

可変ピボット型一般化による多様性向上と高速化

佐藤 圭二^{†1,a)} 佐藤 寛之^{†1,b)} 高玉 圭樹^{†1,c)}

概要: 本研究では、ピボット型一般化の効率的な一般化手法として、一般化の促進による計算コストの増加を抑え、かつより多様な一般化解を生成する手法を提案する。ピボット型一般化とは遺伝的アルゴリズムを用いて、多目的環境にロバストな（解の一部を変更しても性能劣化を抑える）解の探索に有効な手法であるが、従来手法では、解の変更可能カ所はハミング距離が2の解同士しか一般化の対象にならず、多様な一般化解を生成することができなかった。また、一般化された解が増えるにつれ、計算コストが増加するという問題も抱えていた。そこでわれわれは、解の変更可能点を自在に設定できる手法に拡張し、また、一般化解の増加していた計算コストを一定おさえるモデルにした。さらに、このモデルの改良により、従来手法よりもより多様な一般化解の生成に成功した。

SATO KEIJI^{†1,a)} SATO HIROYUKI^{†1,b)} TAKADAMA KEIKI^{†1,c)}

1. はじめに

実環境において、地震などで鉄道、道路のインフラが機能停止した場合、早急な路線網構築が求められる [6] が、駅の状態などの環境は時間と共に変化し、路線の一部を変更せざるを得なくなった場合、路線網のパフォーマンスが低下する恐れがある。しかし、新しい環境に適応するために路線網を大幅に変更しては利用客の混乱に繋がり、効率的ではない。具体的には、求めた解が環境に対応できなくなった（路線の一部を変更せざるを得なくなった）際に、求めた解を若干修正し、新たな環境に適応できる（路線の一部のみを変更し、輸送性能を保つ）ような解が求められ、佐藤らは学習分類子システム (Learning Classifier System) [1] の概念である一般化 (generalization) に着目し、多目的空間で、若干の変更で新たな環境に適応できる解をまとめて表現するピボット型一般化 [4] を提案した。ピボット型一般化とは、ベースとなる 0/1 ビットからなる解（ピボット解）に "s#" (swap#) の文字の組を付加することにより、ピボット解の、"s#" の付加されたカ所を変更 (0/1 のビット反転) されても解の性能をある程度保つような解をまとめて保持するものである（ピボット解の"s#"を0/1反転した解を特殊解と呼ぶ）。しかし、ピボット型一般化解の評

価方法は特殊解のフロントランクの値を使用するため、一般化が促進（特殊解の数が増加）すればするほど一般化解の評価時間が増大するという問題があった。また、ピボット型一般化解を生成する手法も、母集団内の解のハミング距離を元に"s#"を付加（ハミングマッチング）するものであったが、"s#"の数を2個に限定していることに加え、世代交代後にハミングマッチングを実施するため、一般化解の多様性が落ちてしまうという問題があった。

以上の背景から、本論文では一般化の促進による計算コストの増加を抑え、かつ多様な特殊解をもつピボット解を生成するための手法を提案し、ナップサック問題に適用することによって有効性を検証することを目的とする。

2. 多目的最適化とピボット型一般化

2.1 多目的進化型アルゴリズム

2.1.1 多目的最適化問題

多目的最適化問題は、単一目的の場合と異なり、唯一の最適解を得ることは難しい。これは、複数の評価基準がトレードオフの関係にある場合に、一方の評価の改善が他方の改悪になるからである。そのため、多目的最適化では、パレート最適解という概念を用いて解探索を行う。図 1 は 2 目的の最大化問題の例を示しており、円はそれぞれ解を示している。front F_1 に属するすべての解同士は、それぞれ f_1 または f_2 の値が優れており、どの解が優れているかを決定できない集団となる。一方、front F_1 と front F_2 の関係を見ると front F_2 の全ての解は front F_1 のいずれか

^{†1} 現在、電気通信大学
Presently with University of Electro Communications

a) keiji@cas.hc.uec.ac.jp

b) sato@hc.uec.ac.jp

c) keiki@inf.uec.ac.jp

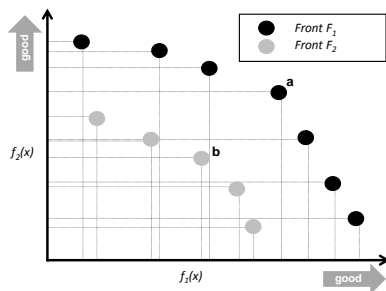


図 1 フロントランキング

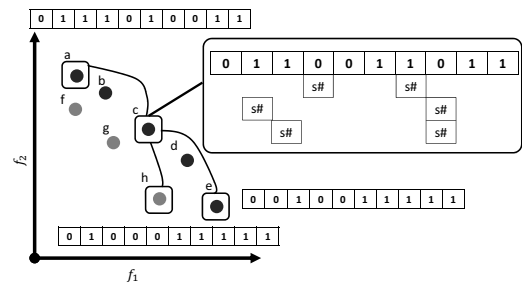


図 3 s# によるピボット一般化

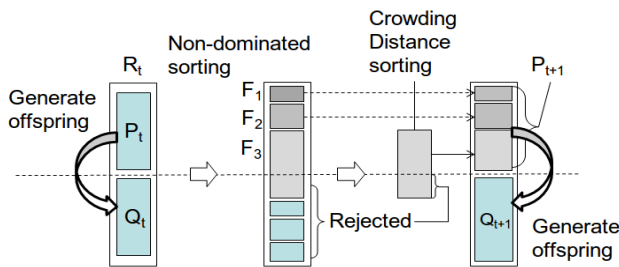


図 2 NSGA-II フロー

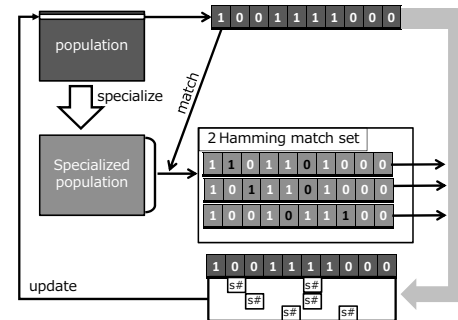


図 4 ピボット型一般化解の生成

の解に支配 (f_1, f_2 共に劣る) されている。

2.1.2 NSGA-II

多目的進化型アルゴリズムは、図 1 に示すような、複数の目的関数の下で互いに優劣がつけられない解集団を求める手法である。多目的進化型アルゴリズムの代表的な手法である、NSGA-II[2] のアルゴリズムを図 2 に示す。 t 世代の集団 R_t について、まず集団内の各個体を支配されないレベルでランク付けし、フロントと呼ばれる階層に分類 (F_1, F_2, \dots) する。そして、上位フロントから順に解集団全体の半数の個体を選び、親集団 P_{t+1} とする。同一フロントに存在する解の優劣は、混雑距離 (Crowding distance: CD) と呼ばれる目的関数空間における解同士の距離を考慮して決定する (具体的には、混雑距離が大きいほど、選択されやすくなる)。次に、 P_{t+1} からフロントと CD を考慮したトーナメント選択によって親個体を選び、これらに交叉、突然変異を施して子集団 Q_{t+1} を生成する。親集団 P_{t+1} と子集団を統合した $P_{t+1} \cup Q_{t+1}$ を次世代の集団 R_{t+1} とし、同様の操作を繰り返して解を進化させる。

2.2 ピボット型一般化

2.2.1 ピボット型一般化の解表現

ピボット型一般化は、学習分類子システムの“#”(don't care) という、0/1 のどちらでも良いという一般化の概念を拡張したものであり、その概念を図 3 に示す。“#”はすべての 0/1 の組合せを表現するため、“##”は {“00”, “01”, “10”, “11”} の 4 つすべてを表現してしまうが、ピボット型一般化は、“s#”(swap #) を用いて {“01”, “10”} だけをまとめる表現が可能となる。図 3 は 2 目的空間でのピボット型一般化の例を示しており、解 c の遺伝子の左から 4 番

目と 7 番目を“10”と変更することで解 a を表現し、同様にベースとなる解 (ピボット解) の交換可能な部分を“s#”で表現することで解 a, c, e, h をまとめて表現が可能となる。

2.2.2 一般化解の生成 (2 ハミングマッチング)

ピボット型一般化は、図 4 に示すように母集団内の個体同士で実施する。まず、図 4 の左上の母集団 (population) から、特殊解の集団 (Specialized population) を生成する。ここで、特殊解とは、すべて 0/1 で表現される解のことである。次に、population から 1 つ個体を抜き出し、ピボット解と、Specialized population の中で 2 カ所だけ異なる部分をもつ解の集団 (2 ハミングマッチセット) を作成する。最後に、ピボット解と異なる箇所に s# を加えた、新たな一般化解を作成し population の個体を更新する。これを population 内の個体すべてに実施する。

2.3 多目的空間におけるピボット型一般化解の評価

s# をもつ一般化解の評価方法を図 5 に示す。図 5 は横軸を f_1 、縦軸を f_2 とした最大化問題で、s# を持つ解 A (一般化された解) は、解 b と解 c (特殊解) を一般化している。ここで、一般化解のランクは特殊解のランクの平均値、混雑距離は特殊解の混雑距離の和を与える。

2.4 Sharp distance

Sharp distance とは目的関数値が大幅に異なる特殊解の一般化を防ぐために (可能な限り、ピボット解と同程度の性能を持つ解を求めるために) 導入され、図 6 のように表せられる。具体的には、式 (1) に示すように、ピボット解と

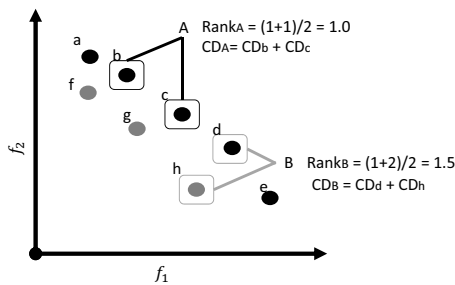


図 5 一般化解の評価値計算

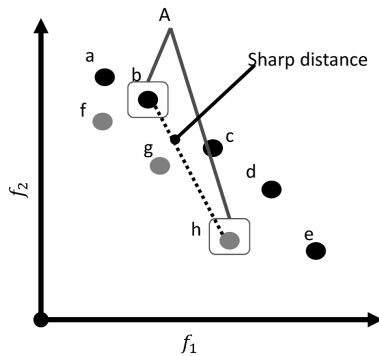


図 6 Sharp distance

それぞれ特殊化された解との距離であり、2ハミングマッチングにより一般化解を生成する際に、ピボット解とそれぞれ特殊解（2ハミングマッチセット内の解）との Sharp distance を計算し、Sharp distance が閾値 θ_{SD} を超えていた場合、その特殊解を一般化の対象外とする。このとき、2つの解の距離は母集団内での最小値と最大値で正規化して求める。ここで、 $\#object$ は目的数、 $Ind_{piv}Fit_i$ はピボット解の目的軸 i の評価値、 $Ind_{sp}Fit_i$ は特殊解の目的軸 i の評価値、 $maxFit_i$ 、 $minFit_i$ は全ての特殊解の中での目的軸 i における最大、最小評価値を示している。

$$SharpDistance = \sqrt{\sum_{i=1}^{\#object} \left(\frac{Ind_{piv}Fit_i - Ind_{sp}Fit_i}{maxFit_i - minFit_i} \right)^2}$$

(1)

2.5 ピボット型一般化を用いた多目的最適化

ピボット型一般化を用いた多目的最適化のフローを図 7 に示し、詳細を以下に示す。

Step1: NSGA-II と同様にランクと混雑距離を用いたトーナメント選択で親選択し、子集団 Q_t 生成し、 R_t を作成する。

Step2: R_t 内の重複する個体を削除し、重複する解を取り除いた R'_t を生成する。

Step3: 各個体を 2.3 節の通りに計算し、NSGA-II と同様にランキングと混雑距離から次母集団を選択する。

Step4: 次母集団に 2ハミングマッチングを実施し、一般化解を生成する。このとき、2.4 節の Sharp distance

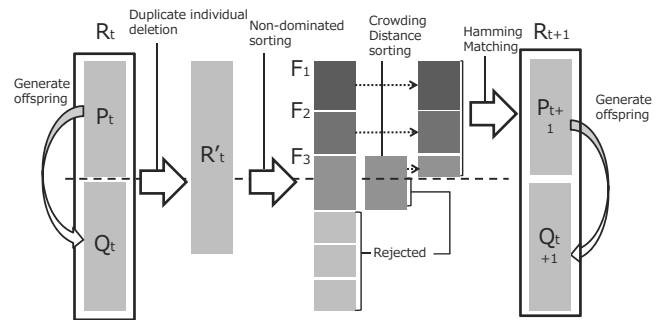


図 7 ピボット型一般化の最適化フロー

を求め、ピボット解と特殊解との Sharp distance が閾値 θ_{SD} を超えた場合、一般化の対象から外す。

Step5: 最終世代なら終了し、そうでないなら Step1 に戻る。

2.6 ピボット型一般化の問題点

2.6.1 計算コストの増大

ピボット型の一般化方式では、特殊解の集団 (Specialized population) を作成した後に 2ハミングマッチングを実施するため、一般化が促進すればするほど計算時間が増加する問題がある。ハミングマッチングの計算コストは個体数を N 、遺伝子長を l としたとき、 $O(lN^2)$ であるため、特殊解が増加すると指数的に計算コストが増加する。また、一般化解の計算方式も特殊解のランクや混雑距離を元に計算するため、ランキングの計算も特殊解の集団で実施する必要があり、こちらの計算コストも個体数を N 、目的数を M とすると $O(MN \log N)$ であるため、こちらも個体数の増加による計算コストが無視できない。

2.6.2 一般化解の多様性の限界

ハミングマッチングの実施箇所は世代交代が終わった後に実施するため、Sharp distance が閾値 θ_{SD} 以内にある解も世代交代時に削除される可能性がある。特に解空間が広い問題では世代が進むにつれてほとんどの解がランク 1 となるため、ランク 2 だとしても評価値がほとんど変わらないという解を一般化する機会を失う。

また、問題に応じてはハミング距離が 3 でも許容されるケースが考えられるが、ハミング距離を 2 に限定しているため、ハミング距離を 3 の解を一般化できず、より多様な解を一般化することができない。

3. ピボット型一般化の高速化と多様化

これらの問題点を解決するため、特殊解の増加による計算コストの増加を抑え、多様性を増加させる手法を提案する。

3.1 多様な一般化解の生成 (可変ハミングマッチング)

ピボット型一般化は、図 8 に示すように R_t 内の個体同士 (親母集団 P_t と子母集団 Q_t からなる集団) で実施する。

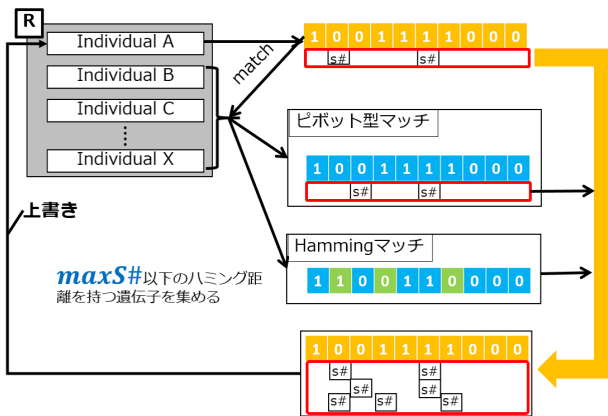


図 8 改良型一般化解生成

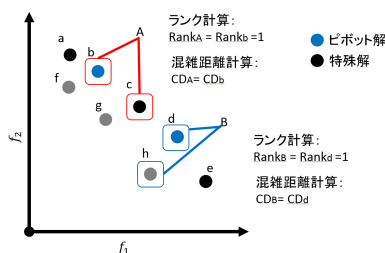


図 9 改良型一般化解評価計算

まず、図 8 の左上の母集団 (population) から 1 つ個体を抜き出し、母集団内のその他の解のピボット解と同じピボット解をもつ一般化解を抜き出し、”s#”の組をコピーする。

次に、母集団内のその他の解のピボット解と指定した $maxS\#$ だけ異なる部分をもつ解の集団 (ハミングマッチセット) を作成し、ピボット解と異なる箇所に $s\#$ を加えた、新たな一般化解を作成し母集団の個体を更新する。これを母集団内のすべての個体すべてに実施する。

R_t でハミングマッチングを実施する事により、 θ_{SD} 以内の Sharp distance を持つが世代交代で削除される解 (P_{t+1} に残らない解) を事前に一般化し、より多様な一般化解を作成することができる。

3.2 多目的空間におけるピボット型一般化解の評価

多目的空間で $s\#$ を持つ解は、複数のランクを持つことになる。そこで、 $s\#$ をもつ一般化解の評価方法を図 9 に示す。図 9 は横軸を f_1 、縦軸を f_2 とした最大化問題で、 $s\#$ を持つ解 A (一般化された解) は、解 b と解 c (特殊解) を一般化している。ランクとピボット解のランクと混雑距離はピボット解の値をそのまま設定する。

特殊解のランクと混雑距離を使用する従来手法では、特殊解のランクと混雑距離を求める必要があり、特殊解すべてでランクと混雑距離を計算する必要があったが、この改良により、ピボット解のみのランクと混雑距離の計算をすれば良く、一般化の促進にかかわらず計算コストの増加を抑えることができる。

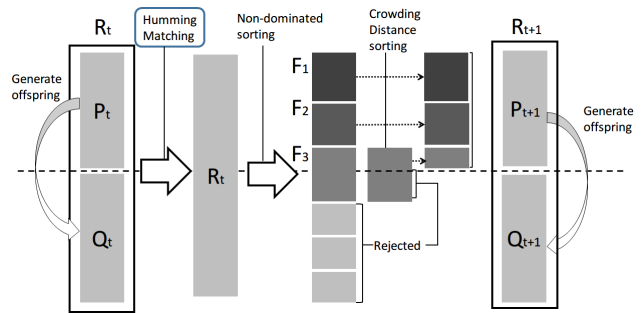


図 10 提案ピボット型一般化最適化フロー

3.3 改良型ピボット型一般化を用いた多目的最適化

ピボット型一般化を用いた多目的最適化のフローを図 10 に示し、詳細を以下に示す。

Step1: NSGA-II と同様にランクと混雑距離を用いたトーナメント選択で親選択し、子集団 Q_t 生成し、 R_t を作成する。

Step2: R_t にピボットマッチングとハミングマッチングを実施し、一般化解を生成する。このとき、Sharp distance を求めピボット解と特殊解との Sharp distance が閾値 θ_{SD} を超えた場合、一般化の対象から外す。

Step3: 各個体を 3.2 節の通りに計算し、NSGA-II と同様にランキングと混雑距離から次母集団を選択する。

Step4: 最終世代なら終了し、そうでないなら Step1 に戻る。

4. 実験

4.1 多目的ナップザック問題

多目的最大化問題である多目的 0/1 ナップザック問題は、以下のように表される。

$$\begin{aligned} \text{Maximize } f(x) &= (f_1(x), f_2(x), \dots, f_k(x)) & (2) \\ \text{subject to } \sum_{j=1}^n w_{ij}x_j &\leq c_i, i = 1, 2, \dots, k. \end{aligned}$$

$$f_i(x) = \sum_{j=1}^n p_{ij}x_j \leq c_i, i = 1, 2, \dots, k. \quad (3)$$

x は n 次元のベクトルであり、 p_{ij} と w_{ij} はそれぞれナップザック i に入れるアイテム j の価値と重さ、 c_i はナップザック i の容量を示している。解は長さ n のビット列として表現される。ナップザックの容量は、全てのアイテムを詰めた場合の重さに実行可能率 ϕ を乗算した値を設定する。

4.2 実験内容

提案手法の性能を検証するため、実行可能率 $\phi = 0.5$ の 2 目的 500 アイテムナップザック問題を使用する。パラメーターとして、各実験で母集団サイズ N は 200、突然変異確率は 0.002 とする。

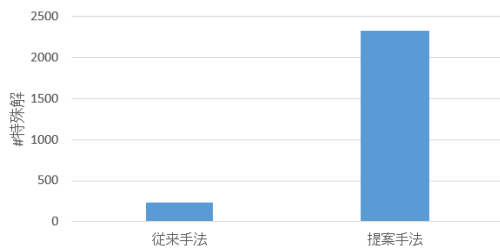


図 11 実験結果 1-1: 改良ハミングマッチングによる特殊解の個数

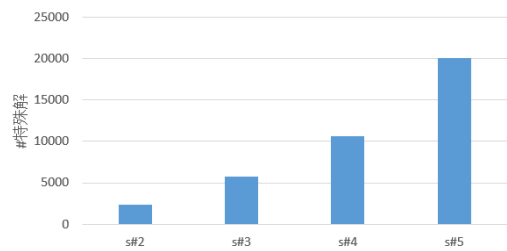


図 12 実験結果 1-2: maxS#による特殊解の個数

4.2.1 実験 1-1: 一般化解多様化検証: ハミングマッチング実行力所の検証

従来のピボット型一般化と、提案したハミングマッチングの実行力所を変更した一般化手法で得られた特殊解の個数を比較、検証する。従来のピボット型マッチングは2ピボット型マッチングを使用し、提案手法のピボット型マッチングは $maxS\# = 2$ を使用する。

4.2.2 実験 1-2: 一般化解多様化検証: s#個数の増加

従来の2ハミングマッチングを可変にしたケースで得られた特殊解の個数の一般化がもつ特殊解の個数を比較、検証する。 $maxS\#$ は2~5に設定する。

4.2.3 実験 2: 高速化検証

従来のハミングマッチングの計算時間と提案手法の計算時間を比較する。比較する力所はハミングマッチングにかかる時間とする(提案手法はピボットマッチングとハミングマッチングの合計)。

4.2.4 実験 3: 最適性検証

従来手法と提案手法の最適性を検証する。結果として得られた最適性は目的関数空間において支配する超体積(HyperVolume: HV)を比較するS-metricを用いる[3]。S-metricは、結果のパレート最適解の多様性及び収束性を同時に評価可能な尺度である。多様性とはパレート最適解が広範囲にわたって得られていることを示し、収束性とは結果のパレート最適解が真のパレート最適解(理論上最適なパレート最適解)にどれだけ近づいているかを示す尺度である。

4.3 実験結果

4.3.1 実験結果 1-1: 一般化解多様化検証: ハミングマッチング実行力所の検証

従来手法と、提案手法の $maxS\#$ を2にしたケースで特殊解が生成された個体数の結果を図11に示す。縦軸が、最終世代で得られた一般化解のがもつ特殊解の合計を示している。この図から明らかなように、提案手法がもつ特殊解の個数が大幅に上回っていることが分かる。

4.3.2 実験結果 1-2: 一般化解多様化検証: s#個数の増加

提案手法の $maxS\#$ を2~5まで増加した結果を図12に示す。横軸が $maxS\#$ の値を、縦軸が、最終世代で得られた一般化解のがもつ特殊解の合計を示している。この図

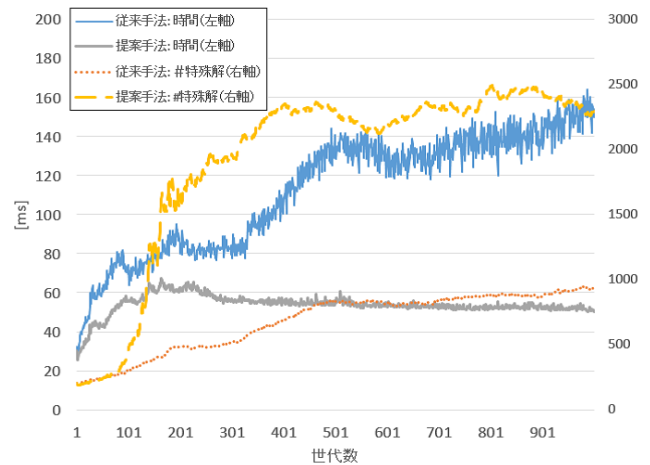


図 13 実験結果 2: 計算時間と特殊解の個数

から明らかなように、投入する $s\#$ の数が増えるほど特殊解が多く、多様な解を一般化できていることが分かる。

4.3.3 実験 2: 高速化検証

図13にそれぞれの世代での特殊解の個数と、そのときの個体評価とハミングマッチングにかかった時間を示す。横軸が世代数で、左縦軸が計算時間 [ms] を、右縦軸が特殊解の合計になっている。まず従来手法では特殊解の増加に伴い計算時間が上がっていることが分かり、このことから一般化が促進すればするほど計算コストが増加している事が見て取れる。一方提案手法の世代数700以降に着目すると特殊解の個数が増加しているにも関わらず、計算時間が増加しておらず一般化の促進による計算時間の増加を抑えることができていることが分かる。

4.3.4 実験 3: 最適性検証

図14に、最終世代で得られたHyperVolumeの値を示した。横軸が実験ケースで、縦軸がHyperVolumeの値となっている。この図から、最適性の面でも従来手法が優れていることが分かる。

5. 考察

5.1 一般化促進の効果

従来手法と提案手法の最終世代の解(特殊解含む)を図15に示した。縦軸と横軸はそれぞれ $f1$ と $f2$ の値を示している。従来手法はパレートフロントに近い解だけが一般化されているのに対し、提案手法は従来手法よりも多様

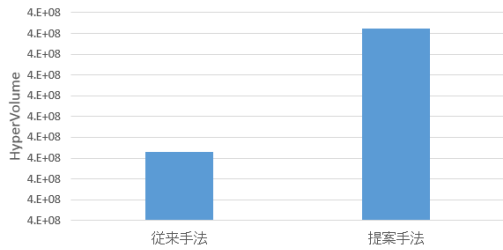


図 14 実験結果 3 : HyperVolume

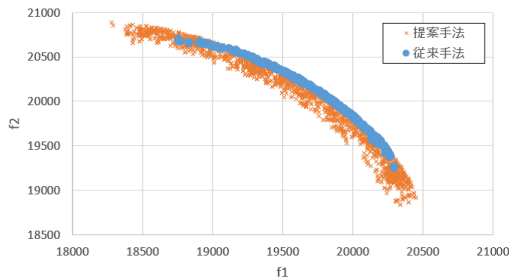


図 15 最終世代の解

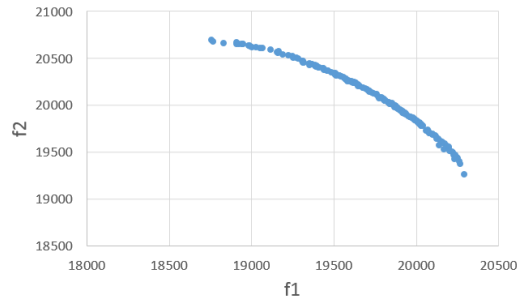


図 16 従来手法の最終世代のピボット解

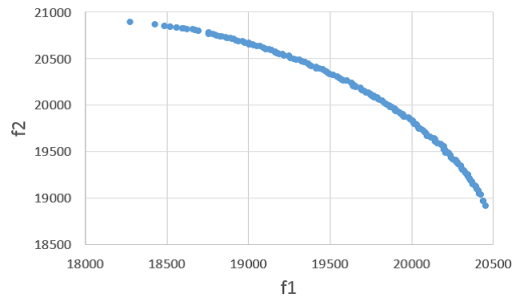


図 17 提案手法の最終世代のピボット解

な解が一般化されていることが分かる。これは、従来手法はハミングマッチングを世代交代後で実施するためにフロントランクが高い解だけで一般化しているのに対し、提案手法は世代交代前にハミングマッチングを実施するため、より多様な (Sharp distance が θ_{SD} 以内の) 解が残っているからだと考えられる。

ハミングマッチングは、一部を修正してもパフォーマンスが θ_{SD} 以内にあるものを一般化することが目的であったが、従来手法のハミングマッチングは世代交代後に実施するため、 θ_{SD} 以内にある解も削除されてしまい、一般化が提案手法よりも促進しなかったのだと考えられる。

5.2 最適性向上の効果

従来手法と提案手法の最終世代のピボット解を図 16 と図 17 に示した。縦軸と横軸はそれぞれ f_1 と f_2 の値を示している。図 16 と図 17 の f_1 の 19000 付近を見ると提案手法の解は一様に分布しているのに対し、従来手法の解は密度がバラバラに分布している。これは、一般化解の混雑距離計算の違いだと考えられる。従来手法の混雑距離は特殊解がもつ混雑距離の和となっていたため、特殊解を多くもつ一般化解が残りやすく、解が一様に分布せずに偏ってしまい、解の探索性能に影響が出たのだと考えられる。一方提案手法の混雑距離はピボット解のみの値を使用しているため、NSGA-II と同様の働きをし、探索性能が維持できたのだと考えられる。

6. おわりに

本研究では、ピボット型一般化のより効率的な一般化手法として、一般化の促進による計算コストの増加を抑え、かつより多様な一般化解を生成する手法を提案した。提案

した一般化の性能検証のためにナップサック問題を用いて詳細に検証した結果、以下の知見を得た。まず、(1) ハミングマッチングを R_t に対し実施することで一般化解の多様性が向上したこと、(2) ハミングマッチングをピボット解のみで実施することで一般化の促進による計算コストの上昇を抑え事ができたこと、(3) $s\#$ の投入個数を可変にし、より多様な一般化が可能になったこと、(4) 一般化解の評価計算をピボット解の値を使用する事で、最適性が向上することが明らかになった。今後の課題としては、適切な $s\#$ の設定方法は、実問題への適用などが考えられる。

謝辞 本研究は JSPS 科研費 25・9902 の助成を受けたものです。

参考文献

- [1] Holland, J. H. : Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-based System, *Machine Learning*, Vol.2, pp.593-623 (1986)
- [2] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", *KanGAL report 200001*, (2000)
- [3] Zitzler, Eckart, and Lothar Thiele, "Multiobjective optimization using evolutionary algorithms? a comparative case study." *Parallel problem solving from nature? PPSN V*. Springer Berlin Heidelberg, (1998)
- [4] 佐藤圭二, 佐藤寛之, 高玉圭樹, 環境変化に適応するためのピボット型一般化, 計測自動制御学会論文集, (to appear)
- [5] 間島 隆博, エージェントに基づく複雑ネットワークによる路線網構築, 博士学位論文, 東京工業大学, (2009)
- [6] 間島 隆博, 樋富 和夫, 勝原 光治郎, 災害時緊急輸送システムの技術開発に関する研究, 海上技術安全研究所報告, Vol.6, No1, pp.1-35 (2006)