処理北海道シンポジウム 2015 学術研究賞受賞。



渡邊 真也(正会員)

2003年同志社大学大学院工学研究科 博士後期課程修了。工学(博士)。同年 産業総合研究所 生命情報科学研究センター 特別研究員, 2004年 立命館大学 情報理工学部 講師等を得て現在 室蘭工業大学大学院 しくみ情報系領域 准教授。進化計算, 最適設計, データマイニング等の研究に従事。2005年情報処理学会山下記念研究賞, 2009年 IEEE CIS Japan Chapter Young Researcher Award 等受賞。IEEE, 情報処理学会, 進化計算学会, 日本知能情報フェジ学会,

システム制御情報学会各会員。

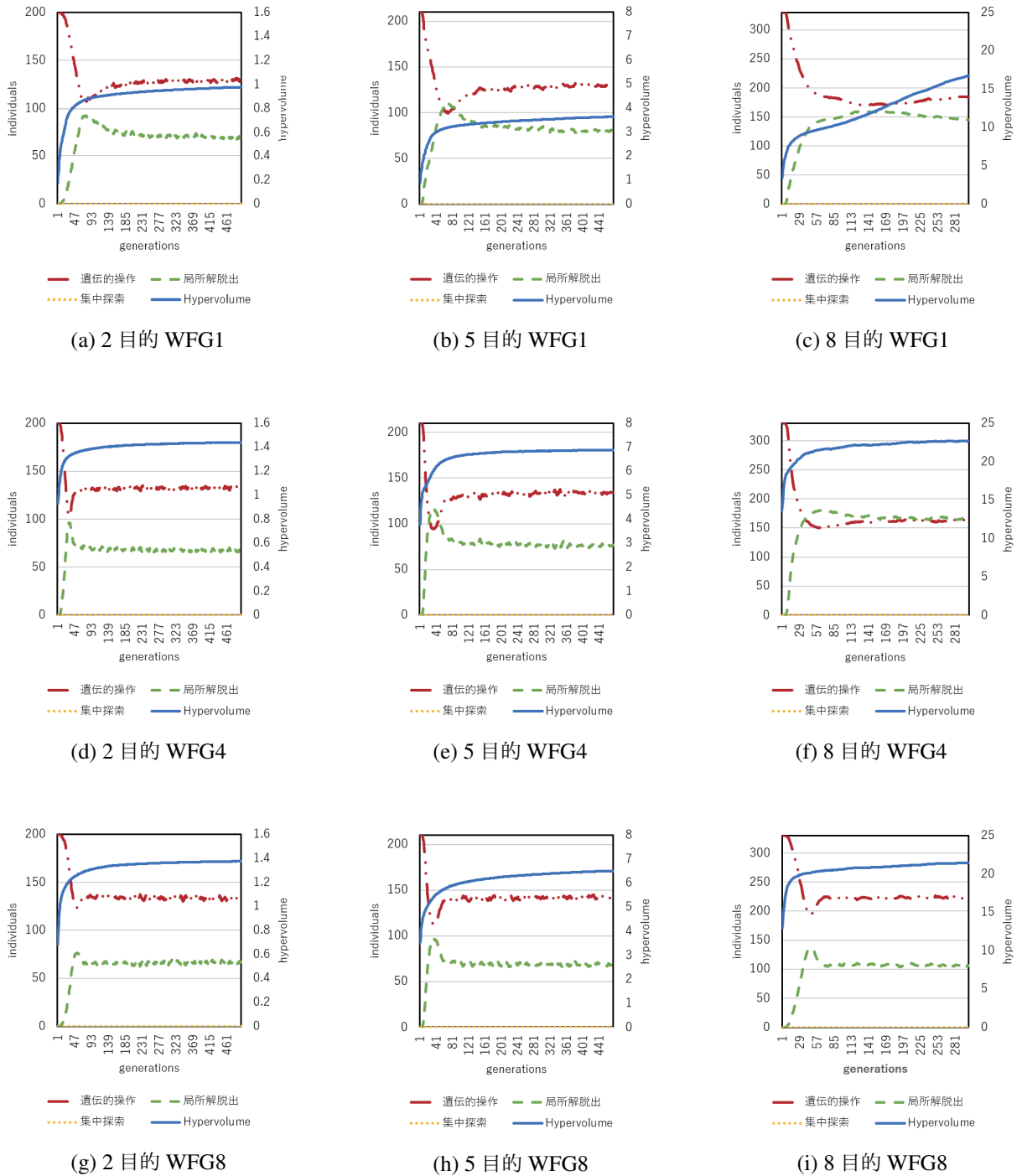


図 5 脱出のみにおけるメカニズム使用割合及び HV の推移

表 5 各メカニズムにおける解の更新率及び使用回数

		2 目的			5 目的			8 目的		
		遺伝的	脱出	集中	遺伝的	脱出	集中	遺伝的	脱出	集中
WFG1	使用数	34051	28718	37031	31370	33684	34736	37864	35508	26297
	更新率	0.012	0.006	0.010	0.008	0.014	0.016	0.012	0.022	0.019
WFG4	使用数	33587	31195	35019	30231	33854	35706	26469	41862	31339
	更新率	0.004	0.002	0.005	0.003	0.008	0.011	0.005	0.019	0.021
WFG8	使用数	34947	28099	36753	34825	31147	33818	39677	30171	29822
	更新率	0.010	0.001	0.006	0.006	0.002	0.009	0.008	0.001	0.003

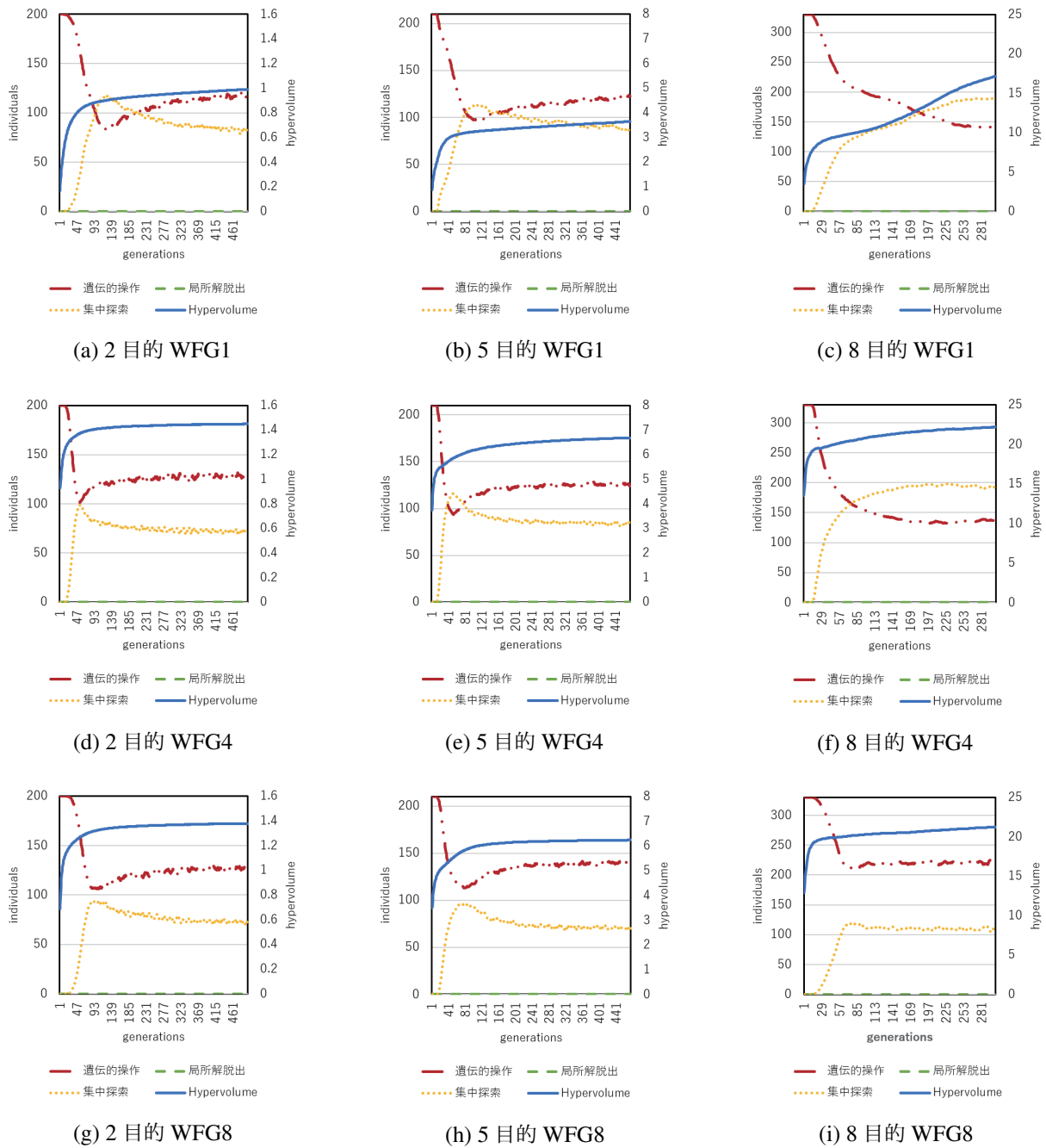


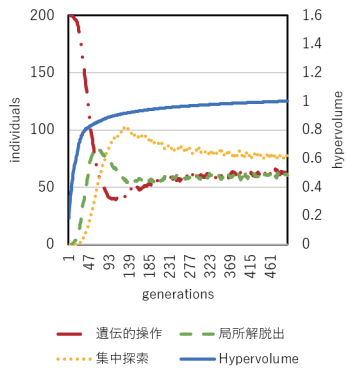
図 6 集中のみにおけるメカニズム使用割合及び HV の推移

表 6 D の変化による HV 値への影響 ($K=5$)

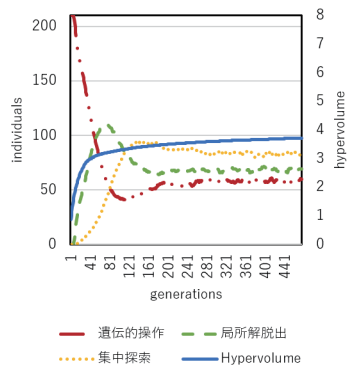
	2 目的				5 目的 ($\times 10^1$)				8 目的 ($\times 10^2$)			
	$D=2$	$D=5$	$D=25$	$D=50$	$D=2$	$D=5$	$D=25$	$D=50$	$D=2$	$D=5$	$D=25$	$D=50$
WFG1	0.967	0.972	0.995	1.003	0.356	0.361	0.367	0.371	0.154	0.149	0.147	0.153
WFG4	1.443	1.447	1.451	1.453	0.685	0.689	0.692	0.693	0.226	0.225	0.228	0.228
WFG8	1.368	1.371	1.370	1.371	0.656	0.654	0.655	0.655	0.216	0.214	0.216	0.218

表 7 K の変化による HV 値への影響 ($D=50$)

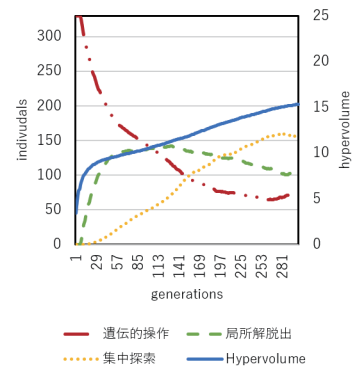
	2 目的				5 目的 ($\times 10^1$)				8 目的 ($\times 10^2$)			
	$K=2$	$K=5$	$K=10$	$K=20$	$K=2$	$K=5$	$K=10$	$K=20$	$K=2$	$K=5$	$K=10$	$K=20$
WFG1	1.003	1.003	0.999	0.993	0.368	0.371	0.371	0.366	0.138	0.153	0.173	0.144
WFG4	1.452	1.453	1.453	1.452	0.692	0.693	0.693	0.695	0.228	0.228	0.229	0.227
WFG8	1.371	1.371	1.374	1.377	0.655	0.655	0.653	0.656	0.215	0.218	0.216	0.216



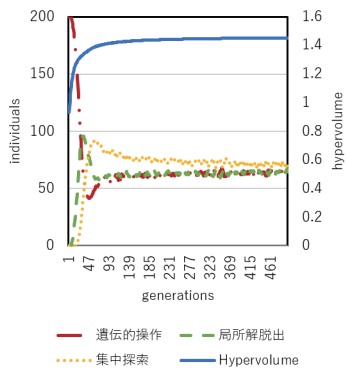
(a) 2 目的 WFG1



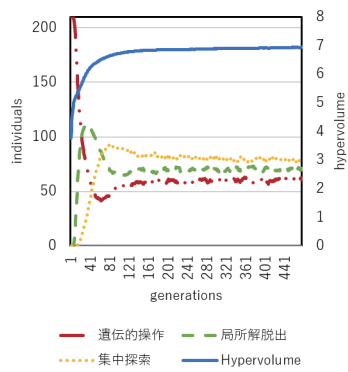
(b) 5 目的 WFG1



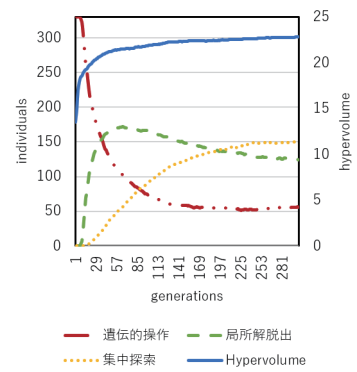
(c) 8 目的 WFG1



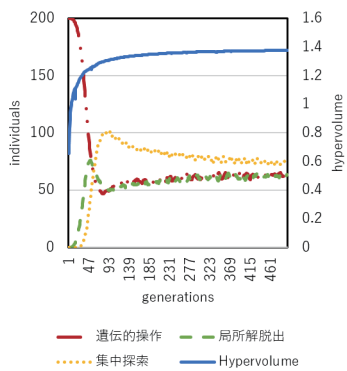
(d) 2 目的 WFG4



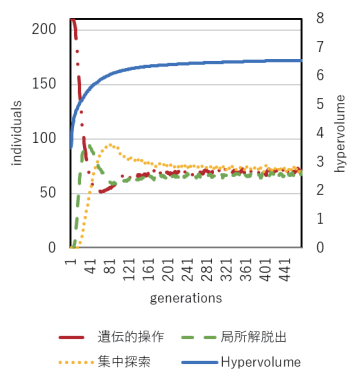
(e) 5 目的 WFG4



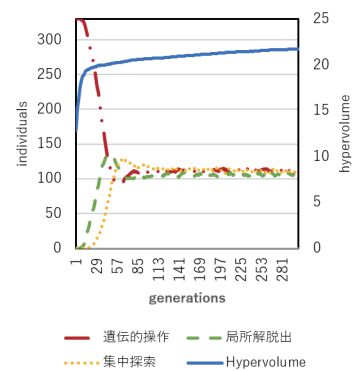
(f) 8 目的 WFG4



(g) 2 目的 WFG8



(h) 5 目的 WFG8



(i) 8 目的 WFG8

図 7 提案手法におけるメカニズム使用割合及び HV の推移