

設計変数間の依存関係を反映したハイブリッドメタ戦略による 多峰性問題の整数最適化[†]

長谷川 浩 志*・吉 川 向**・上 原 広 靖***・川 面 恵 司****

ABSTRACT In this paper, Authors proposed a hybrid meta-heuristic method for the integer optimization of multi-peak problems to support the engineering field. For the search ability of the optimal process, the proposed method did not define a traditional combination, i.e., combining global search ability and local search ability, but newly defined the combination of multi search ability (MSA), local search ability (LSA) and neighborhood optimal search ability (NOSA). For NOSA, this proposed method introduced direct search process which self-changes design variables with reflecting recognition of dependence relation among design variables automatically. The proposed method was applied to some benchmark functions to estimate its performance, and was confirmed that it improved amount of computational cost and convergence up to the optimal solution.

1. はじめに

近年、デジタルエンジニアリングは、製品の競争力をつけるための有効な技術として考えられている。特に、最適設計は、デジタルエンジニアリングにとって最も重要な技術といえる。工学分野における最適設計の設計変数は、機械構造の寸法、材質、取り付け位置等が挙げられる。設計変数の値は、一般に連続的な値ではなく、離散的な値をとるものが多く、例えば1.12(図面の寸法精度の指定により小数点以下2桁までを考慮)、400(鋼材の規格番号, SS400)、0(機器が配置されない)、1(機器が配置される)などが考えられる。これらの設計変数の値は、112(1.12×100)、400、1、0として取り扱うことができる。すなわち、設計変数は、整数にて定義された最適化問題となる。さらに、製品開発のための設計空間は、製品に対する様々な要求や

要望のために、複雑さを増しており、その景観は、多峰性を有する空間といえる。

このような多峰性最適化問題に対する有力なアプローチとして、メタ戦略が注目されている。最も有名なメタ戦略には、Genetic Algorithm (GA)^{1, 2)}があり、様々な多峰性問題や工学問題に適用され、その有効性が報告されている。しかしながら、GAは、安定的に最適解へ収束させるために、膨大な計算コストを必要とする。このため、遺伝的操作の研究や並列化の研究が盛んに行なわれている。

Digalakis と Margaritis は、GA に対して著名なベンチマーク関数を用いた数値実験を行い、その結果からGAの集団サイズを200~250個体の規模にすることや交差確率を0.70~0.75にすることを提唱している³⁾。佐久間と小林は、real-coded GA に対して2通りのタイプの交差、UNDX-*m*とEDXを組み合わせた遺伝的操作の手法を提案し、複数のベンチマーク関数を通してその効果を報告している⁴⁾。並列化の研究では、例えば、Li と Kirley が⁵⁾パーコレーションの原理から導かれたfine-grained parallel GAを提案し、その手法をベンチマーク関数にて評価している⁵⁾。筆者らも、GAによる機械構造物の位相最適化^{6~8)}、General-purposed Optimization Engine (GOE)の開発と最適設計への適用^{9, 10)}を通して、GAの有効性を確認している。さらに、最適解への収束性向上と計算コストの低減のために、大域的探索と局所的探索をうまく組み合わせたハイブリッド

The Hybrid Meta-heuristics by Reflecting Recognition of Dependence Relation among Design Variables for The Integer Optimization of Multi-peak Problems. By Hiroshi Hasegawa (Shibaura Institute of Technology), Mukau Yoshikawa (Toyota Technical Development Corporation), Hiroyasu Uehara (PENTAX Corporation) and Keishi Kawamo (Techno Intelligence Corporation).

* 芝浦工業大学

** トヨタテクニカルディベロップメント(株)

*** ペンタックス(株)

**** (株)テクノインテリジェンス

† 2005年7月14日受付 2006年1月10日再受付

メタ戦略を導入することが重要であることを示している。

このハイブリッドメタ戦略は、現在、多くの戦略が提案されており、最も有名な戦略の1つが大域的探索を行うGAと局所的探索を行うSimulated Annealing(SA)¹¹⁾を組み合わせたものである。MahfoudとGoldbergにより1992年に提案されたParallel Recombinative SA (PRSA)は、GAの選択処理にSAを利用する戦略であり、先駆的な研究である¹²⁾。河田らは、SAのメトロポリスループをGAのエリート戦略内にて利用する戦略を提案し、その戦略を様々なベンチマーク問題に適用し、その有効性を報告している^{10, 13)}。廣安らは、Parallel SA using Genetic Crossover (PSA/GAc)を提案し¹⁴⁾、三木らは、Parallel SA with adaptive neighborhood range determined by GA (PSA/ANGA)を提案している¹⁵⁾。PSA/GAcとPSA/ANGAはSAを主体とした戦略であり、多点探索のために並列SAを適用し、その有効性を報告している。GAとSAのハイブリッドメタ戦略は、確率的探索手法をベースにしているため、局所的探索だけでなく大域的探索についても優れた探索手法と言われている。しかしながら、上原のParallel SA with Selection (PSAS)によると、多次元の多峰性問題で、設計変数間に依存関係のあるベンチマーク関数として著名なGriewank (GR)関数では、大域的探索についてはうまく作用しているが、局所的探索がうまく作用しない場合があることを指摘している¹⁶⁾。これは、ハイブリッドメタ戦略が確率的探索手法を採用していることから、設計変数値の更新に確率的変動を伴い、最適解近傍に限定した局所的探索がうまく行えないことが原因と考えられる。この原因に対する1つの解決方法として、例えば、OngとKeaneは、Meta-Lamarckian Learning in Memetic Algorithmsにて、大域的探索を担うGAと6通りの局所的探索手法を用意し、対象とする最適化問題に対して、どの局所的探索手法を採用すべきか設計者に判断を委ねるプロセスをMeta-Lamarckianの学習プロセスに導入することで、多次元の多峰性問題に対する探索能力を大幅に改善する方法を提案している¹⁷⁾。しかしながら、本手法は、設計者が選択した局所的探索手法に大きく依存するため、設計者の能力次第で計算コストや最適解への収束性が異なる。これでは、安定的に最適解が得られる手法とはいえない。なお、設計者の判断を含めないMA-S2では、10次元のGR関数にて最適解が得られていない。さらに、文献3, 5)においても多次元のGR関数については、安定的に最適解を導き出すに至っていない。

本論文では、工学分野を支援するために、多峰性の整数最適化問題に対する新たなハイブリッドメタ戦略の手法を提案する。この手法は、確率的探索手法をベースとしたハイブリッドメタ戦略に対して、従来苦手としていた最適解近傍の探索のために、新たに設計変数間の依存関係を自動的に認識・反映する決定論的手法を導入し、局所的探索能力を改善するものである。筆者らは、この提案手法をGR関数を含む4つのベンチマーク問題に適用し、その有効性が得られたので本論文にて報告する。

2. メタ戦略について

2.1 最適化問題の定式化

ここでは、最適化問題を定式化する。設計変数、目的関数、ペナルティ関数、制約条件、設計変数の範囲制約を以下のように定義する。

$$\text{設計変数;} X_i = (x_i^1, \dots, x_i^n) \quad (1)$$

$$\text{目的関数;} F_i(X_i) = f_i(X_i) + \gamma P_i(X_i) \rightarrow \text{Max} \quad (2)$$

ペナルティ関数;

$$P_i(X_i) = \sum_{j=1}^M \left\{ \text{Max} \left(\frac{\text{Weight}_j}{\text{Scale}_j} g_{i,j}(X_i), 0 \right) \right\} \quad (3)$$

$$\text{制約条件;} g_{i,j}(X_i) \leq 0 \quad j = 1, 2, \dots, M \quad (4)$$

$$\text{範囲制約;} X^{LB} \leq X_i \leq X^{UB} \quad (5)$$

ここで、 i は繰り返し番号、 n は設計変数の番号、 γ はペナルティ係数、 M は制約条件数、 Weight_j は重み、 Scale_j はスケール、 X^{LB} は設計変数の下限値、 X^{UB} は設計変数の上限値である。

2.2 提案するハイブリッドメタ戦略

現在、提案されているGAやハイブリッドメタ戦略は、先に述べたように、GR関数に代表される設計変数間に依存関係がある多峰性問題に関しては、最適解近傍に存在する満足解までは得られるが、安定的に最適解を導き出すことが困難である。この問題に対して、筆者らは、新たにハイブリッドメタ戦略を使った多峰性の整数最適化手法を提案する。この手法は、一般的なハイブリッドメタ戦略にて用いられている大域的探索能力と局所的探索能力の組み合わせとするのではなく、最適解近傍の探索能力を改善すべく、新たに複合探索能力、局所的探索能力、最適解近傍の探索能力の3つの組み合わせであると再定義する。また、新たに導入された最適解近傍の探索能力として、局所的探索能力を担うメタ戦略内にて設計変数間の依存関係を試行錯誤により認識させ、その認識した知識を、設計変数

の自己更新プロセスに反映させる直接探索プロセスを提案する。本手法の概念プロセスを図1に示す。本手法は、最適化プロセス全体をコントロールするためにGAを用いる。設計変数値を自己更新するプロセスには、SAと新たに提案した直接探索プロセスを用いる。この提案手法の特徴を以下に示す。

- 1) 複合探索能力 (Multi search ability, MSA) : 広範囲の設計空間を対象とし、複合探索を行うためにGAを用いる。これは、複数の探索点(個体)を発生させ、それらの探索点の適応度を評価し、遺伝的操作を行い優れた解、すなわち最適解に近い近傍点を探索する。
- 2) 局所的探索能力 (Local search ability, LSA) : 設計空間内に局所的最適解が複数存在する問題を想定しているため、確率的な探索手法を用いることにより局所的最適解に陥ることを防ぐ。LSAは、SAのメトロポリスループの受理判定によって設計変数の自己更新プロセスを実現する。
- 3) 最適解近傍の探索能力 (Neighborhood optimal search ability, NOSA) : NOSAは、設計変数の自己更新プロセスにて直接探索プロセスを用いる。直接探索プロセスは、ステップ幅 SP_i により元の探索点 X_i から新しい近傍点 X_{i+1} を生成する。近傍点 X_{i+1} の生成には、LSAにて試行錯誤により記録された設計変数間の依存関係を認識・反映する。もし、 $F_i(X_i) < F_{i+1}(X_{i+1})$ ならば、 X_i を X_{i+1} へ推移させる。それ以外の場合には、 $X_{i+1} = X_i$ とする。このプロセスを繰り返すことで探索点は最適解に近づく。

本手法の探索プロセスについて、図1を用いて述べ

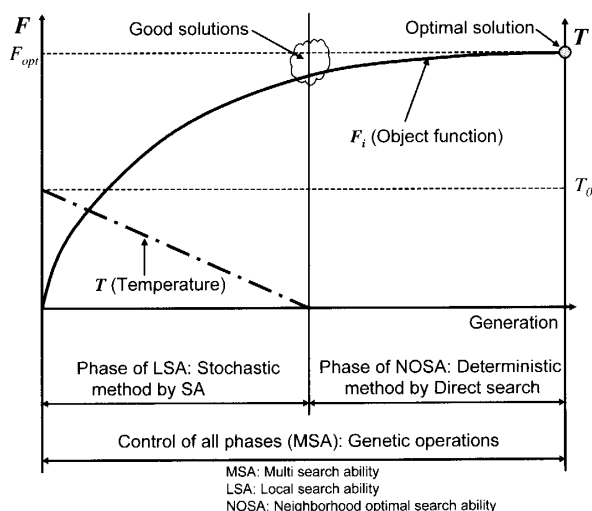


図1 ハイブリッドメタ戦略の概念プロセス

る。本プロセスは、MSAの戦略として、プロセス全体をGAにより多点探索する。つぎに、各個体の染色体にコーディングされた設計変数に対して、LSAの自己更新プロセスを実施する。図1に示すように、MSAとLSAのハイブリッド戦略により満足解を探索する。さらに、SAの冷却プロセスの温度が最小温度に到達した時点で、探索手法をLSAからNOSAの直接探索プロセスに切り替える。NOSAの直接探索プロセスにより最適解近傍の探索を実施する。このとき、LSAにて試行錯誤により記録された設計変数間の依存関係を本プロセスにて認識・反映させる。最適解近傍領域では、MSAとNOSAのハイブリッド戦略により探索され、最終的に最適解が得られる。

2.3 MSAのためのGA

最適化問題の設計変数は、図2のように染色体に設計変数の整数値を用いて表現される。本論文では、GAの個体数を20個体とし、選択・淘汰にはルーレット戦略を用いる。なお、選択確率は $P_s = 1.0$ 、交差については2つの個体をランダムに親として選択し、子孫を生成する。交差は、図2のように設計変数間でのみ行う。交差確率は $p_c = 0.2$ とする。最良な個体を次世代に残すためにエリート戦略を用いる。ただし、突然変異については行わない。

2.4 LSAのためのSA

LSAのフェーズでは、設計変数の自己更新プロセスにSAを用いる。SAのプロセスは、生成・受理判定・冷却の3つのプロセスである。SAの一般的なプロセスを図3のステート図に示す。生成プロセスは、現在の探索点 X_i から、確率的振る舞いにより新たな近傍点 X_{i+1} を生成する。もし、その近傍点が受理判定に合格すれば、 X_i は X_{i+1} の位置に推移する。却下された場合には、その探索位置をそのままとし、 $X_{i+1} = X_i$ とする。このプロセスは、メトロポリスループにて繰り返される。推移確率をコントロールする温度は、GAのプロセスを通じて、冷却スケジュールに基づき下げられる。推

	x_1	x_2	x_3	x_n
Individual A	512	512	512	509
Individual B	512	516	512	512
Crossover				
Individual A	512	516	512	509
Individual B	512	512	512	512

図2 交差

移確率は、温度を下げることで小さくなり、探索点 X_i は最適解へ近づいてゆく。

2.4.1 生成

生成プロセスは、次式によって現在の探索点 X_i から、新たな近傍点 X_{i+1} を生成する。

$$X_{i+1} = \begin{cases} X_i + \text{Sign}(y_i) \cdot SP_0 & \text{if } GP_i = 0 \\ X_i + \text{Sign}(y_i) \cdot GP_i \cdot MR \cdot NR_i & \text{if } 0 < GP_i \leq 1 \end{cases} \quad (6)$$

ここで、 y は $y_i \in [-1, 1]$ の一様乱数、 SP_0 は最小ステップ幅 $SP_0 = 1$ 、 GP_i は生成確率関数、 MR は設計変数の最大範囲、 NR_i は近傍率である。式(6)で使われる生成確率関数を次に示す。

$$GP_i \in \begin{cases} 0 & \text{if } 0 \leq U_i < 0.4 \\ [0, 1] & \text{if } 0.4 < U_i \leq 0.7 \\ 1 & \text{if } 0.7 < U_i \leq 1 \end{cases} \quad (7)$$

ここで、 U は $U_i \in [0, 1]$ の一様乱数である。図4に示したように、この関数は式(6)を使うことで、大まかな探索(最大ステップ幅を用いた場合)と詳細な探索(最小ステップ幅を用いた場合)を行うことができる。

2.4.2 近傍幅

ここでは、冷却スケジュールを用いて近傍幅をコントロールする方法を導入する。近傍幅は、温度が低下

するに伴い小さくなってゆく。つまり、最適解に近づくにつれて温度が低下し、近傍幅も小さくなる。結果として、局所的探索を詳細に実行することができる。近傍幅をコントロールする式を式(8)に示す。

$$NR_i = \left\{ 1 - \alpha \left(1 - \frac{T_i}{T_0} \right) \right\} NR_0 \quad (8)$$

ここで、 T_i は温度、 T_0 は初期温度、 α は温度係数、 NR_0 は初期近傍率である。例として、 $T_0 = 10.0$ 、 $\alpha = 0.8$ 、 $NR_0 = 0.1$ 、 $\alpha = 0.99$ (式(11))としたときの、式(8)を用いた近傍率を図5に示す。

2.4.3 独立と依存の関係

一般的な最適化アルゴリズムは、設計変数がそれぞれ独立関係にあるものとして定式化されているものが多い。しかし、本論文で取り扱う最適化問題は設計変数が依存関係にある多峰性問題も想定している。このため、新たに設計変数の独立・依存関係を反映する手法を考案する。この手法は、式(9)により設計変数値を変更する設計変数の数をランダムに決定する。つまり、 $SD_i = 1$ ならば独立関係であると定義し、一つの任意の設計変数を式(6)により変更する。また、 $SD_i = 2$ ならば依存関係であると定義し、二つの任意の設計変数を同時に式(6)により変える。

$$SD_i = \begin{cases} 1 & \text{if } 0 \leq U_i < 0.5 \\ 2 & \text{if } 0.5 \leq U_i < 1.0 \end{cases} \quad (9)$$

ここで、 SD_i は選択される設計変数の数、 U は $U_i \in [0, 1]$ の一様乱数である。

2.4.4 受理判定

受理判定は、現在の探索点 X_i から新しい近傍点 X_{i+1} に推移させるか否かを判断するプロセスである。受理判定は次式で定義される。

$$A(\Delta E, T_i) = \begin{cases} 1 & \text{if } E_{i+1} \geq E_i \\ e^{-\Delta E / T_i} & \text{otherwise} \end{cases} \quad (10)$$

ここで、 E_i は目的関数 $F_i(X_i)$ 、 ΔE はエネルギーの差分 $\Delta E = E_i - E_{i+1}$ 、 T_i は温度である。最適化の初期プロセ

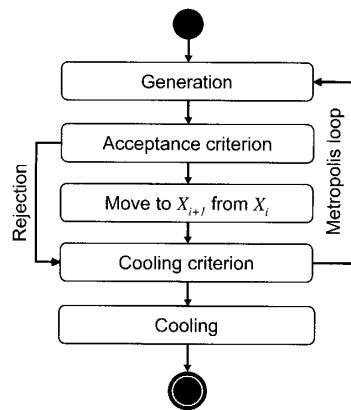


図3 SAのステート図

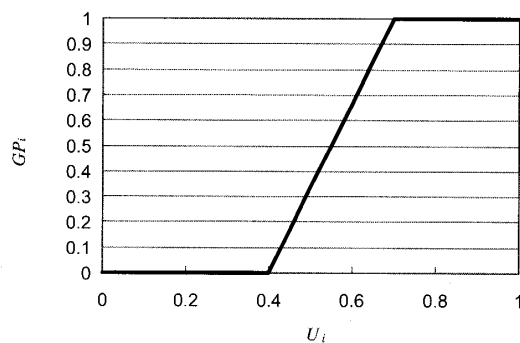


図4 生成確率関数

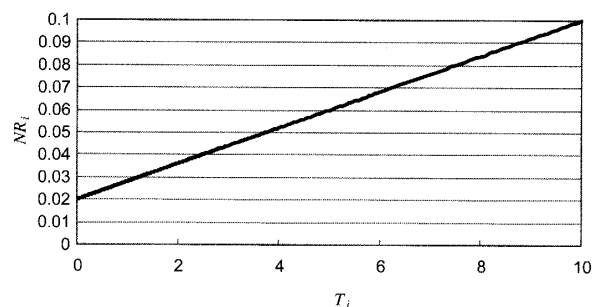


図5 近傍率と温度の関係

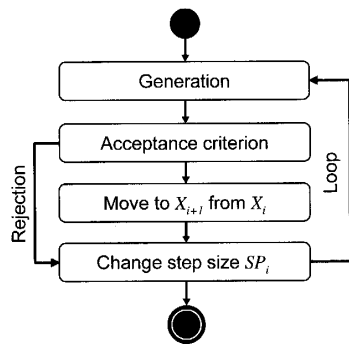


図6 DSPのステート図

スでは、改悪方向への推移を許容するが後半のプロセスでは冷却スケジュールにより温度が小さくなるため、改悪方向への推移は却下される。

2.4.5 冷却スケジュール

SAの温度は、式(11)の冷却スケジュールに則ってGAの世代ごとに温度を変更する。

$$T_{i+1} = \gamma T_i \quad (11)$$

ここで、 γ は冷却率、 T_i は温度である。本論文では、初期温度 $T_0 = 10.0$ 、冷却率 $\gamma = 0.99$ 、終点温度 $T_{end} = 0.1$ として冷却スケジュールを作成する。

2.5 NOSAの直接探索

NOSAのフェーズでは、設計変数の自己変更プロセスに決定論的手法をベースとした直接探索プロセス (Direct search process, DSP)を導入する。DSPのステート図を図6に示す。このプロセスは生成、受理判定、ステップ幅変更の3つのプロセスにより構成される。

2.5.1 生成

生成プロセスでは、次式によって現在の探索点 X_i から新しい近傍点 X_{i+1} を生成する。

$$X_{i+1} = X_i + SP_i \quad (12)$$

ステップ幅 SP_i は、図7に示すアクティビティ図により変更される。例えば、 $MR = 1024$ 、 $NR_0 = 0.01$ とした場合のステップ幅 SP_i とその繰り返し数を図8に示す。

2.5.2 独立と依存の関係

設計変数の独立・依存の関係、つまり $SD_i = 1$ 、 $SD_i = 2$ は、それぞれの繰り返しで式(13)を使って更新される。式(13)を次に示す。

$$SD_i = \begin{cases} 1 & \text{if } 0 \leq U_i < AD_1 / (AD_1 + AD_2) \\ 2 & \text{if } AD_1 / (AD_1 + AD_2) \leq U_i < 1.0 \end{cases} \quad (13)$$

ここで、 U は $U_i \in [0, 1]$ の一様乱数、 AD_1 は受理判定にて受理された $SD_i = 1$ の総数、 AD_2 は受理判定にて受

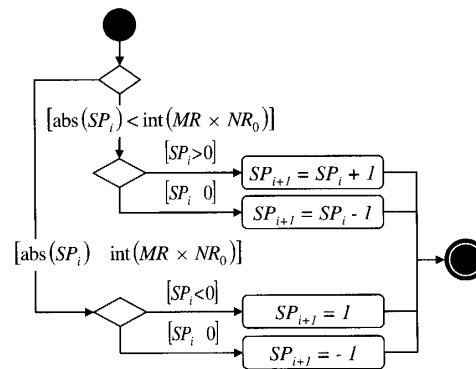
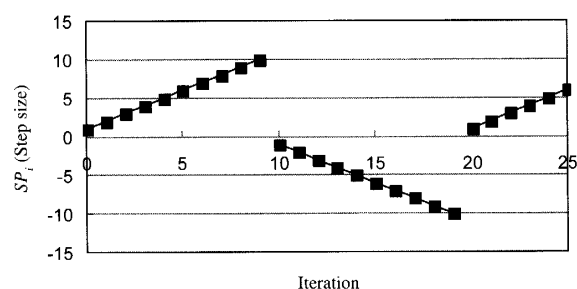
図7 SP_i を変更するアクティビティ図

図8 ステップ幅とその繰り返し数(25回まで)

理された $SD_i = 2$ の総数である。LSAのフェーズでは、 SD_i は式(9)によってランダムに決定され、受理された SD_i が記録される。この記録は、NOSAのフェーズにて学習された知識として式(13)にて反映される。DSPでは、新しい近傍点 X_{i+1} の適応度 $F_{i+1}(X_{i+1})$ が現在の探索点 X_i の適応度 $F_i(X_i)$ よりも高いとき、現在の探索点 X_i から新しい近傍点 X_{i+1} への推移を受理する。このプロセスが繰り返され、探索点は最適解に近づいていく。

3. 数値実験

3.1 ベンチマーク関数

ここでは、RA (Rastrigin)、RI (Ridge)、GR (Griewank)、AC (Ackley) の4種類のベンチマーク関数 (図9は2変数の場合) を用い、提案するハイブリッドメタ戦略の最適解への収束性を評価する。なお、設計変数の数は20変数とする。4つのベンチマーク関数を次式に示す。

$$f_i^{RA} = 10n + \sum_{k=1}^n \{ (x_i^k)^2 - 10 \cos(2\pi x_i^k) \}, \quad f_{\min}^{RA} = 0, \quad X_i = 0, \quad n = 20 \quad (14)$$

$$f_i^{RI} = \sum_{k=1}^n \left(\sum_{l=1}^l x_i^k \right)^2, \quad f_{\min}^{RI} = 0, \quad X_i = 0, \quad n = 20 \quad (15)$$

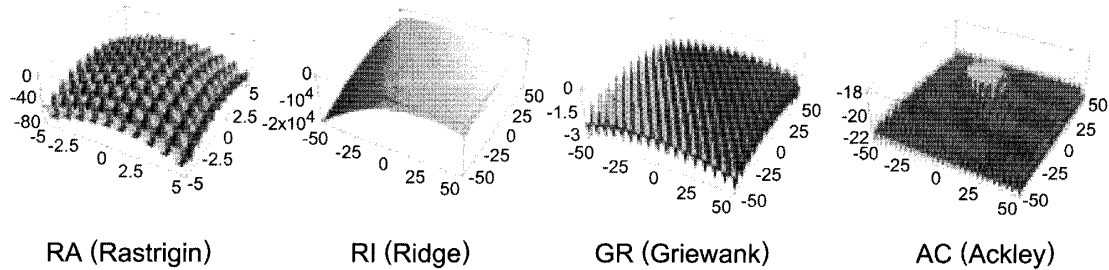
図9 ベンチマーク関数(2変数, $n=2$)

表1 ベンチマーク関数の特徴

Benchmark functions	Dependence relation among design variables	Multi-peak	Steep
RA	No	Yes	Average
RI	Yes	No	Average
GR	Yes	Yes	Small
AC	No	Yes	Average

表2 各種パラメータの設定

Functions	Range of design variables	MR	NR ₀	β
RA	$-5.12 \leq X_i \leq 5.12$	1024	0.1	0.01
RI	$-51.2 \leq X_i \leq 51.2$	1024	0.01	0.1
GR	$-512 \leq X_i \leq 512$	1024	0.01	1.0
AC	$-51.2 \leq X_i \leq 51.2$	1024	0.1	0.1

$$f_i^{GR} = 1 + \sum_{k=1}^n \frac{(x_i^k)^2}{4000} - \prod_{k=1}^n \cos\left(\frac{x_i^k}{\sqrt{k}}\right),$$

$$f_{min}^{GR} = 0, \quad X_i = 0, \quad n = 20 \quad (16)$$

$$f_i^{AC} = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{k=1}^n (x_i^k)^2}\right) - \exp\left(\frac{1}{n} \sum_{k=1}^n \cos 2\pi x_i^k\right) + 20 + e,$$

$$f_{min}^{AC} = 0, \quad X_i = 0, \quad n = 20 \quad (17)$$

各ベンチマーク関数の特徴を表1に示す。表中には、設計変数間の依存関係、問題空間の多峰性、Steep、すなわち最大ピーク値と最小ピーク値の差を示す。本実験に用いるベンチマーク関数は、設計空間が多峰性であるか、単峰性でも設計変数間に依存関係がある関数である。なお、RA関数は、多峰性問題のベンチマーク関数として著名な関数であり、多数の局所的最適解を持つことで特にGAにとって最適化が困難な問題として知られている。RI関数は、単峰性関数であるが設計変数間に依存関係がある問題である。GR関数は、設計変数間に依存関係があり、steepが小さく、多数の局所的最適解を持つ問題である。よって、大変複雑な特性を有している。AC関数は、最適解近傍領域の設計空間を示す包絡曲面(峰の頂点を含む)の曲率が小さい問題であり、多峰性である。

これらのベンチマーク関数は、それぞれ異なった特徴を有する関数であり、全ての関数にて同時に最適解を探索することは難しい。その中でも、多次元のGR関

数は、安定的に最適解を導き出すことが困難な関数である。しかしながら、本提案手法では、設計変数間の依存関係を認識・反映することで、この困難さを解決するアプローチを提案していることから、本手法の評価のために、GR関数の次元を一般的に用いられている10次元から20次元へと増やし、難易度を上げている。

3.2 最適化問題の定式化

ベンチマーク関数に対する最適化問題の設計変数、目的関数、範囲制約は次式にて定義される。

$$\text{設計変数; } X_i = \beta \cdot (X_i^{\text{Gene}} - 512) \quad (18)$$

$$\text{目的関数; } F_i(X_i) = -f_i(X_i) \rightarrow \max \quad (19)$$

$$\text{範囲制約; } 0 \leq X_i^{\text{Gene}} \leq 1024 \quad (20)$$

ここで、 β はコーディング係数、 X_i^{Gene} は遺伝子にコーディングされた設計変数値である。提案するハイブリッドメタ戦略では、探索点が最適解に到達するか世代数が指定した最終世代数に到達した場合に探索プロセスを終了する。LSAとNOSAにおけるメトロポリスループおよびループの数は2ループとする。ベンチマーク関数の最適化にて使用される各種パラメータと範囲制約は表2の通りである。

3.3 結果についての考察

ここでは、実施した数値実験の結果をもとに、本手法の有用性について考察を行う。3.3.1節では、設計変数の独立・依存関係を認識・反映する手法を導入する

意義について考察し、3.3.2節では、本手法が設計変数の独立もしくは依存関係を認識し、利用することができるかどうか検証を行う。3.3.3節では、従来手法であるGA、SAと比較検討を行うことによりMSA、LSA、NOSAを導入する意義について評価を行う。最後に3.3.4節では、本手法の最適解への収束性について考察を行う。

3.3.1 独立・依存関係を認識・反映する意義

設計変数の独立・依存関係を認識・反映する意義について考察を行う。この検討のために、RA関数とGR関数を使用する。数値実験の試行回数は10回とする。また、試行ごとに一様乱数の乱数シードを変える。最終世代数は本試行についてのみ60,000世代とする。数値実験を試行した結果を表3に示す。表中の独立は $SD_i = 1$ の場合を示し、依存は $SD_i = 2$ の場合を示す。

表3の結果からRA関数では、 $SD_i = 1$ の場合は、全ての試行にて最適解が得られている(成功率は100%)。一方、 $SD_i = 2$ の場合については成功率が0%と最適解が全く得られていない。これは、表1と式(14)のRA関数の第2項を見れば明らかのように、設計変数の依存関係は独立、すなわち $SD_i = 1$ である。ゆえに、 $SD_i = 2$ の場合については最適解が得られない。GR関数では、 $SD_i = 1$ の場合の成功率は70%となり、 $SD_i = 2$ の場合の成功率は90%となった。よって、 $SD_i = 2$ 、すなわち設計変数の依存関係を考慮して最適化を行う必要がある。一方、最適解を求解するのに要した世代数は、GR関数の両ケースの場合について比較すると $SD_i = 1$ として試行した6回目の世代数が1,209世代と最も少ない。 $SD_i = 2$ の場合には、成功率は $SD_i = 1$ よりも高いが最適解を求解するのに要した世代数は5桁台の場合も多く計算コストを要する。また、本試行以外の数値実験の最終世代数である30,000世代を上限とすると $SD_i = 1$ と $SD_i = 2$ の成功率は同じ70%となる。よって、GR関数の場合には、計算コストを考慮すると $SD_i = 1$

表3 設計変数の独立・依存関係の検討のための数値実験結果

	RA		GR	
	Independence	Dependence	Independence	Dependence
1	5,212	60,000	1,803	60,000
2	5,345	60,000	60,000	33,823
3	6,652	60,000	60,000	19,911
4	4,274	60,000	2,674	2,170
5	4,266	60,000	60,000	3,748
6	4,033	60,000	1,209	26,787
7	3,888	60,000	3,443	26,874
8	3,945	60,000	6,897	2,988
9	3,625	60,000	7,607	49,783
10	3,258	60,000	2,059	2,695
Success rate (%)	100	0	70	90

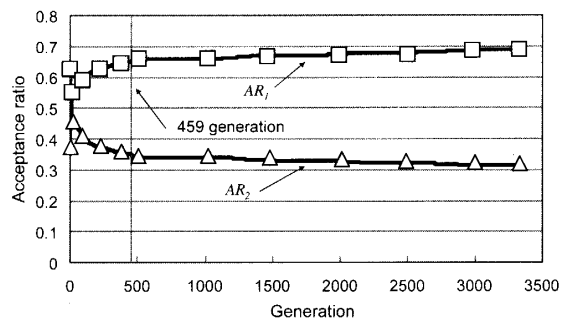
と $SD_i = 2$ の組み合わせによる最適化が必要である。これは、表1および式(16)の第2項が設計変数の和の項であり、第3項が設計変数の積の項であることから明らかであり、 $SD_i = 1$ と $SD_i = 2$ を組み合わせることが重要である。

以上のRA関数とGR関数の試行から、設計変数の自己更新プロセスにて最適化問題ごとに的確に設計変数の独立・依存関係を認識・反映する必要があるといえる。

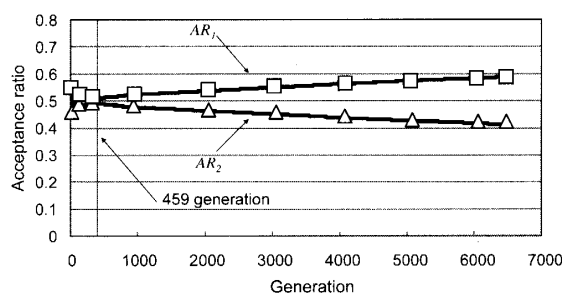
3.3.2 独立・依存関係についての考察

3.3.1節の結果を踏まえて、提案するハイブリッドメタ戦略が設計変数の独立・依存関係を的確に設計変数の自己更新プロセスに認識・反映できるか考察を行う。数値実験は、RA関数とGR関数を使用する。最終世代数は30,000世代とする。図10は受理率の結果を示す。図10(a)はRA関数の受理率、図10(b)はGR関数の受理率である。 $SD_i = 1$ の受理率 AR_1 は $AR_1 = AD_1 / (AD_1 + AD_2)$ 、 $SD_i = 2$ の受理率 AR_2 は $AR_2 = 1 - AR_1$ で計算される。図11はエリート個体の適応度の履歴である。この数値実験では、459世代目でLSAからNOSAに切り替わる。

RA関数の場合には、図10(a)の結果からLSAのフェーズにて受理率は指数関数的に0.7へ近づく。NOSAのフェーズでは、 SD_i の選択確率は AR_1 が約0.7

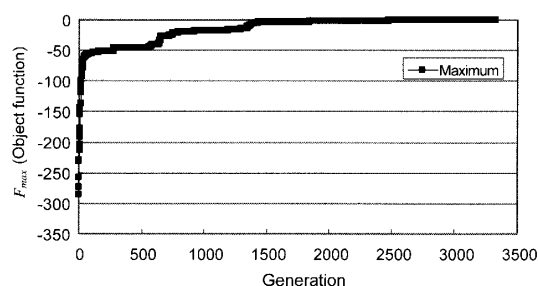


(a) Rastrigin

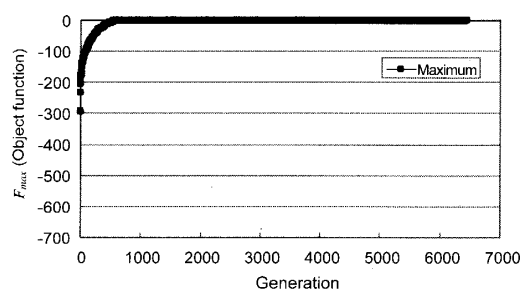


(b) Griewank

図10 受理率と世代数



(a) Rastrigin



(b) Griewank

図 11 エリート個体の適応度の履歴

であることから式(13)により一様分布とはならない。それゆえに、 $SD_i = 1$ は高い確率にて選択される。RA関数のエリート個体の適応度(図11(a)参照)は、初期世代の-286.197から459世代(LSAの最終世代)の-45.797に増加している。エリート個体のすべての設計変数の値は18番目の設計変数を除いて2,629世代で最適値に達している。その時のエリート個体の適応度は-0.0792となる。以降は、18番目の設計変数を変えるための最適化プロセスであり、最適解は3,328世代で得られる。RA関数の AR_1 は約0.7に収束し、NOSAフェーズの後半は1変数の最適化が実施されている。3.3.1節で述べたように、RA関数の設計変数は独立関係であるから、独立・依存関係が的確に設計変数の自己更新プロセスに認識・反映されているといえる。

GR関数の場合には、図10(b)の結果からLSAのフェーズで受理率 AR_1 , AR_2 は共に約0.5であることがわかる。さらに、NOSAのフェーズで、受理率 AR_1 は長い世代をかけて0.6に収束している。GR関数のエリート個体の適応度(図11(b)参照)は、初期世代の-296.118からLSAの最終世代の-9.5に増加している。エリート個体のすべての設計変数の値は1番目と4番目の設計変数(2つの設計変数)を除いて4,111世代で最適値に達し、その時のエリート個体の適応度は-0.0246となる。それ以降の最適化プロセスでは、2つの設計変数を対象とし、最適解は6,452世代で求まる。この

試行の結果、 AR_1 が0.6となり、 AR_2 が0.4となる。3.3.1節にて示したように、GR関数の場合には $SD_i = 1$ と $SD_i = 2$ の組み合わせによる最適化が必要であることから、これらの受理率の結果から本試行においても設計変数の依存関係が自己更新プロセスに認識・反映されていることがわかる。

RA関数とGR関数の試行結果から以下のようにまとめられる。

- 1) 提案したメタ戦略は、式(9)と式(13)によって、設計変数間の独立もしくは依存関係を認識することができる。
- 2) NOSAフェーズのDSPは、 SD_i の記録を用いて、設計変数の自己更新プロセスに設計変数の独立・依存関係を反映できる。

3.3.3 MSA, LSA, NOSAを導入する意義

ここでは、MSA, LSA, NOSAを導入する意義についてGAおよびSAとの比較を通して有効性を評価する。評価に利用するベンチマーク関数は、RA関数とGR関数である。本試行回数は10試行とし、試行ごとに一様乱数の乱数シードを変更する。最終世代数は3.3.2節と同じである。本手法と他の手法の評価単位を統一するためにSAについても世代を用いる。なお、世代内の繰り返し評価回数を一致させるために、GAを40個体、SAのメトロポリスループ数を40ループとし、40ループで1世代とする。GAの遺伝的操作は、ランキング戦略($P_s = 1.0$)、2点交差(グレイコーディング, $P_c = 1.0$)、エリート戦略、および突然変異(突然変異率 $P_m = 0.05$)を用いる。SAの各種設定は、初期温度 $T_0 = 10.0$ 、冷却率 $\gamma = 0.99$ 、終点温度 $T_{end} = 0.001$ 、近傍率を初期近傍率 NR_0 とする。

表4.5には、RA関数およびGR関数に対する試行結果を示す。表中の世代数は、最適解に収束した場合にはその到達世代数、収束しなかった場合には最終世代数とした。また、目的関数値はその世代の最大値である。なお、表中のPM(The proposed method)は本手法となる。表6には、各試行において最適な設計変数値が得られた設計変数の数を示す。各手法にて最適解が探索できた場合には、設計変数数が20変数であることから20となる。図12には、10試行目の2,500世代までのGA, SA, 本手法の最大目的関数値の履歴を示す。

はじめに、表4.5の結果からGAとSAの試行結果を考察する。GAの場合には、RA, GR関数の全ての試行にて最適解が得られていない。しかしながら、図12のように、SAと比較して急速に最適解近傍へ探索点を近づける能力は計算コスト低減の観点から魅力的であ

表4 GAおよびSAとの比較(RA関数)

RA	GA		SA		PM	
	generation	objective	generation	objective	generation	objective
1	30,000	-28.831	6,685	0.000	3,439	0.000
2	30,000	-27.891	12,658	0.000	3,525	0.000
3	30,000	-29.993	8,052	0.000	4,084	0.000
4	30,000	-27.821	6,940	0.000	3,914	0.000
5	30,000	-27.014	5,676	0.000	3,404	0.000
6	30,000	-32.964	7,281	0.000	3,298	0.000
7	30,000	-28.122	9,292	0.000	3,857	0.000
8	30,000	-30.947	5,430	0.000	2,723	0.000
9	30,000	-22.521	14,937	0.000	3,835	0.000
10	30,000	-29.588	10,423	0.000	4,217	0.000
Average	30,000	-28.569	8,737.4	0.000	3,629.6	0.000

PM: The proposed method

表5 GAおよびSAとの比較(GR関数)

GR	GA		SA		PM	
	generation	objective	generation	objective	generation	objective
1	30,000	-12.000	30,000	-0.113	2,805	0.000
2	30,000	-14.169	30,000	-0.128	6,875	0.000
3	30,000	-13.162	30,000	-0.033	5,357	0.000
4	30,000	-12.896	30,000	-0.067	13,373	0.000
5	30,000	-10.387	30,000	-0.121	6,775	0.000
6	30,000	-11.958	30,000	-0.054	14,594	0.000
7	30,000	-11.091	30,000	-0.031	22,681	0.000
8	30,000	-11.325	30,000	-0.044	6,704	0.000
9	30,000	-11.774	29,681	0.000	8,475	0.000
10	30,000	-14.463	30,000	-0.065	2,394	0.000
Average	30,000	-12.323	29,968.1	-0.066	9,003.3	0.000

PM: The proposed method

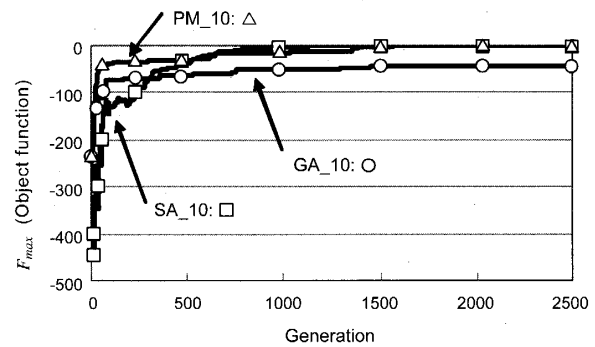
表6 最適な設計変数値が得られた設計変数数

	RA			GR		
	GA	SA	PM	GA	SA	PM
1	0	20	20	0	16	20
2	0	20	20	1	14	20
3	0	20	20	0	18	20
4	0	20	20	0	16	20
5	0	20	20	0	16	20
6	0	20	20	0	18	20
7	1	20	20	0	18	20
8	2	20	20	0	18	20
9	0	20	20	0	20	20
10	1	20	20	0	16	20
Average	0.4	20	20	0.1	17	20

PM: The proposed method

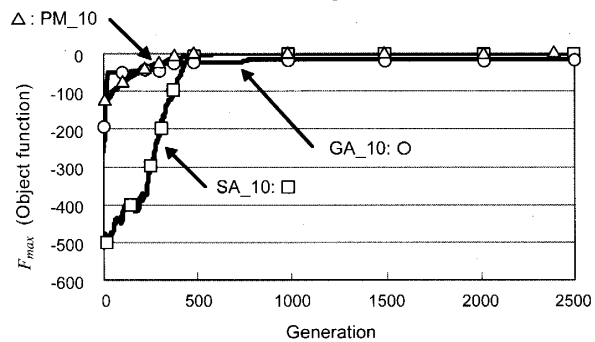
る。SAの場合には、RA関数については、本手法の2.4倍の平均世代数にて全ての試行で最適解が得られている。GR関数については、得られた最適解が10試行中1試行のみである。さらに、表6の結果をみるとSAの場合には、20変数中14～18変数で設計変数の最適値であるゼロが求められている。このことから、得られた最終解は、最適解近傍の満足解であると推定できる。また、GR関数は、設計変数間に依存関係のある問題であることから、得られた最終解が2の倍数の組み合わせになっていると推測できる。以上のことから、SAは計算コストを要するが確実に最適解近傍まで局所的探索が可能であるといえる。しかし、設計変数間に依存関係のある問題に対しては、最適解近傍領域での最適化が困難であることも確認できる。

つぎに、最適解への収束性と計算コストの観点からMSA, LSA, NOSAの導入する意義についてGAとSAの考察結果を通して述べる。



PM: The proposed method

(a) Rastrigin



PM: The proposed method

(b) Griewank

図12 10試行目のGA, SA, 提案手法の F_{max} の履歴(2,500世代まで)

本手法の最適解への収束性は、表4, 5に示すようにRA, GR関数の全ての試行にて最適解が得られている。これは、LSAのフェーズにて、設計変数の自己更新プロセスにSAを導入した結果、SAの考察結果と同様に、本手法でも確実に最適解近傍まで探索を行なっているためである。また、図12からLSAの最終世代である459世代目の目的関数の最大値をそれぞれ列挙すると、RA関数では、本手法が-32.877, SAが-41.311, GR関数では、本手法が-6.006, SAが-17.587となる。さらに、探索点の設計変数に最適値が含まれる数は、459(LSAの最終世代), 1,000, 2,000, 2,500世代目までそれぞれ列挙すると、RA関数では、本手法が7, 7, 15, 16変数, SAが1, 7, 12, 12変数, GR関数では、本手法が0, 18, 18, 20変数(2,394世代), SAが0, 12, 18, 18変数となる。RA, GR関数ともに、459世代目の目的関数の最大値がSAの結果よりも大きいことや、探索点の設計変数に最適値が含まれる数がGR関数の場合には両手法ともにゼロであるが、RA関数では、SAよりも多数含まれていることがわかる。これらのことから、LSAにより、SAよりも確実に最適解近傍まで探索できていることがわかる。また、NOSAのフェーズとなる1,000, 2,000, 2,500世代目の両関数の設計変数

に最適値が含まれる数を比較すると、SA よりも本手法のほうが早期の世代にて設計変数に多数の最適値を含むことがわかる。これは、NOSAのフェーズにて、設計変数の自己更新プロセスに設計変数間の依存関係を認識・反映するDSPを導入することで、RA関数のみならず、SAでは解決困難な設計変数間に依存関係のあるGR関数についても確実に探索点を更新できている結果であるといえる。よって、SAよりも最適解への収束性が優れていることがわかる。以上のことから、本手法におけるLSAおよびNOSAの有効性は明らかである。

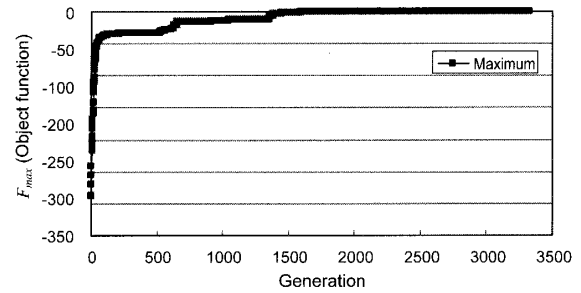
本手法の計算コストは、表4,5に示すように、GA、SAの平均世代数よりも大幅に少ない世代数にて最適解が得られている。これは、MSAにて、最適化プロセス全体をGAによりコントロールし、LSAとNOSAをハイブリッドする戦略であることから、最適解への収束性を維持したうえで、計算コストの低減ができたものと考えられる。また、図12に示すように、明らかにSAよりも早期の世代にて、最適解近傍へ探索点が急速に近づいていることが確認できる。このことは、GAの考察結果にて述べた「急速に最適解近傍に探索点を近づける能力」をMSAにて、本手法に反映できた結果といえる。以上のことから、MSA、LSA、NOSAが本手法にて有効に機能しているといえる。

3.3.4 最適解への収束性の考察

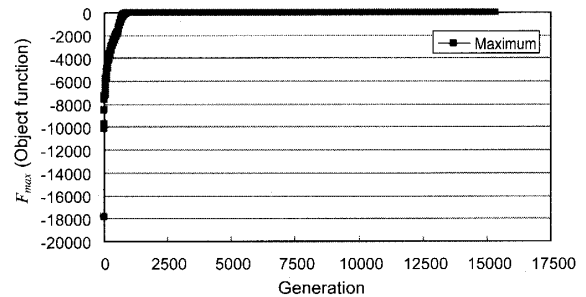
本節では、4つのベンチマーク関数を用いて、10回の数値実験を試行する。乱数シードや最終世代数は3.3.3節と同じである。本手法の総評価数は、(総評価数)=(個体数)×(メトロポリスループ、もしくはループ数)×(世代数)で計算される。図13と表7は、4つのベンチマーク関数を最適化した結果である。図14は、4つのベンチマーク関数に対する10試行目のエリート個体の適応度履歴である。本試行結果から、すべてのベンチマーク関数に関して最適解が得られてい

表7 最適解への到達世代と試行数

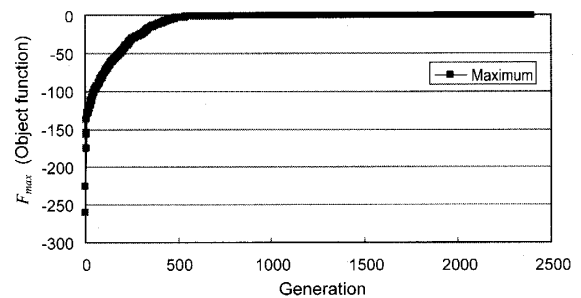
	RA	RI	GR	AC
1	3,439	21,246	2,805	3,864
2	3,525	15,333	6,875	3,944
3	4,084	15,996	5,357	3,181
4	3,914	23,486	13,373	2,557
5	3,404	11,739	6,775	5,035
6	3,298	23,698	14,594	1,755
7	3,857	16,335	22,681	2,409
8	2,723	16,657	6,704	3,782
9	3,835	17,063	8,475	3,380
10	4,217	15,306	2,394	3,179
Average	3,629.6	17,685.9	9,003.3	3,308.6



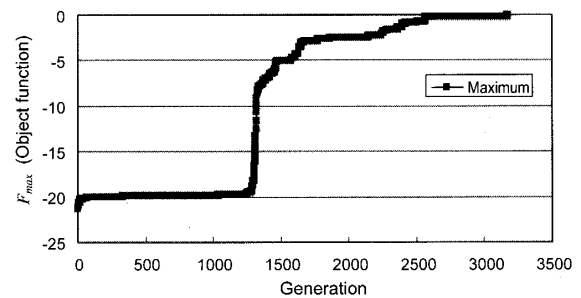
(a) Rastrigin



(b) Ridge



(c) Griewank



(d) Ackley

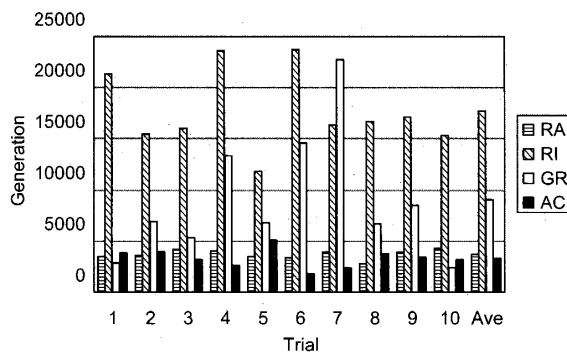


図13 最適解への到達世代と試行数

図14 10試行目のエリート個体の履歴

ることがわかる(成功率は100%)。よって、設計変数に依存関係のある多峰性関数に対しても、本手法にて最適化が可能である。また、文献3)のGAの場合には、20変数のRA関数の成功率が50%~70%、10変数のGR関数の成功率が75%~90%であることから、本手法が文献3)のGAよりも最適解への収束性に優れた手法であることがわかる。さらに、文献4)のreal-coded GAによるRA関数の結果と比較するために、次元を30変数に増やし最適化した結果、総評価数は $20 \times 2 \times 6,626.2 = 265,048$ 回(試行回数は10試行とし、成功率は100%)となる。この結果と文献4)の総評価数4,010,000回(成功率100%)を比較すると、本手法の計算コストが文献4)の計算コストよりも大幅に少ないことがわかる。

つぎに、ハイブリッドメタ戦略である文献14)のPSA/GAcと文献16)のPSASの2つの手法を用いて、RA関数とGR関数の試行結果から計算コストと最適解への収束性について比較を行う。なお、初期温度と試行回数は、本手法、PSASおよびPSA/GAcで同じ条件である。総評価数は10回の試行の平均値を用いる。設計変数の数は、PSASが20変数、PSA/GAcが10変数である。RA関数の総評価数は、 $20 \times 2 \times 3,629.6 = 145,184$ 回、GR関数の総評価数は、 $20 \times 2 \times 9,003.3 = 360,132$ 回である。文献16)より、PSASのRA関数の総評価数は、約150,000回、GR関数の総評価数は、約410,000回である。文献14)より、PSA/GAcのRA関数の総評価数は、3,034,881回、GR関数の総評価数は、3,008,201回である。本手法の総評価数はPSASの値とほぼ同じであるが、PSASをGR関数に用いた場合には、10回中5回の試行のみ最適解に到達している(成功率は50%)。本手法の最適解への収束性は、PSASと同レベルの総評価数でありながら、全ての試行で最適解に収束している。よって、PSASより大幅に収束性が改善していることがわかる。また、本手法は、20変数にもかかわらず、10変数のPSA/GAcの総評価数よりも1桁小さい総評価数となる。

さらに、文献17)のMAと比較を行なう。この比較のために、GR関数のみ10変数とする。本手法にて、10変数のGR関数を最適化すると、総評価数が $20 \times 2 \times 2,150.6 = 86,024$ 回(試行回数は10試行)にて、すべての試行にて最適解が得られている(成功率100%)。しかしながら、MAの終了評価回数は、40,000回であることから、本結果を用いて比較検討することができない。このため、同じ終了評価回数となるように、本手法の最終世代数を1,000世代とする。本試行は、最終

世代数が成功率100%の場合の到達世代数の0.46倍と極端に少ないことから、最適解への収束性が期待できない。そこで、最適解への収束性を向上させるために、LSAのパラメータを数値実験し、その結果から冷却率と終点温度を $\gamma=0.95$ 、 $T_{end}=0.01$ とする。この条件にて、本手法によりGR関数の最適化を行なう。その結果、総評価数が $20 \times 2 \times 879.9 = 35,196$ 回となり、10試行中6試行にて最適解が得られる(成功率60%)。なお、この試行の目的関数の平均最大値は-0.011となる。RA関数による比較のために、図14(a)のRA関数の結果から、1,000世代目の目的関数の最大値を求めると-15.017となる。

文献17)のGR関数の最適化は、設計者の判断を含むMA-S2Bと設計者の判断を含まないMA-S2の2通りの数値実験を行なっている。MA-S2Bは、総評価回数が21,250回にて、最適解が得られている。しかし、設計者の判断能力に大きく依存するため、安定的に最適解が得られるとは考えづらい。さらに、RA関数には、MA-S2Bを適用していない。そこで、本論文では、MA-S2の結果と比較する。GR関数のMA-S2の結果は、総評価数が40,000回にて、目的関数の平均最大値が-2.80(平均最大値とするために、文献17)の平均最小値にマイナスを付加)となり、最適解が得られていない。これに対し、本手法の場合には、最適解が得られ、さらに総評価数が35,196回にて、目的関数の平均最大値が-0.011となる。よって、MA-S2よりも優れた結果といえる。また、MA-S2のRA関数の場合も、総評価数が40,000回にて、目的関数の平均最大値が-1.55となり、最適解が得られていない。しかし、MA-S2の40,000回以降の最適解への収束性については判断できないが、1,000世代目の本手法の目的関数の最大値-15.017よりも大きいことがわかる。

以上、既存手法の最適化結果^{3,4,14,16,17)}と比較検討した結果、本論文にて提案したハイブリッドメタ戦略は、計算コストの総数、および最適解への収束性を大幅に改善することが可能であることがわかった。

4. おわりに

本論文では、工学分野を支援するために、新たに多峰性の整数最適化問題に対するハイブリッドメタ戦略の手法を提案した。具体的には、最適化プロセスにて必要となる探索能力を従来のような大域的探索能力と局所的探索能力の組み合わせとするのではなく、複合探索能力、局所的探索能力、最適解近傍の探索能力の3つの組み合わせであると定義する。この最適解近傍

の探索能力として、設計変数間の依存関係を自動的に認識・反映したうえで、設計変数の自己更新プロセスを行う直接探索プロセスを導入した。この提案手法を4つのベンチマーク関数に適用し、最適解への収束性や計算コストについて評価を行った。その結果、最適解への収束性の大幅な改善と計算コストの低減を実現した。また、設計変数が依存関係にある多峰性の最適問題についても最適化ができることを確認した。以上のことから、本提案手法は、工学分野を支援するのに有効な多峰性の整数最適化手法であるといえる。今後は、実用化のために、さらなる性能改善とともに工学分野の実問題に対して適用を試み、その有用性を検討する予定である。

参 考 文 献

- 1) Holland, J.: Adaptation in Natural and Artificial Systems, The University of Michigan (1975), MIT Press (1992)
- 2) Goldberg, D. E.: Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley (1989)
- 3) Digalakis, J. D. and Margaritis, K. G.: An Experimental Study of Benchmarking Functions for Genetic Algorithms, Proc. of IEEE Conference on Transactions, **V**, 3810/3815 (2000)
- 4) Sakuma, J and Kobayashi, S.: Extrapolation-Directed Crossover for Real-coded GA: Overcoming Deceptive Phenomena by Extrapolative Search, Proc. of Congress on Evolutionary Computation (CEC 2001), 685/662 (2001)
- 5) Li, X. and Kirley, M.: The Effects of Varying Population Density in a Fine-grained Parallel Genetic Algorithm, Proc. of Congress on Evolutionary Computation (CEC 2002), **2**, 1709/1714 (2002)
- 6) 長谷川浩志, 川面恵司: GA利用による機械構造物の位相最適化の一方法(有限要素の除去および付加パラメータを染色体とする位相最適化法), 日本機械学会論文集, **61**-581 A 編 (1995)
- 7) 鶴田靖, 長谷川浩志, 川面恵司: GA利用による機械構造物の位相最適化の一方法(第2報, 有限要素の除去・付加パラメータを染色体とする方法の収束性の検討), 日本機械学会論文集, **63**-605 A 編 (1997)
- 8) Tsuruta, Y., Suzuki, Y., Hasegawa, H. and Kawamo, K.: A Method for the Shape and Topology Optimization of Mechanical Structures Using Genetic Algorithm, JSME Centennial Grand Congress by the JSME International Symposium on OPID97, Paper #137 (1997)
- 9) Nishi, H., Okamoto, T. and Kawamo, K.: Study on the Development of General-purpose Genetic Algorithm Engine, Extend Abstracts of WCSMO-4, 428/429 (2001)
- 10) 河田晴信, 上原広靖, 川面恵司: 汎用最適化エンジンの開発とそのテストラン, シミュレーション, **21**-4, 280/288 (2002)
- 11) Rosen, B. E., 中野良平: シミュレーテッドアニーリングー基礎と最新技術一, 人工知能学会誌, **9**-3 (1994)
- 12) Mahfoud, S. W., Goldberg, D. E.: A genetic algorithm for parallel simulated annealing, Parallel Problem Solving from Nature 2, 301/310 (1992)
- 13) 上原広靖, 河田晴信, 川面恵司: 遺伝的アルゴリズムとシミュレーテッド・アニーリングとの混合法による最適値探索実験, 第52回理論応用力学講演会講演論文集, 117/118 (2003)
- 14) Hiroyasu, T., Miki, M. and Ogura, M.: Parallel Simulated Annealing using Genetic Crossover, Proceedings of the ISCA 13th International Conference on PDCS-2000 (2000)
- 15) Miki, M., Hiroyasu, T. and Fushimi, T.: Parallel Simulated Annealing with Adaptive Neighborhood Determined by GA, IEEE International Conference on System, Man and Cybernetics (2003)
- 16) 上原広靖: 汎用最適化エンジンの開発および性能評価に関する研究 一淘汰・繁殖の概念を利用した並列シミュレーテッドアニーリング法の提案一, 芝浦工業大学大学院機械工学専攻修士論文 (2004)
- 17) Ong, Y. S. and Keane, A. J.: Meta-Lamarckian Learning in Memetic Algorithms, IEEE transactions on evolutionary computation, **8**-2, 99/110 (2004)