分散型多目的 GA による自動車エンジン排気系形状の最適設計

Design Optimization of Exhaust Manifold for a Car Using Parallel MOGA

森川雅司，東北大学大学院工学研究科，仙台市青葉区荒巻字青葉 01, morikawa@ad.mech.tohoku.ac.jp
金崎雅博，東北大学大学院情報科学研究科，仙台市青葉区片平 2-1-1, kanazaki@mail.cc.tohoku.ac.jp
大林茂，東北大学流体科学研究所，仙台市青葉区片平 2-1-1, obayashi@ieee.org
中橋和博，東北大学大学院工学研究科，仙台市青葉区荒巻字青葉 01, naka@ad.mech.tohoku.ac.jp
Masashi Morikawa, Dept. of Aeronautics and Space Engineering, Tohoku University, Sendai 980-8579, Japan
Masahiro Kanazaki, Institute of Fluid Dynamics, Tohoku University, Sendai 980-8577, Japan
Shigeru Obayashi, Institute of Fluid Dynamics, Tohoku University, Sendai 980-8577, Japan
Kazuhiro Nakahashi, Dept. of Aeronautics and Space Engineering, Tohoku University, Sendai 980-8579, Japan

Abstract: Multiobjective design optimization of exhaust manifold for a car engine has performed using DRMOGA (Divided Range Multi-Objective Genetic Algorithm). Two design objectives are considered: engine power and catalyst performance. Tradeoffs between these objectives are revealed. Performance of DRMOGA is examined and the result shows that DRMOGA is suited to complex problems.

1. 緒言

近年のユーザーニーズの多様化に伴い、様々な設計要求を考慮した自動車エンジンの開発が必要になってきた。

環境問題が広く認知されている近年では、環境への適合性も重要視されてきている。有害物質を取り除くために排気マニホールド出口には触媒が搭載されている。自動車エンジンの有害物質除去能力は排気マニホールド出口での排気温度により評価できる。また、燃焼室出口での負圧を大きくすることが出来れば燃焼室での新気充填効率を上げることが出来る。この充填効率を改善出来れば、自動車エンジンの出力増大、燃費改善につながる。

これらの設計要求は経験上両立させることが難しいことが知られており、設計の効率化と新たな設計指針の把握を目的として GA を用いた最適設計システムが開発された[1]。このシステムを用いて、Fig.1(a)に示す形状を初期形状とした形状最適化が行われ、有害物質除去能力の高い形状を得ることができた。しかしながら、充填効率については顕著な上昇を見ることができなかった。この原因として、初期形状が一般車への搭載を前提とされたもので、高出力を要求するものでは無かったことが挙げられる。

そこで本研究では、高出力が期待できる形状(Fig.1(b))を初期形状とした多目的最適化を行う。この形状は分岐形態を変更しやすく、排気干渉の最適化による高出力化が期待できる。また、多目的 GA には、探査性能の優れた領域分割型多目的 GA(DRMOGA)[2]を適用し、更なる性能の向上を目指す。DRMOGA は多様性を維持し優れたパレート解を得ることが出来ると知られている。

a) b)

Fig.1 Geometries of exhaust manifolds

2. 最適化手法

最適化手法には数ある手法の中で遺伝的アルゴリズム（GA）を使用する。GA はダーウィンの進化論に基づく最適化手法であり、適用分野における数学的背景や理論に対する深い知識や経験を必要としないため、実用問題に適用されている。勾配法のように目的関数の勾配を求める必要がないため、目的関数が微分可能、凸関数でなければならないといったような制限はない。さらに、決定論的な解の探索ではなく、確率論的に解を探索する。これらの特徴から、GA を用いた最適化により、従来の最適化法で問題となっていた局所解への収束を防ぐことができ、適用問題がたとえ非線形性や多峰性の設計空間を持つような問題であっても、大域的最適解を探索することが可能となる。

2.1 遺伝的アルゴリズム (Genetic Algorithm, GA)

遺伝的アルゴリズム（GA）とは 1975 年に J.H.Holland によって提案された、生物の進化を模倣した学習的アルゴリズムである。すなわち、何万年もかけて進化してきた生物の遺伝法則をモデル化し、工学的な実用問題に応用したアルゴリズムである[3]。

自然界における生物の進化過程では、ある世代の個体群で、生活環境に対する適応度が高い個体が高い確率で生き残り、それらの個体による交叉や突然変異によって次の世代が形成される。その繰り返しにより、環境により適合した種が現存している。この世代間のつながりを介しているのが遺伝子であり、遺伝子の集合である染色体によって各個体のマクロな特徴が決定される。

この進化の流れをアルゴリズムに取り入れた GA の流れを Fig. 2に示す。まず、ランダムに設計変数を発生させ、「初期集団」を形成する。個体集団の数は多いほど設計空間を広く覆うことができるが、その分個体の評価に必要な計算時間も増大するため、適当な数に抑える必要がある。次に、遺伝子として設計変数を与えられた各個体に対し、目的関数を計算することにより適応度を「評価」する。全個体の評価値が求まった後、評価に従って親となる２個体を「選択」

する。そして「交叉」、つまり親個体の遺伝子の一部交換を行うことにより子を作る。また、新しくできた個体に対しある確率で設計変数に変化を与え「突然変異」を行う。この「選択」、「交叉」、「突然変異」を子の個体数が指定個体数になるまで繰り返し、次世代の個体群を形成する。最後に「世代交代」を行い、次世代の集団が構成される。これらの動作を指定された世代数、もしくは収束するまで繰り返す。

GA の一連の操作からわかるように、GA では 1 世代中の個体数と世代数の積の数だけ目的関数を評価するため、勾配法と比べ評価回数が多くなり、計算負荷が大きいという欠点がある。特に空力最適化問題においては個体の空力評価に CFD コードを用いるため、計算負荷が大きい。従って、全体の計算時間のうちほとんどが個体の「評価」の部分に費やされてしまう。そこで、並列計算機を利用して各個体評価の計算を並列計算させて、同時にすべての個体を評価することが考えられる。これによって計算時間の大幅な短縮になる。

以下に本研究で用いた各遺伝的操作について簡単に説明する。

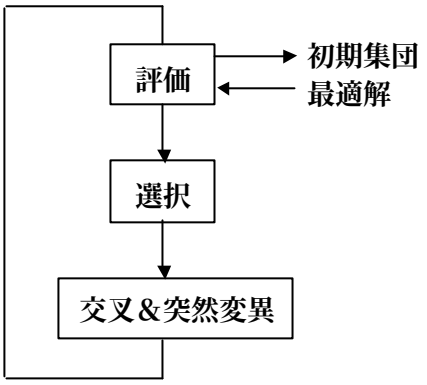

Fig. 2 Flowchart of GA

初期集団の生成（Initialization）

設計変数の定義域に一様乱数を発生させ、個体を分布させた。また、制約条件を設けたため、これを満たさない個体は切り捨て、再びその分個体を発生させて、指定個体数になるまで、一様乱数による発生を繰り返した。

コード化・デコード化（Encoding・Decoding）

一般に GA では、実数の設計変数を遺伝子として扱うために 2 進数の文字列にそれらの値をコード化するが、本研究では Michalewicz による方法[4]に従い、実数の設計変数をそのまま遺伝子として扱うこととした。実数値を用いると 2 進数に変換する際の精度落ちを防止することができる。また、PJ. Fleming により実数コーディングによる MOGA の探索性能の向上が証明されている[5]。

評価（Evaluation）

各個体について目的関数の計算を行い、その優劣を決定する適応度(Fitness)を決める。適応度の高い個体は次の世代に残る確率は高くなる。空力最適化問題の場合、計算時間の大部分はこの部分であるため、評価の仕方次第で最適化の効率が大きく左右される。本研究では、各個体ごとに並列計算することにより、各個体の評価を同時に行い、計算時間を短縮した。

選択（Selection）

子を生成するために親となる 2 個体を選択する。ここでは、適応度が高いほど選択されやくする。本研究では、一つの個体は適応度の高い個体を選択し、もう一つの個体はランダムに選択することにした。

交叉（Crossover）

選択された親 2 個体間で、遺伝子の入れ替えを行い、異なる 2 個体を形成する。最もよく用いられる方法として Eshelman と Schaffer により提案された Blend crossover(BLX-α)がある[6]。この方法では、親 2 個体により定義された区画内において子が作られるが、その区画は設計者がαの値を決定することにより拡張される。たとえば、子は以下のように作られる。

$$Child1 = g \times Parent1 + (1-g) \times Parent2$$
$$Child2 = (1-g) \times Parent1 + g \times Parent2$$
$$g = (1+2a)ran - a$$

$Child1, Child2$ は子を、$Parent1, Parent2$ は親を示す。ran は[0,1]の一様乱数を示し、設計変数ごとに発生させる。例えば、BLX-0.5 の場合、子が親の外にできる確率と間にできる確率は等しい。一方、BLX-0.0 の場合は、親の間に必ず子が作られる。αは大域的探索能力と最適解探索能力の重み付けの役割を果たす。αが 0.0 の場合、子が親同士の間にできるため、収束が速いが探索設計空間は徐々に狭くなっていく。一方、αを 0.5 にした場合、大域的探索能力と最適解探索能力の重みが等しく、バランスのとれた最適化を行うことができる。従って、本研究ではα=0.5 を採用する。

突然変異（Mutation）

突然変異は個体中の遺伝子に対してある確率で何らかの強制的な変化を施すことにより、解の多様性を維持し、交叉のみでは探索することのできない設計空間の探索を可能にすることができる。しかし、突然変異率を大きくとりすぎると、GA はランダムサーチに近づき、解の収束速度が落ちる。そのため、適切な確率で突然変異をすることが重要である。本研究では、テスト関数による検証により、10%と設定した。

一般に突然変異は、ビット列の中のある位置を反転させることにより行われるが、本研究では、実数値で表される遺伝子に、設計変数値の上限と下限の差の 10%までの値を擾乱として与える。

世代交代（Alternation of generations）

交叉、突然変異により子が生成されるが、次世代には限られた数の個体しか残せない。一方、生成された子が必ずしも親よりも優れているとは限らないため、親の集団からも次世代に個体を残す必要がある。そこで、本研究では、親と子、すべての適応度を比較し、適応度の高い順に上位半分を次世代に残す Best N 選択[7]と呼ばれる方法を使用する。

2.2 多目的遺伝的アルゴリズム

本研究におけるエンジン排気マニホールドの最適設計には後に詳しく述べるように、複数の目的関数が存在する多目的最適化問題となる。複数の目的関数間にはそれぞれトレードオフの関係になっている場合が多い。そのようなとき、最適解は一つではなく多数存在することになり、パレート最適解と呼ばれる。このパレート最適解を得ることが多目的最適化問題の目的となる。

パレート最適解とは、ある目的関数の値を改善するためには少なくとも 1 つの他の目的関数値を改悪せざるを得ない解のことであり、目的関数間のトレードオフに関して最適な解の集合を形成することになる。ここで、複数の個体の集団により多点同時探索を行うという GA の特徴を考慮すると、目的関数をパレート最適性で評価し、パレート最適集合の要素を同時に多数求めることが可能であることがわかる。

パレート最適解の定義

パレート最適解という概念は、経済学者パレートにより定義されたものである。経済学において、消費者行動の分析においては「消費者は自己の満足を最大にするように行動する」事が前提となっており、その消費者の満足を効用と表現する。そして、ある人が効用を増加させようとすると、他の人の効用を減少させなければならない状態を最も効率的な状態であるという。これは言い換えると、経済全体に存在する資源から効率的に財が生産され、それらの財が効率的に分配されている状態とも言える。このため、ある人への満足を大きくするために、資源や生産物の配分を変更すると、他の人への満足を下げてしまうのである。この状態がパレート最適な状態として定義されている[8]。

ここで、パレート最適性の概念をより厳密に定義する。k個の最適化目的を持つ最大化問題において、X を実行可能解の集合とした場合、$x^* \in X$ に対して、$f_i(x) \geq f_i(x^*), i = 1, \cdots, k$ で、しかも、あるjについて $f_j(x) > f_j(x^*)$ となるような $x \in X$ が存在しないとき、x^* をパレート最適解と呼ぶ。この定義によると、パレート最適解は、ほかよりも劣っていない解という意味で非劣解、もしくはほかのどの解にも支配されない解という意味で非支配解とも呼ぶことができる。

多目的最適化の最終的な目標は、以上のように定義されるパレート最適解の集合を探索し、そのパレート面から目的関数間のトレードオフに関する情報を得ることとなる。

多様性の維持

GA の大きな特徴として多点同時探索を行うことが上げられるが、その探索能力を十分に発揮するためには、常に解空間を広く覆っている必要がある。これは、パレート最適解という解の集合を偏りなく一様に得ることを目的とする多目的最適化を考えた場合には特に重要である。本研究においては、目的関数空間において、個体が密集している領域のランクを下げるフィットネスシェアリングを用い、多様性を持たせることとした。

2.3 分散型 MOGA

分散型 GA とは、GA における個体群（母集団と呼ぶ）を分割し、その分割された各子集団ごとに 2 章で述べた GA の操作を行う方法である。また、数世代ごとに各子集団間で個体の交換（移住）を行う。

各子集団に分けることにより個体数が減り、収束が早くなる。収束が早いということは局所解に陥りやすいという問題があるが、ある子集団で初期収束してしまっても他の子集団と個体を交換することにより多様性が維持されることにより大域的な最適解が探索できるというものである。この効果は単一目的問題の場合に関して証明されている。しかし、多目的問題の場合となると、最適解は一つではなく 2.2 章で述べたようなパレート解により表される。より良いパレート解というものはただ単に真のパレート解に近いというだけではなく、目的関数空間にいかに広く分布しているかということも重要となる。すなわち、多様性の維持が必要となる。母集団を分割することにより、一つの集団の個体数が減ることは多様性の維持とは相反するものとなりうるため、単一目的問題の場合と同様な分割方法や移住方法を用いると、あまり良い性能が出ない場合が出てくる。そこで、母集団の分割、移住の方法にある工夫を施す必要がある。

本研究では目的関数空間における領域を考慮し子集団に分割する領域分割型 MOGA を採用することとした。次にその手法について簡単に説明する。

2.3.1 領域分割型 MOGA (Divided Range Multiobjective GA, DRMOGA)

DRMOGA は目的関数空間を考慮し、母集団の分割や各子集団の移住を行うものである。DRMOGA の流れを以下に示す。

Step1　初期集団を設計変数領域内にランダムに N 個生成する
Step2　N 個の個体全体を一つの目的関数 F_i に関してソートを行う
Step3　F_i の値の最大値 $F_i(x)$ から目的関数値順に N/m 個の子集団に分割する
Step4　各子集団ごとに MOGA の操作を行う
Step5　ある決められた世代数 k ごとに Step2 に戻り別の目的関数 F_j により集団の再分割を行う。このとき自動的に移住が行われることになる。

ここで、N は全体の個体数、m は子集団の数、F_i は i 番目の目的関数値、k は子集団内で MOGA を行う世代数を示す。Step3 により Fig. 3 のように領域が分割され子集団に分配される。

この手法の優れている点は、Fig. 3 からわかるようにそれぞれの子集団に含まれる個体同士が目的関数空間において近傍にいることである。これにより、まったくかけ離れた個体同士の交叉が防がれ、近傍探索を行うことにより、効率的に解を探索できる。また、移住する個体数や、どの子集団と移住を行うのかといったパラメータが不要になることで、一定した性能を発揮できることが指摘されている。

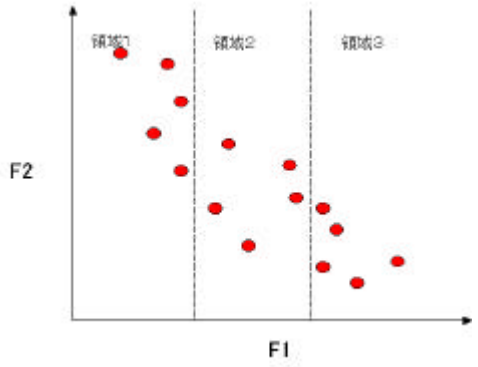

Fig. 3 Image of division by DRMOGA

2.3.2 性能比較

DRMOGA の性能を検証するために、テスト関数に適用し、通常の MOGA と比較した。以下にその関数を示す[4]。

$$Minimize \quad f_1(\vec{x}) = f_1(x_1, x_2, \ldots, x_m)$$

$$Minimize \quad f_2(\vec{x}) = g(x_{m+1}, \ldots, x_N)h(f_1, g)$$

$$f_1(x_1) = x_1$$

$$g(x_2, \ldots, x_N) = 1 + 10(N-1) + \sum_{i=2}^{N} \left(x_i^2 - 10\cos(2\pi x_i)\right)$$

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{0.5}$$

実問題における目的関数は複雑なものであり、設計変数空間は多峰性の強い場合が多い。そのため、今回使用するテスト関数は 2 目的最小化問題で、多峰性の強いものを選んだ。Table. 1に性能検証に用いた GA パラメータを示す。また、Fig.4 にその結果を示す。

結果より、通常の MOGA よりも分散型 MOGA の方がより良いパレート解を得ることができたことがわかる。

Table. 1 Parameter of GA

	MOGA	分散型 MOGA
突然変異率	0.1	0.1
子集団の数	—	2
移住間隔	—	10 世代
交叉方法	BLX0.5	BLX0.5
シェアリング	有	有
個体数	500	500
世代数	600	600

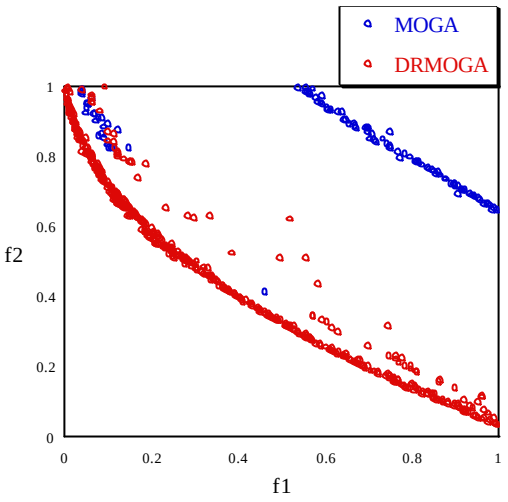

Fig. 4 Comparison of DRMOGA and MOGA

3. 自動車エンジンサイクルシミュレーション

3.1 一次元モデルを用いた吸排気性能予測

一般に自動車用エンジンの吸排気系は管と容器で構成されているが、系の中にはバルブや燃焼室などの機械部分も含まれ、流れは非定常となり、その形状も複雑である。そこで、本研究では空力評価を効率的に行うため、吸気から排気までの流れを、基本的に一次元モデルを用いた既存の非定常コード[5]を用いて計算し、三次元的な効果を評価したい部分(分岐部分など)について圧縮性三次元オイラーコード[6]をカップリングさせることによって空力評価を行った。

本研究で用いた一次元コードは、設計、開発現場ですでに活用されているもので、実用上十分な精度で性能予測ができる。一次元コードにおける管や容器部分モデル化には、一次元非定常圧縮性流体の基礎式が適用されており、壁面摩擦及び曲がり損失、壁面での熱交換を考慮した保存式になっている。摩擦損失係数や曲がり損失係数は定常流試験から得た値が用いられている。

燃焼室、エアクリーナなどは容器モデルとしてあらわされ、容器内の状態は一様で、流入したガスは瞬時に混合すると仮定されている. 燃焼による発熱量は実験により得た熱発生率が用いられている。

3.2 部分的な三次元計算コードの導入

本研究では、排気マニホールドの三次元形状についての最適化を行うために、サイクルシミュレーションを部分的に三次元非構造 CFD 計算に拡張する。この手法により、目的の部分に対して三次元形状を考慮した評価を行う。特に、一次元モデルでは管の長さしか考慮されないのに対して、三次元コードでは管の曲率や分岐部での交わり角度などが評価できる。

三次元部分は非構造格子法により、三次元圧縮性オイラー方程式を解く。支配方程式は積分表示により次式で表される。

$$\frac{\partial}{\partial t}\iiint \mathbf{Q}dV + \iint \mathbf{F}\cdot\mathbf{n}dS = 0$$

ここで、Q は変数ベクトル、F は非粘性数値流束ベクトルである。支配方程式はセル節点有限体積法によって離散化される。流束の計

算には流束差分離法を用いる。時間積分には非構造格子法のために拡張された LU-SGS 陰解法を適用する。

また、このカップリングを行う際の境界条件として、一次元計算で得たメッシュでの物理量を三次元計算での流入出境界条件として与え、三次元形状の境界の物理量を一次元計算で対応するメッシュに与える。

4. 排気マニホールドの多目的最適化

4.1 設計対象

本研究で最適設計を行う排気マニホールドの形状はFig. 6に示されるとおりである。また、モデルとしたエンジンの排気のタイミングは、排気干渉を小さくするために、近接したポートからの排気を避けて#1→#3→#4→#2 の順番としている。本研究で扱った形状は充填効率の向上が期待されるものとなっている。

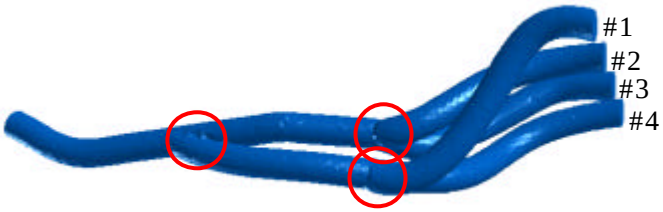

Fig. 5 Original shape

4.2 形状定義方法

本研究では排気マニホールドの形状最適化をMOGAにより行うが、その際に排気マニホールドの形状を自動的かつ計算に適合するように変形させなくてはならない。また、極力広い設計空間をカバーするためにはこの形状の大変形を行う必要がある。本研究では、オリジナル形状の分岐位置 (Fig. 6の円で囲われた部分) に擾乱を与えることにより様々な形状を作成することとした。

#3 の管は固定とし、#2 と#3、#3 と#1 の分岐位置に擾乱を与え、オリジナル形状の軸線から相似的に変形をおこなう。そして#1 と#4 の分岐位置にも同様に擾乱を与え分岐位置が変化した分、オリジナル形状との相似変形を行う。変形された形状はB-スプライン図で表され、それに基づいて非構造格子法により格子生成を行う。

また、排気マニホールドの管端はシリンダや触媒の位置関係から固定される

4.3 問題設定

本章の最適化では次のような多目的最適化問題を考える。

目的関数　：

低回転時(1500RPM)の排気マニホールド出口での排気温度の最大化

高回転時(6000RPM)のシリンダ充填効率の最大化

設計変数　：　オリジナル形状の 3 つの分岐位置

制約条件　：　各ポートの管端面の位置が一定

分散型MOGA を用いて最適化する。そのGA パラメータはTable. 1 と同様にし、空力評価計算は多くできないため、個体数は個体数は 32 とした。また、移住間隔は 8 世代ごととした。

評価は表 2 に示すエンジンを仮定し、サイクルシミュレーションを用いて行う。集団サイズは 32 としたが、低回転運転時と高回転運転時でそれぞれ条件を変えて行うために、実際の評価は 2×（個体数）行う必要がある、本研究での多目的最適化では実際の評価回数は 1 世代あたり 2×32 = 64 回となるので、東北大学流体科学研究所 ORIGIN2000 の 64PE を用いて並列に処理を行った。

Table.2　Engine specifications

ボア径	7.80 cm
ストローク長	7,84 cm
圧縮比	10.90

4.4 結果

分散型 MOGA による 17 世代後のパレート解を Fig. 7 に示す。また、パレート解の 4 点(A,B,C,D)の形状をそれぞれ Fig.8 に示す。

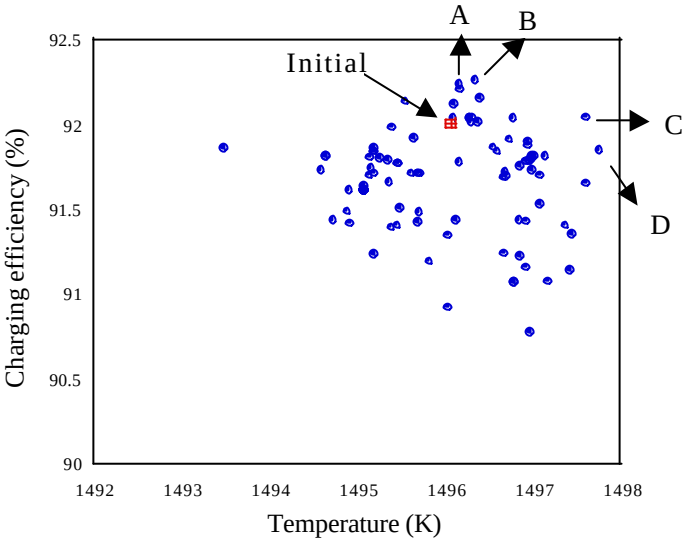

Fig. 6 Pareto solutions in the objective function space

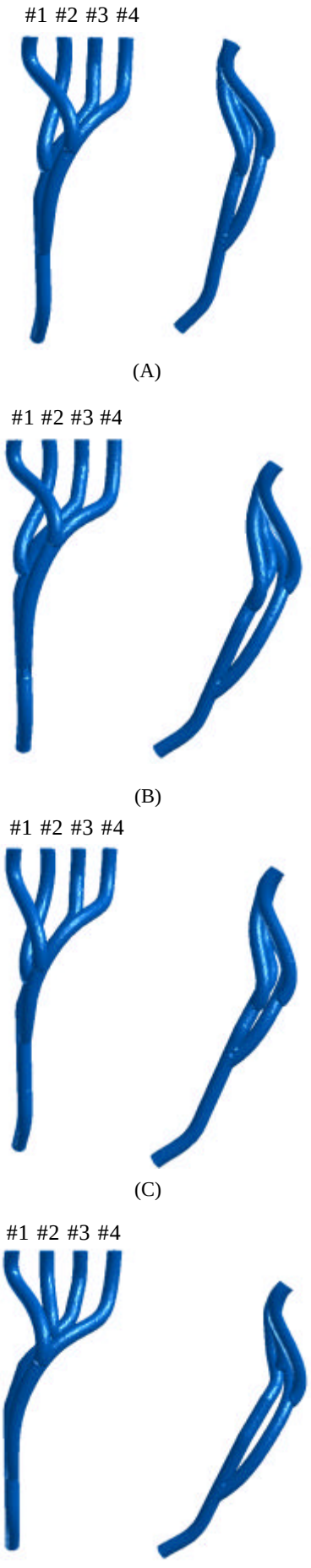

Fig. 7 Comparison of optimized shapes (A, B, C, D)

Fig. 6 より、オリジナル形状のものより性能の向上した形状が探索されたことがわかる。

形状 A、B は高い充填効率、形状 C、D は排気温度の維持において優れた性能を持つものである。形状 A、B を見ると、これらはほぼ同じような形状をしており、4 本の管すべてが十分な長さをとることによりシリンダポートとの距離も大きくなり、また分岐位置がそれぞれ離れた位置に置かれている。これらの要素により、他の管との干渉が減ると考えられる。次に、形状 C、D を見ると、これらも高い充填効率を得ているため、4 本の管のうち 3 本は形状 A、B と似た形状となっている。しかし、形状 C では#4 の管が、形状 D では#2 の管が短いものとなっている。これにより、分岐位置での干渉が起きたり、シリンダポートへの影響が大きくなるなどして充填効率は多少落ちるものの、排気温度が上昇する。

5. 結言

本研究では、領域分割型 MOGA の性能を検証し高い探査能力を持つことを確認した。そして領域分割型 MOGA を用いて自動車エンジン用排気マニホールドの最適設計を行い、パレート解を得た。

そのパレート解より排気マニホールドにおける設計のトレードオフが明らかとなった。充填効率上昇には管同士の干渉を最小にする設計が必要である。本研究で得られた形状ではシリンダポートから分岐までの距離が大きくなるものであった。これにより、同時に分岐形状が管同士鋭角に交わることにより、流れの干渉も低減することができていると考えられる。一方、排気温度最大形状は分岐による排気干渉を最小にすることよりも、なるべく早く排気を合流させるものとなった。

謝辞

エンジンの吸排気シミュレーションと排気マニホールドの設計に関する共同研究におきまして、データの提供や様々なご助言をいただきましたマツダ株式会社の技術研究所 パワートレイン研究グループ 矢野 康英氏、中川　正氏、本研究に必要な計算環境を提供していただいた東北大学流体科学研究所未来流体情報創造センター関係各位の皆様に心より感謝の意を表します。

参考文献

(1) Masahiro Kanazaki, Shigeru Obayashi, Kazuhiro Nakahashi, “ The Design Optimization of Intake/Exhaust Performance of a Car Engine Using MOGA,” EUROGEN 2001, Athens, Sep. 19-21, 2001, in print.

(2) Tomoyuki HIROYASU, Mitsunori MIKI and Sinya WATANABE. “The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems （Divided Range Multi-Objective Genetic Algorithm）,” IEEE Proceedings of the Congress on Evolutionary Computation 2000, Vol. 1, (2000) , pp.333-340

(3) Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, Inc., Reading, 1989.

(4) Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*, Second extended edition, Springer-Verlag, Berlin, 1994.

(5) Purshouse. R. C, Fleming. P. J , “ The Multiobjective Genetic Algorithm Applied to Benchmark Problems – An Analysis,” Research Report No. 796, Department of Automatic Control and System Engineering, University of Sheffield, 2001.

(6) Eshelman, L. J. and Schaffer, J. D., “Real-coded genetic algorithms and intrval schemata,” *Foundations of Genetic Algorithms2*,Morgan Kaufmann Publishers, Inc., San Mateo, pp.187-202, 1993.

(7) **奥野 正寛, *ミクロ経済学*, 日本経済新聞社, 東京** 1996.

(8) Deb. K., “Construction of Test Problems for Multi-Objective Optimization,” Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, (1999), pp. 164-171.

(9) **大西晃二, : コンピュータシミュレーションによる吸排気性能予測技術の開発 , マツダ技報,** No. 6, 1988, pp. 84-93.

(10) Sharov. D. and Nakahashi K., “ Reordering Hybrid Unstructured Grid for Lower-Upper Symmetric Gauss-Seidel Computations,” *AIAA Journal*., Vol. 36, No. 3, (1998), pp. 484-486.