

多様な満足解獲得手法の多数制約付き 最適化問題への適用に関する検討

丸山 功貴^{1,a)} 吉川 大弘¹

概要：遺伝的アルゴリズムを実問題に適用する場合、設計の選択肢を確保するため、設計変数パターンが異なり、評価値が目標値を満たした満足解を、複数獲得することが求められる場合がある。本稿では、多様な満足解の獲得手法を提案し、多数制約付き最適化問題に適用する。また、従来の設計変数の多様性を維持する手法との比較を行い、実問題への適用可能性について検討する。

1. はじめに

計算機の性能向上に伴い、進化計算手法の一つである遺伝的アルゴリズム (Genetic Algorithm: GA) の工学的応用が盛んとなった [1][2]。GA においては一般的に、評価値を基準として、高い評価値を持つ解の獲得を目的とする。しかし、その工学的応用、すなわち実問題においては、評価値が高い 1 つの解を獲得するより、評価値が要求される目標値を満足している解 (満足解) を、複数獲得することが求められる場合がある。また、設計の選択肢を確保するため、満足解は設計変数空間上において多様である (互いに離れている) ことが要求される。しかし、このような問題設定で GA を適用した場合、GA の性質上、ひとたび満足解が獲得されると、その周辺を集中的に探索し、最初に得られた満足解の近く、すなわち設計変数が極めて類似した多くの満足解が獲得されてしまうことがある。これは、類似した解はその評価値も近い、すなわち満足解の類似解がやはり満足解となることにもよる。そのような極端な類似解は、実用上意味を持たないことが多い。

GA において、設計変数の多様性を維持した探索を行うアルゴリズムは、古くから数多く提案されてきた [3][4][5]。これらの手法では、主に、局所解に陥ることを避けるために多様性を維持する機構が設けられており、その収束性を犠牲にしていることが多い。また、基本的には、満足解の獲得ではなく、評価値の高い解の獲得が目的となるため、満足解獲得後もその周辺を探索し、評価値を高めようとする。さらに獲得された満足解の多様性は保証されていない。つまり、これらの手法は、多様な満足解を効率よく獲

得するという目的には、必ずしも合致しない。

また、多目的最適化問題に対して GA を用いた場合には、評価値空間を広く満たすように、すなわち、評価値の上で多様な解の探索が行われる。しかしそのような探索においても、設計変数の多様性は考慮されず、評価値の上で多様な解が獲得される。一般的には、評価値が多様であれば、その設計変数も多様となることが多いが、少なくとも設計変数の多様性は保証されていない。

一方、複数の目的関数がある場合でも、各目的関数の最大化/最小化ではなく、全目的関数において目標値に近い値を求めるような問題であれば、その目標値からの誤差の総和を目的関数とすることにより、単目的最適化問題として扱うことができる。問題を単目的化することにより、評価値空間の多様性を確保することに充てられていた計算資源を、設計変数空間の多様性の確保に充てることができると考えられる。

また、実問題は多数制約付き最適化問題であることが多く、複数の制約を満たした実行可能解の獲得が求められる。ここで、制約の満足という考え方は、制約関数の値 ≤ 0 などのように、上述の目標値を求めるという問題設定と親和性が高い。そのため、上述の単目的化手法を制約に対しても用いることで、設計変数空間で多様な実行可能解 (満足解) を獲得することが可能であると考えられる。そこで本稿では、目的関数の単目的化手法を多数制約問題 [6] に適用するとともに、多様な満足解の獲得手法について検討する。

2. 目的関数の単目的化

2.1 多目的最適化問題

複数の目的関数がある場合でも、全ての目的関数が、特定の目標値に近づくことを目的としている場合には、目

¹ 名古屋大学

^{a)} maruyama@cmplx.cse.nagoya-u.ac.jp

標値からの誤差の総和を目的関数とすることにより、単目的化することができる。単目的化を行うことにより、評価値空間の多様性に充てられる計算資源を設計変数空間の多様性に充てることができる。

$$\min F = \sum_{i=1}^m |\hat{f}_i| \quad (1)$$

$$\hat{f}_i = \frac{\bar{f}_i}{\bar{f}_i^{\max}} \quad (i = 1, 2, \dots, m) \quad (2)$$

$$\bar{f}_i = \begin{cases} \text{Max}(|f_{ti} - f_i| - th_i, 0) & (a) \\ \text{Max}(f_{ti} - f_i, 0) & (b) \\ \text{Max}(f_i - f_{ti}, 0) & (c) \end{cases} \quad (3)$$

ここで f_i は目的関数 i の評価値、 $\bar{f}_i^{\max}$ は $\bar{f}_i$ の最大値であり、 th_i は許容誤差の閾値、 f_{ti} は f_i の目標値である。(3)式において、 f_i が f_{ti} からの誤差を許容値 th_i に抑えたい場合は (a) 式を選択し、 f_i が f_{ti} 以上であればよい場合は (b) 式、 f_i が f_{ti} 以下であればよい場合は (c) 式を選択する。 $F = 0$ となる解が満足解であり、各目的関数において誤差が許容される値以内に収まっていることを表す。

2.2 多数制約付き最適化問題

多数制約付き最適化問題においても同様の考え方で単目的化することができる。

$$\min F = \sum_{i=1}^m |\hat{f}_i| + \sum_{i=1}^l |\hat{g}_i| \quad (4)$$

$$\hat{g}_i = \frac{\bar{g}_i}{\bar{g}_i^{\max}} \quad (i = 1, 2, \dots, l) \quad (5)$$

ここで、 $\hat{f}_i$ は 2.1 節の (2) 式で与えられる。また、 $\bar{g}_i$ は制約 i の違反量、 $\bar{g}_i^{\max}$ は $\bar{g}_i$ の最大値である。多目的最適化問題の場合と同様に、 $F = 0$ となる解が実行可能解かつ満足解である。

3. 提案手法

多様な満足解を獲得するために、以下の 5 つの特徴を持つ手法を提案する。提案手法は、満足解を獲得しさえすればよいという問題を想定した手法であるため、満足解の評価値を良くすることは一切考慮しない。また、満足解の近傍に存在する満足解の類似解の獲得も目的としない。むしろ、満足解獲得後のこれらの探索を不要なものとし、他の満足解獲得のための探索資源に充てること、多様な満足解を効率よく獲得することを目的とする。提案手法のフロー (評価値の最小化の例) を図 1 に示す。

- 動的探索資源分配
 - 生成した個体の近傍個体に満足解がある場合は、その個体の評価を行わず、計算資源を他の領域の探索に充てる (= 満足解周辺は探索しない)。
- 近傍半径 (設計変数空間) による近傍個体の定義

- 設計変数空間で近傍半径を用いることで、設計変数空間における”近傍”関係を明示的に定義する。個体間の距離が近傍半径より小さい場合、互いの近傍個体となる。
- 満足解の設計変数空間における間隔の粒度の調節ができる。近傍半径を小さく設定すると細かい粒度での探索を行うことができる。すなわち、比較的類似した満足解が獲得される。逆に近傍半径を大きく設定すると荒い粒度での探索を行い、互いに離れた (類似していない) 満足解が獲得される。
- 近傍交叉
 - 近傍個体と交叉を行うことにより高い収束性が期待できる [7]。
- 設計変数空間における多様性確保機構
 - 設計変数空間内で近傍個体が多い個体は、子個体に淘汰される。
 - 生成された子個体 C の近傍個体数 $|C_n|$ が最大値 $n_{\max}$ を超えた場合、子個体の評価値 $f(C)$ が近傍個体内の最悪個体 (最も悪い (大きい/小さい) 評価値を持つ個体) C_n^{nad} の評価値 $f(C_n^{\text{nad}})$ より良ければ入れ替え、最悪個体情報を更新する。
 - 生成された子個体 C の近傍個体数 $|C_n|$ は最大値 $n_{\max}$ を超えていないが、最も近傍個体数が多い個体の近傍個体数 $|I_n^{\max}|$ が最大値 $n_{\max}$ を超えている場合、その中で最も評価値が悪い個体 $I_n^{\max}$ と生成された子個体 C を入れ替え、最悪個体情報を更新する。
- 逐次更新型
 - MOEA/D[8] などと同様、優秀な子個体が即座に親になることができるため、高い収束性が期待できる。

4. 実験

本稿では、3 つの実験を行った。全ての実験で、実問題に近い条件に基づいて作成された、多数制約付き 2 目的最適化ベンチマーク問題 [6] を用いた。この問題では、設計変数の数は 222 個、制約の数が 54 個であり、目的関数の数は 2 つである。多数制約問題 [9][10] では、全ての制約を充足する解を実行可能解と呼ぶ。また、本問題ではさらに、実行可能解の中で、評価関数の値が条件 (目標値以上、あるいは目標値以下) を満たしている個体を満足解とする。ここでは、目的関数の目標値は、 $f_1 = 3.0$ (以下)、 $f_2 = 34$ (以上) とした。この値は、ベンチマーク問題 [6] で紹介されている、initial design の目的関数の値 (人手により設計された解の評価値) となっている。

4.1 実験 1

4.1.1 問題設定

実験 1 では、2 章で述べた目的関数の単目的化の効果の検証を行う。制約付き多目的最適化問題をそのまま多目的

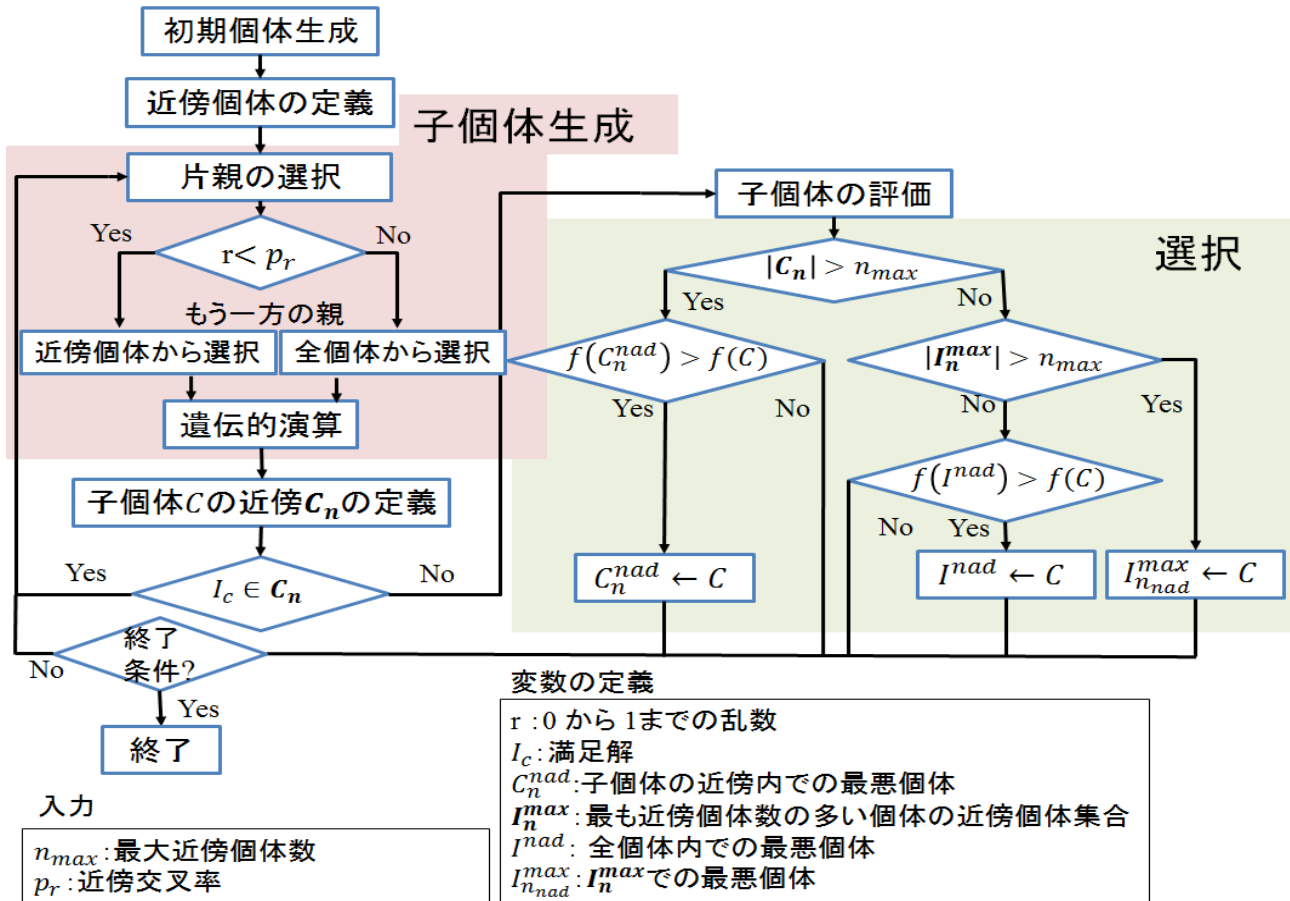


図 1 提案手法のフローチャート

最適化問題として CNSGA-II[11] を用いて探索を行った場合の結果と、単目的化手法を目的関数と制約に対して適用し、単目的最適化問題として提案手法を用いて探索を行った場合の結果を比較する。

4.1.2 実験条件

提案手法、CNSGA-II とともに、個体数 $N = 100$ 、評価回数 30,000 回とし、21 試行を行った。提案手法は、最大近傍個体数 15、近傍半径はマンハッタン距離で 22.2 で実験を行った。

4.1.3 実験結果

表 1 に、HV(Hyper Volume) の値を示す。また図 2 に、HV が中央値となる試行で獲得された満足解の評価値空間上での分布を示す。HV が中央値となる試行で獲得された満足解の数は CNSGA-II では 2482 個、提案手法では 5 つであった。表 1 と図 2 より、評価値空間上では、CNSGA-II の方が提案手法よりも圧倒的に多様で、評価値の上で優れた満足解を獲得できていることがわかる。これは、CNSGA-II が評価値空間を広く満たすように探索を行ったのに対し、提案手法では、評価値空間上の多様性を考慮せずに探索を行ったためであると考えられる。

次に、設計変数空間の多様性について確認する。表 2 に、獲得された満足解間の距離 (マンハッタン距離) の最小値、

最大値、平均を示す。また、獲得された満足解間の設計変数空間での距離を多次元尺度構成法 (Multi Dimensional Scaling: MDS)[12] を用いて可視化 (222 次元 → 2 次元) したものを図 3 に示す。これは、それぞれの手法ごとに MDS で 2 次元写像にした後、2 つの手法を、重心の位置が一致するようにプロットしたものである。表 2、図 3 を見ると、提案手法が CNSGA-II と比較して十分に満足解間の距離が大きく、設計変数空間上で多様な満足解が獲得できていることがわかる。実用上、例えば満足解が提案手法で設定した 22.2 離れたものでなければ意味がないとするならば、CNSGA-II では実質的に”1 つ”の満足解を獲得しただけとなる。この結果は、目的関数と制約関数を単目的化し、評価値空間上の多様性に充てられる計算資源を削減し、設計変数空間の多様性に充てることができたためであると考えられる。またこれにより、評価値空間上での多様性が必ずしも設計変数空間の多様性につながらないことが確認できた。

表 1 HV

	CNSGA-II	提案手法
最小	0.072	0.047
最大	0.140	0.057
平均	0.102	0.052

表 2 獲得された満足解間の距離

	CNSGA-II	提案手法
最小	0.1	44
最大	13.7	54.1
平均	5.2	49

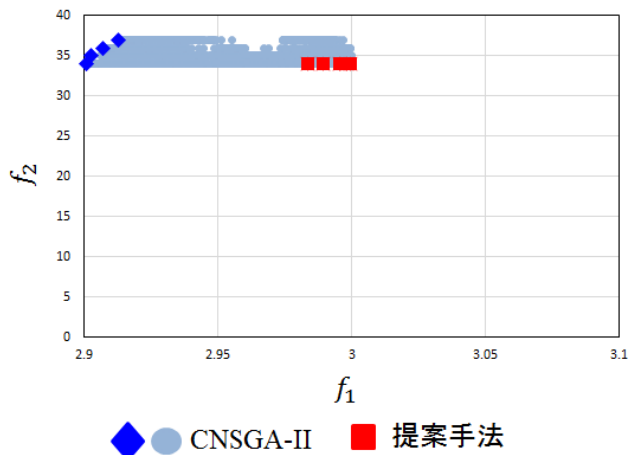


図 2 満足解の評価値空間での分布

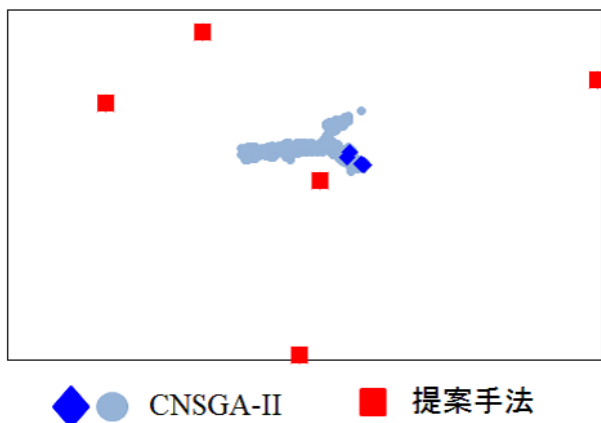


図 3 満足解の設計変数空間での分布

4.2 実験 2

4.2.1 問題設定

実験 2 では、3 章で提案した提案手法の探索性能について検討を行う。比較手法として Sequential Niching[3](以下、SN 法)を用いる。両手法ともに、単目的化手法を多数制約付き最適化問題の目的関数と制約に対して適用し、単目的最適化問題として探索を行い、探索性能の比較を行う。SN 法は、多峰性最適化問題で広く用いられる手法である。単目的化手法により、目的関数と制約関数が単目的化された問題で満足解を獲得するという問題設定への適用性が高く、提案手法と比較しやすいと考え、SN 法を比較手法とした。また、より効率的に満足解を獲得できるよう、SN 法に少し改良を加えた。

4.2.2 実験条件

提案手法、SN 法ともに、個体数 $N = 100$ 、評価回数 30,000 回とし、21 試行を行った。提案手法は、最大近傍個体数 15、近傍半径はマンハッタン距離で 22.2 で実験を行った。また SN 法の niche radius は、マンハッタン距離で 22.2 とした。

4.2.3 実験結果

表 3 に、獲得された満足解の個数の最小値、最大値、平均を示す。表 3 より、提案手法の方が、SN 法より平均で約 1 個の満足解を多く獲得できていることがわかる。次に図 4 に、満足解獲得までに要した評価回数の平均を示す。これは例えば、SN 法で 1 つ目の満足解が獲得された評価回数の平均が約 10,000 評価目であったことを表す。また、例えば満足解の個数が 4 の場合の評価回数は、満足解を 4 つ以上獲得することができた試行のみの平均値を表している。図 4 より、1 つ目の満足解を獲得するのは SN 法の方が早い、2 つ目で提案手法が追いつき、3 つ目以降では提案手法の方が少ない評価回数で満足解を獲得できていることがわかる。また、その差は、満足解の数が増加するほど大きくなっていく傾向が見てとれる。つまり、評価回数が増えれば、提案手法と SN 法で獲得される満足解の個数の差が大きくなると予想できる。

次に、獲得された満足解の設計変数空間上での多様性を確認する。表 4 に、HV が中央値となる試行で獲得された満足解間の距離 (マンハッタン距離) の最小値、最大値、平均を示す。満足解の個数は、SN 法では 4 個、提案手法では 5 個であった。表 4 より、SN 法、提案手法の両手法で獲得された満足解の設計変数空間上での多様性には大きな差がないことがわかる。これは、SN 法、提案手法の両手法での機構上、それぞれ 22.2 以上離れた満足解だけを獲得するように探索を行ったためであると考えられる。獲得された満足解の設計変数空間での多様性が同程度であれば、満足解の個数は多い方がよいといえる。

以上より、この問題において、提案手法は、多様な満足解を SN 法より多く獲得できていることが確認できた。これは、3 章で述べた提案手法の特徴が有効に働いたためであると考えられる。SN 法との違いを考慮すると、SN 法では、集団内の全個体で一斉に集中探索を行い、満足解が得られるとその周辺を探索しにくく (本稿では探索しない領域と) して、再度別の領域を全個体で探索するというメカニズムであるのに対し、提案手法では、各個体に対して近傍距離と近傍個体数を与えることで近傍個体を定義し、分散探索をしながらも満足解が得られた場合にはその周辺を探索しないというメカニズムが、複数の多様な満足解獲得に有効に働いたと考えられる。

表 3 獲得された満足解の個数

	SN 法	提案手法
最小	2	1
最大	4	6
平均	2.81	3.90

表 4 獲得された満足解間の距離

	SN 法	提案手法
最小	46.1	44
最大	51.9	54.1
平均	48.9	49

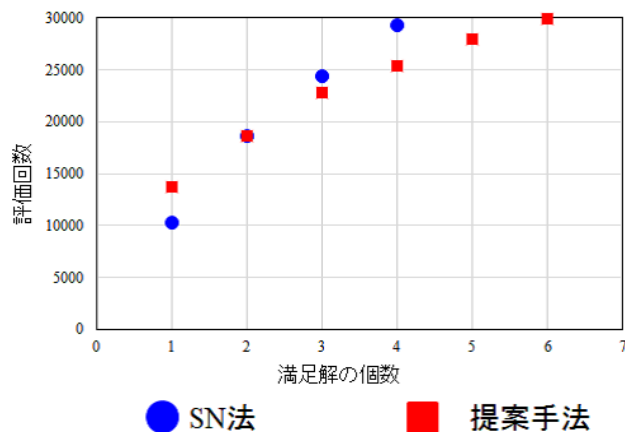


図 4 満足解獲得に要した評価回数

4.3 実験 3

4.3.1 問題設定

実験 3 では、比較手法として島モデル [5] を用いる。島モデルでは、母集団をいくつかの「島」と呼ばれるサブ集団に分割して探索を行う。各島における解の進化は基本的に他の島から独立して行われるが、数世代ごとに島と島の間で移住と呼ばれる解の交換が行われ、島間の情報交換がなされる。提案手法も近傍半径を用いて近傍個体を定義することにより、複数のグループ探索を行っており、島モデルとの類似性が高いと考え比較対象とした。両手法ともに、単目的最適化問題として探索を行い、探索性能の比較を行う。

4.3.2 実験条件

提案手法、島モデルともに、個体数 $N = 100$ 、評価回数 30,000 回とし、21 試行を行った。提案手法は、最大近傍個体数 15、近傍半径はマンハッタン距離で 4.4, 8.8, 22.2 で実験を行った。また島モデルの島数は 5, 10, 15 とし、移住は行わなかった。これは、予備実験において、島モデルで移住を行うことで、結果的に得られた満足解が島間で類似してしまい、多様な満足解が得られなかったことによる。

4.3.3 実験結果

表 5 に、島モデルで探索を行った場合に満足解獲得に成功した島の数と満足解の総数の 21 試行平均の値を示す。表 5 より、島数の数が多いほど獲得できる満足解の個数が

減少していることがわかる。これは、島数を多くすると島に所属する個体が少なくなるため、満足解まで到達できなかった島が増えたためであると考えられる。

次に、5 島で探索を行った場合について、21 試行中、獲得された満足解で算出した HV 値が中央値となる試行に注目する。この試行では、5 島中 3 島で満足解を獲得することに成功した。その 3 島 (島 1-島 3 とする) について、満足解の設計変数上の距離の最大値、最小値、平均を表 6 示す。表 6 より、島内の満足解間の距離は小さく、極めて類似した満足解が獲得されていることがわかる。これは、島モデルの性質上、ある島で満足解が獲得されると、その島では満足解周辺を集中的に探索し、類似した満足解が獲得されたためであると考えられる。実用上は、島内の満足解は 1 つの満足解とみなされると考えられるため、この試行で得られた実質的な満足解の数は 3 つ (=満足解を獲得した島の数) である。注目している試行で、満足解獲得後に満足解周辺の探索に充てられてた計算資源は、3,143 評価/30,000 評価であった。これは、提案手法において、不要として他の満足解獲得のために充てられる計算資源である。

また、島モデルには、明示的に島間の距離を維持する機構がないため、島間の距離が小さくなり、設計変数空間の多様性が維持されないことも考えられる (実際に移住を行うことでその傾向が見られた)。そこで表 7 に、満足解が獲得できた島 (これを 1 つの満足解として扱う。つまり、満足解) 間の距離の最大値、最小値、平均の 21 試行平均を示す。島間の距離は、ある島で獲得された満足解の重心をその島の代表点として算出した。表 7 より、島間の距離が十分に大きいことがわかる。また同時に、島の数が増えるほど満足解間の距離に与える影響は小さいことがわかる。これにより、母集団を分割し、島間で情報交換 (移住) を行わなければ、島間はマンハッタン距離で 50 ほど離れている。しかしこれは、島モデルでは、多様な満足解を獲得することは可能であるが、島間、すなわち満足解間の距離を、明示的に調整できないことを表しており、満足解の粒度の調節は困難であると考えられる。

表 8 に、提案手法で探索を行った場合に獲得した満足解の個数の 21 試行平均を示す。また表 9 に、獲得された満足解間の距離の最小値、最大値、平均の 21 試行平均を示す。表 8 より、近傍半径が小さいほど、満足解の個数が増加していることがわかる。また、表 9 より、近傍半径を調節することで、設定した近傍半径通りとはならないものの、満足解の間隔の粒度の調節ができていくことがわかる。

以上により、島モデルでも、移住を行わないことで多様な満足解を獲得することができることを確認した。しかし、島モデルでは、島間の距離を明示的に与えるわけではないため、満足解の間隔の粒度の調節は困難であるのに対し、提案手法では、近傍半径を用いることで、粒度の調節ができることを示した。

表 5 満足解獲得に成功した島数と満足解の総数

探索島数	5 島	10 島	15 島
満足解を獲得した島数	4.0	2.6	1.3
満足解の総数	80.0	25.7	8.4

表 6 島内の満足解間の距離

	島 1	島 2	島 3
最小	0.05	0.05	0.05
最大	2.05	1.80	2.30
平均	0.69	0.80	0.97

表 7 島間の距離

	5 島	10 島	15 島
最小	42.42	47.27	45.53
最大	52.18	50.69	48.26
平均	48.15	49.04	47.05

表 8 獲得した満足解の数

近傍距離	4.4	8.8	22.2
満足解の数	14.24	4.95	3.90

表 9 獲得された満足解間の距離

	4.4	8.8	22.2
最小	5.03	16.66	43.69
最大	25.05	38.69	50.60
平均	14.78	31.14	47.54

5. まとめ

本稿では、目的関数の単目的化手法について説明し、単目的最適化問題において、多様な満足解を獲得するための手法を提案した。提案手法の有効性を示すために、多数制約付き最適化問題を用いて 3 つの実験を行った。実験 1 では、単目的化の効果の検証を行うため、多目的最適化問題のままで CNSGA-II を用いて探索を行った結果と、目的関数と制約を単目的化し、単目的最適化問題として提案手法を用いて探索を行った場合の結果の比較を行った。CNSGA-II では、評価値空間上では多様な満足解が獲得できるものの、設計変数空間上では偏った満足解が獲得されること、提案手法では、設計変数空間上で多様な満足解を獲得できることを示した。また、評価値空間上での多様性が必ずしも設計変数空間の多様性につながらないことを示した。実験 2、実験 3 では、提案手法の探索性能についての検討を行うため、従来の設計変数の多様性を維持する手法との比較を行った。実験 2 では、Sequential Niching(SN 法)と提案手法の比較を行った。提案手法の方が、多様な満足解を SN 法より効率よく獲得できることを示した。特に獲得する満足解の数が多いとき、提案手法の方がより多くの満足解を獲得できる傾向があることを示した。実験 3 では、島モデルと提案手法の比較を行った。提案手法、島モデルともに多様な満足解を獲得できることを確認した。島モデルでは、満足解の間隔の粒度の調節が困難であるのに

対し、提案手法では近傍半径を調節することで粒度の調節ができることを示した。また、島モデルでは満足解を獲得した島で満足解周辺の探索を続け、多様な満足解の獲得という視点では、効率が悪い探索を行っていることを示した。

今後は、提案手法に対し、より実应用到に適した手法になるように改善を行っていく予定である。

参考文献

- [1] D.Dasgupta, Z.Michalewicz, “Evolutionary algorithms in engineering applications,” Springer-Verlag Berlin Heidelberg, 1997
- [2] C.A.C.Coello, G.B.Lamont, “Applications of multi-objective evolutionary algorithms,” World Scientific, 2004
- [3] D.Beasley, D.R.Bull, R.R.Martin, “A sequential niche technique for multimodal function optimization” Evolutionary Computation, 1(2), pp. 101-125, 1993
- [4] S.W.Mahfoud, “Niching methods for genetic algorithms.” Urbana 51.95001:62-94,1995
- [5] D.Whitley, R.Soraya, H.Robert B, “Island model genetic algorithms and linearly separable problems.”,In: AISB International Workshop on Evolutionary Computing. Springer, Berlin, Heidelberg, 1997. p. 109-125.
- [6] 小平剛央, 鉦持寛正, 大山聖, 立川智章, 応答曲面法を用いた複数車種の同時最適化ベンチマーク問題の提案, 進化計算学会論文誌, Vol8, No.1, p.11-21, 2017 <http://ladse.eng.isas.jaxa.jp/benchmark/index.html>
- [7] S.Kikuchi, T.Suzuki, “The Effect of Neighborhood Crossover in Evolutionary Search Methods for Landscape Photograph Geocoding Support,” IPSJ SIG Technical Report, Vol. 2010-FI-98, No. 10, 2010
- [8] Q.Zhang, H.Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” IEEE Transactions on Evolutionary Computation, Vol. 11, No. 6, pp. 712-731, 2007
- [9] K.Deb, “An efficient constraint handling method for genetic algorithms,” Computer Methods in Applied Mechanics and Engineering, Vol. 186, No. 2, pp. 311-338, 2000
- [10] E.Mezura-Montes, “Constraint-Handling in Evolutionary Optimization,” Springer, 2009
- [11] K.Deb, S.Pratap, T.Meyarivan, “A fast elitist multiobjective genetic algorithm: NSGA-II” IEEE transactions on evolutionary computation, Vol.6, No.2, pp182-197, 2002
- [12] 斎藤 堯幸, “多次元尺度構成法,” 朝倉書店, 初版, 1980