

Abstract

In this paper, we propose two types of parallel genetic algorithm models for multi objective optimization problems. One of them is called Divided Range Multi-Objective Genetic Algorithm (DRMOGA). The other is Master-Slave model with Local Cultivation model(MSLC) .

DRMOGA is one of the divided population models. The population is not randomly divided into sub populations. In this model, the population of GA is sorted with respect to the values of one of the objective functions and divided into sub populations in order. Therefore, the individuals that are close to each other are collected in a sub population.

MSLC is one of master slave models. Usually, in a master slave model, only the operation of evaluation is performed in a slave processor. On the otherhand in this model, a master process chooses two individuals randomly and sends to a slave processor. In a slave processor, the operation of crossover, mutation, evaluation and selection are performed. Then, a slave processor returns two individuals to a master process. After these operations, the individuals are sent to the slave processor from the master processor and renewed. The master process updates the rankings of the solutions.

To clarify the characteristics and effectiveness of these models, the proposed models are applied to knapsack problems. Thorough the numerical examples, it is become cleared that these models are suited to parallel computers and can keep the diversity of the solutions.

目 次

1	序論	1
2	GA による多目的最適化への応用	2
2.1	背景	2
2.2	多目的最適化	3
2.3	多目的遺伝的アルゴリズム	4
2.4	得られた解候補の評価方法	6
2.5	探索の終了判定	8
3	多目的遺伝的アルゴリズムの並列化	9
3.1	並列 GA モデル	9
3.2	多目的 GA における並列化	11
4	提案する並列アルゴリズム	12
4.1	背景	12
4.2	領域分散型多目的 GA	12
4.3	局所的培養型マスタースレーブモデル	13
5	数値実験	16
5.1	ナップザック問題	17
5.2	多目的遺伝的アルゴリズムの構成法	21
5.3	シミュレーション環境と設定するパラメータ	21
5.4	MSLC に対するルーレット選択とトーナメント選択の比較	22
5.5	数値計算結果	26
6	結言	50

1 序論

遺伝的アルゴリズム (Genetic Algorithm: GA) は生物の遺伝と進化を模擬した確率的探索手法の 1 つである⁴⁾。GA を用いて最適化問題を解く試みは非常に多くなされており、特に従来の最適化手法では解けなかった多峰性のある問題や離散的な問題に対してその有効性が検証されている^{1, 4, 11, 13)}。

近年、GA の持つ「集団による探索（多点探索）」を行うという特徴に注目し、GA を多目的最適化問題へ適用する試みが盛んに行われその有効性が検証されている^{3, 4, 14)}。これらの研究は一般に多目的 GA と呼ばれ、Schaffer らの VEGA¹⁴⁾ に始まり、パレート最適解集合のフロンティアを陽に取り扱う Goldberg のランキング法⁴⁾ や Fonseca らの MOGA³⁾ などが代表的な研究としてあげられる。また、玉置らの並列選択と同時に、得られているパレート最適個体を保存する手法の提案、Murata らの多目的関数にそれぞれ重みを加え单一目的として解を求める方法¹⁰⁾ などの提案も行われている。

このように、GA の多目的最適化問題に対する有効性が検証される一方で、複数の目的関数および制約条件の値を繰り返し評価する必要があり、膨大な計算時間が必要となるという問題点も指摘されている。このため、並列処理により計算時間を短縮することは重要な課題となる。特に多目的最適化問題では、目的関数の数が 3 以上になると求めるべきパレート最適解の目的関数空間上の分布形態も複雑になり、結果として求める個体数が膨大に必要となる場合が多い。よって必要となるメモリ領域の確保という意味からも多目的 GA では並列化は必要である。

单一目的における GA の並列化に関する研究は近年活発に行われている^{11, 13)}。その中でも、適合度関数の値を求める部分の並列化を行うモデルであるマスタースレーブ型モデルや母集団を幾つかのサブ母集団に分割し、それぞれのサブ母集団内で GA を行い、数世代に 1 度の割合で解交換を行う分割母集団モデル (Distributed GA:DGA) が代表的である。また、分割母集団モデルは単一母集団モデルと比較して並列化効率が良いだけでなく、解を求めるのに必要な計算量そのものが減少することが知られている¹⁾。

一方、多目的 GA の並列化に関する研究は幾つか行われているがその数は多くない^{6, 7, 15)}。また、そこで使用されている計算モデルは GA の並列化手法として最も一般的な島モデル並列 GA を基にしているものがほとんどである。

しかしながら、GA という同じ操作によって解を求める計算であっても、単一目的の場合と多目的の場合では最終的に求める解が複数の解集団である、評価が一意的に行えないなどの違いがある。特に求める解が複数の解集団であるため、母集団は全世代を通して常に多様性を保持している必要があり、個体の多様性は多目的問題を解くにあたり最も重要な要素の一つである。

单一目的における GA の場合、その探索初期段階において各個体は探索領域全体に広がり、大局的探索を行う。そして、探索が進むにれて各個体は最適な 1 つの解に収束し、徐々に局所探索を行う。そのため、少ない個体数の場合には初期収束を起こし個体数が必要以上に多い場合には余分な計算が必要となる。

それに対して多目的における GA では、パレート解集合が探索領域全体に広がっている場合が多いため、探索の全ての段階において探索領域全体における大局的探索が必要となる。同時に、各個体は

真のパレート解へ近づく必要性があるので，局所探索も必要となる．これにより，個体数は多ければ多いほど，より広い範囲での精度の良いパレート解が求められることになる．すなわち，单一目的の場合と多目的の場合とでは，GA に求められる性能が異なる．

そこで本研究では，より多目的の特性を考慮に入れた 2 つの新たな多目的並列分散モデルを提案する．2 つの提案するモデルは，それぞれ分割母集団モデルとマスタースレーブ型モデルに対応しており，それぞれの異なる特徴を持っている．

分割母集団モデルに対応するモデルとしては，パレート解候補群を目的関数空間における領域で分割して並列処理を行う領域分割型多目的遺伝的アルゴリズム (Divided Range Multi-Objective Genetic Algorithm: DRMOGA) を提案する．本モデルは，探索の各段階において得られたパレート解候補群を任意の目的関数値によって各島に分割するという方法を用いている．そのため，より広範囲のパレート最適解が得られる，近傍交叉が可能となるといった利点を持っている．

また，マスタースレーブ型モデルとしては局所的培養型マスタースレーブ型モデル (Master-Slave model with Local Cultivation: MSLC) を提案する．本モデルは，佐藤ら¹²⁾により提案された MGG モデルの考えを取り入れ，個体集団の多様性を特に考慮したモデルとなっている．本モデルは，個体の選択が局所的に行われるため全体として多様性の維持が行われやすく，従来のマスタースレーブ型モデルと比べマスターの計算負荷が軽いという特徴を持っている．また，本モデルはマスタースレーブ型モデルであるためスレーブの数による影響を受けないという特徴も持っている．

本研究では，幾つかの数値実験を行い従来の並列分散モデルとの比較を通して提案するモデルの有効性の検証を行っている．

本論文の構成について述べる．本論文は，6 章から構成されている．第 1 章は序論であり，本研究の位置づけについて説明している．第 2 章では，多目的最適化問題の数学的な定義，多目的最適化問題への GA の適用例の代表的な方法について説明している．第 3 章では，本研究の主目的である GA の並列化モデルについて幾つか代表的なモデルをとりあげ，各モデルの仕組み，及び特徴について説明する．第 4 章では，3 章において説明した代表的な並列モデルを踏まえた上で，多目的最適化問題により適した新たな 2 つの並列モデルを提案し，その特徴について説明する．第 5 章では，多目的 0/1 ナップザック問題に対して GA の最も一般的なモデルである単一母集団モデル，通常の母集団分割 GA，提案する 2 つのモデルを適用し，その結果から提案するモデルの有効性について検証を行っている．そして，第 6 章において本研究のまとめとして結論を述べている．

2 GA による多目的最適化への応用

2.1 背景

多目的最適化問題では，单一目的と異なり複数もしくは無限に存在するパレート解集合を探索することがまず第一の目的となる．従来の多目的最適化問題に対する手法として，複数の目的関数を任意の重み付けにより单一化する重みパラメータ法，任意の目的関数以外の目的関数を制約条件化し，任意目的関数の最適化に集約する 制約法などが提案されている．しかしながら，これらは複数もしくは無限にある解集合の中にある一つの解しか求めることができず，多目的最適化における目的関数間

でのトレードオフの関係をバランスさせた妥協解を得るという意味では不十分である場合が多い。

一方、GA はその特徴の 1 つとして多点探索を用いて探索を行うという特徴を持っている。そのため、GA では多目的最適化問題において直接的にパレート解集合を 1 回の試行で求めることができる。

本章では、まず多目的最適化問題および多目的最適化における解であるパレート解の数学的な定義について概説する。次に GA によるこれまでの多目的最適化への代表的な適用手法を紹介する。その中で、多目的最適化における GA の選択手法および幾つかの選択手法の中に用いられるシェアリング手法について説明を行う。その上で、GA を多目的最適化問題へ適用した際、問題となる探索の終了判定方法および得られた解の評価方法について説明する。

2.2 多目的最適化

多目的最適化 (Multiobjective Optimization Problems, MOPs) とは「複数個の互いに競合する目的関数を与えられた制約条件の中で何らかの意味で最小化する問題」と定義される。目的関数が互いに競合しあっているため、与えられた複数の目的関数に対して完全最適解を求めることはできない。そのため、多目的最適化では「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。多目的最適化では、このような解集合をパレート最適解 (Pareto optimal solution) と呼んでいる。故に、多目的最適化の 1 つの目標は、このパレート最適解 (集合) を導出することであると言える。

2.2.1 多目的最適化問題の定義

一般に多目的最適化問題は、 k 個の互いに競合する目的関数

$$f_i(x_1, x_2, \dots, x_n) \quad (i = 1, 2, \dots, k) \quad (1)$$

を、 m 個の不等式制約条件

$$g_j(x_1, x_2, \dots, x_n) \leq 0 \quad (j = 1, 2, \dots, m) \quad (2)$$

のもとで最小化するという問題として定式化される。多目的最適化問題では、一般に全ての目的関数 $f_i(x)$ を同時に最小化することはできない。実問題においては、目的関数間にトレードオフの関係が存在する場合が多い。

2.2.2 パレート最適解

パレート最適解は、多目的最適化問題における解の優越関係により定義される。多目的最適化問題における解の優越関係の定義を以下に示す。

定義 (優越関係): $x^1, x^2 \in \mathbb{S}$ とする。

a) $f_k(x^1) \leq f_k(x^2)$ ($\forall k = 1, \dots, p$) の時、 x^1 は x^2 に優越するという。

b) $f_k(x^1) < f_k(x^2)$ ($\forall k = 1, \dots, p$) の時、 x^1 は x^2 に強い意味で優越するという。

もし、 x^1 が x^2 に優越しているならば、 x^1 の方が x^2 より良い解である。従って、他のいかなる解にも優越されない解を選ぶことが合理的な方法であるといえる。次にこの優越関係に基づくパレート解の定義について以下に示す。

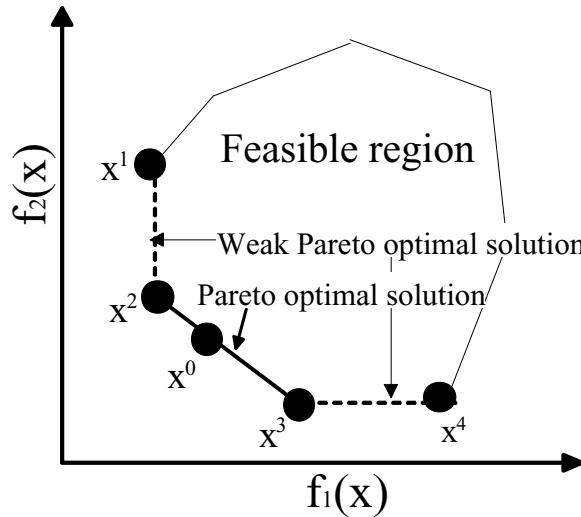


Fig. 1 Pareto optimal solution

定義(パレート解): $x^0 \in \mathbb{S}$ とする .

- a) x^0 に強い意味で優越する $x \in \mathbb{S}$ が存在しないとき, x^0 を弱パレート解という .
- b) x^0 に優越する $x \in \mathbb{S}$ が存在しないとき, x^0 を(強)パレート解という .

目的関数が 2 つ($p = 2$)の場合におけるパレート最適解の例を Fig. 1 に示す . 図中, 実線がパレート最適解を, 破線が弱パレート最適解をそれぞれ示している .

2.3 多目的遺伝的アルゴリズム

GA は自然界における生物の遺伝と進化をモデル化した, 最適化手法である⁶⁾ . 従来までの一点探索による手法と異なり, GA は多点探索であるため多峰性のある問題においても最適解を探索でき, かつ離散的な問題にも対応できる非常に強力な最適化ツールの 1 つである .

このように, GA では個体群を用いて探索が進められるので, 探索の各段階で, 個体評価における多目的性を直接扱うことが可能である . すなわち, それぞれの目的関数に対してある程度良い値をとる個体を同時に持ちながら探索を進めることができ, Fig. 1 に示されるようなパレート最適解集合を直接求めることが可能となる .

多目的最適化問題に対して GA を適用する際, 最も問題となるのが得られた解候補に対する選択方法である . 多目的最適化問題では個体が複数の評価値を持っているため, 単一目的最適化のように一意的に個体を評価することができない . 以下では, GA を用いたパレート解の生成法について個体の選択方法, 特に個体の適合度の求め方に焦点を当てて説明する .

2.3.1 GA によるパレート解生成法

多目的最適化問題における GA では, 設計領域内に遺伝子を生成し, 交叉により新たな遺伝子を発生させ何らかの方法で選択することにより, 各個体を真のパレート最適解集合に近づける . GA の各世代において優越している個体によって決定できる曲面を一般に解のフロンティアと呼ぶ . 概念とし

ては、世代が進むに従い個体の作り出すフロンティアは真のパレート曲線（最適解集合）に近づいていくものとして捉えることができる。

GAを多目的最適化問題に対して適用する場合、パレート最適解を適切に評価し、次世代に残していくことがキーポイントとなる。従来の「1つの最適解」を求める単一目的の場合と異なり、多目的では、他の解に劣っていない解（パレート解）全てが解候補となるため、単純に単一目的における適合度をそのまま適応させることはできない。その点に関して、従来

- おのの目的関数について独立に選択演算を行う（非パレート的アプローチ）
- 解の優越関係に基づいて選択演算を行う（パレート的アプローチ）

という2つ考え方に基づいて、種々の方法が提案されている。非パレート的アプローチには、J.D.SchafferのVEGA（Vector Evaluated GA）がある⁹⁾。この研究は、GAの持つ多点探索という特徴に注目し多目的最適化へ応用した初めての例である。また、パレート的アプローチとしては代表的な手法としてランキング法が挙げられる。本発表では、より多様なパレート解（特に、各目的関数のバランスを考慮するようなパレート最適解）が得られやすいランキング法を用いた。また、解の多様性をより重視するためシェアリングという手法も組み合わせて用いている。以下、ランキング法とシェアリングについて説明する。

・ランキング法

GAでは、個体の各目的関数を各世代内で相対的に評価し、個体に順番をつけることが可能である。パレートランクによる方法とは、上記におけるGAの特徴を生かし、解の優越関係に基づいて定められるランクとして適合度関数を作り、これにより選択を行うという手法である。パレートランク法は、Goldberg, Fonsecaらによる方法がある^{3,6)}。ここでは、明確に個体間の区別が行えるFonsecaによるランキング法を説明する³⁾。Fonsecaらのランキング法では、個体 X_i が n_i 個の個体に優越されているとき、 X_i のランク $r(X_i)$ を

$$r(X_i) = 1 + n_i \quad (3)$$

のように定めることにしている。この手続きによるランキング例を Fig. ??に示す。

このランキング法を用いた選択手法としては、ランクの値を適合度に変換し用いるルーレット選択、各世代でパレート最適個体（ランク1の個体）のみ残すパレート最適個体保存選択などがある。

・シェアリング

多目的最適化問題に対しGAを適用する際の重要な点は、解をパレート最適解集合上により広範囲にかつ均等に分布するような解を求めることがある。その1つの有効的な方法としてHornらによって提案されたシェアリングがある⁴⁾。以下に、シェアリングの概念を適合度に反映させる手法について述べる。

まず、各個体についてその個体の近傍の混み具合をあらかじめ与えられた関数に従ってニッヂ数として計算する。ここではニッヂ数 m_{x_i} を

$$m_{x_i} = \sum_{j=1}^N s(d(x_i, x_j)) \quad (4)$$

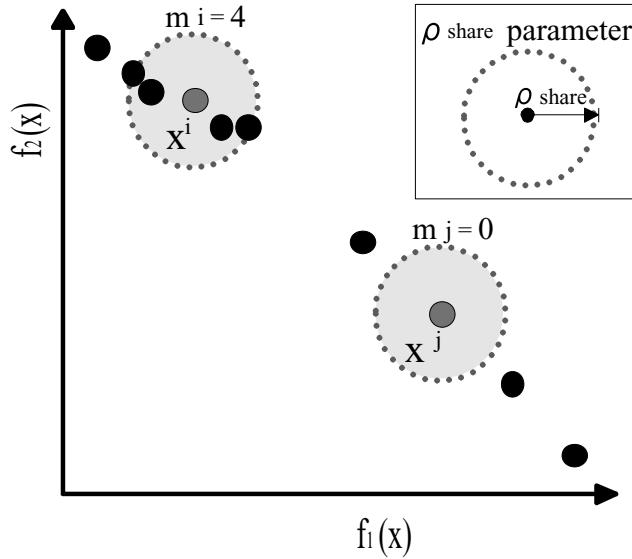


Fig. 2 Sharing

と定義しておく。 $d(x_i, x_j)$ は個体 i, j 間の距離で、その定義としては幾つかの方法が提案されている。シェアリングの適用に関し、表現空間で行うことを提案しているもの⁴⁾、目的関数空間で行うことを提案しているもの³⁾があるが、本研究では、個体 i と j の表現型でのユークリッド距離を用いるものとする。

また、 $s(d)$ はシェアリング関数と呼ばれ、例えば近傍を定めるパラメータ(シェアリング半径) $\sigma > 0$ を与え次式を用いる。

$$s(d) = \max \left(1 - \frac{d}{\sigma}, 0 \right) \quad (5)$$

このようにして算出したニッチ数 m_{x_i} でその個体の適合度 $g(i)$ を割り、それを新たな適合度 $g_s(i)$ とする：

$$g_s(i) = \frac{g(i)}{m_{x_i}} = \frac{g(i)}{\sum_j s(d(x_i, x_j))} \quad (6)$$

上式により再計算された適合度は、個体間の集中度合いも考慮に入れているため、この適合度を用いた選択を行うことにより個体が均一に分散された状態で次世代に受け継がれるものと考えられる。

本研究におけるシェアリングは、個体数の膨張を防ぐ目的で用いられている。故に、シェアリングは毎世代行うのではなく、個体数がある一定数(limit population size)を越えた段階でのみシェアリングが適用されるものとした。具体的には、シェアリングを用いて計算した適合度より、ルーレット選択を用いて任意の選択個体数(select population size)を選択するというものである。Fig. 2 に 2 次元空間におけるシェアリングの概念図を示す。

2.4 得られた解候補の評価方法

得られたパレート最適解に対する評価方法は、適用したモデルの性能を評価する上でも不可欠であるが、多目的最適化の分野においては幾つかの研究により提案されている^{6, 17)}ものの未だ確立には

至っていない。これは、単一目的の場合と異なり、多目的では解の評価のポイントとして、

- ・ 真のパレート解へどれだけ近づいているか（精度）
- ・ 真のパレート解をどれだけ被覆しているか（被覆）

の両方を考慮して判断する必要があり、この2つのポイントを適切にかつ定量的に判断するような評価方法は非常に難しいためである。特に、真のパレート最適解が未知であった場合にはこれらの評価項目を適切に評価することは非常に困難である。

本研究では、比屋根⁶⁾が提案している定量的な評価方法を参考に、幾つかの評価項目を設けて総合的に解の評価を行った。具体的な評価項目について以下で説明する。

2.4.1 誤差 (error)

真のパレート解が既知の場合、各パレート最適個体と真のパレート解とのユークリッド距離の平均は誤差とみなせる。誤差が小さいときには、パレート最適個体が真のパレート解集合に近いことを示している。ただし、この評価基準は真のパレート解が既知の場合でなければ使用できない。

2.4.2 被覆率 (cover rate)

パレート解を探索する場合、たとえ誤差が0.0であったとしても1点に集中しては良い解集合とは言えない。そのため、解のばらつきを示す指標が必要となる。その指標が被覆率である。

まず、Fig. 3に示すように（図では2目的の場合）、各目的関数の最大値および最小値を検索し、その領域を任意の数で分割する。それぞれ分割された領域の中に解が存在する場合は1、存在しない場合には0とする。これらの数値を合計し、領域の数で割ったものを被覆率とする。よってこの被覆率が1に近い方がすべての領域に解が存在していることになり、解が集中することなく全体に行き渡っていることがわかる。本研究の数値計算では分割された領域の数を50としている。また、真のパレート解が既知でない場合には、経験的に得られたパレート解を基準に各目的関数値の最大値、最小値を用いて被覆率を計算している。

2.4.3 計算時間もしくは評価関数の計算回数

計算時間や評価関数の計算回数も重要な指標のひとつである。本数値計算例では、次章で説明するようにパレートフロントを構成するパレート最適個体の進展が微少であることを終了条件として使用している。そのため、得られるパレート最適個体の誤差はどのようなモデルを選択してもほとんど変化がない場合も多いものと考えられる。パレート最適個体の誤差や被覆率に差が生じない場合には、できるだけ短い計算時間もしくは目的関数の値の計算回数ができるだけ少なく解を求めるモデルを評価する。

2.4.4 各目的関数ごとの最大値と最小値の平均

得られたパレート解集合における各目的関数ごとの最大値と最小値の平均を求ることにより、得られた解の幅広さについて評価することができる。また、この値より探索が目的関数空間におけるどの範囲に及んでいるかを推測することができるため、探索の進行具合、すなわち解の精度についてもある程度の評価を行うことができる。

特に、この値は真のパレート解が既知・未知に関わらず用いることができるという利点がある。

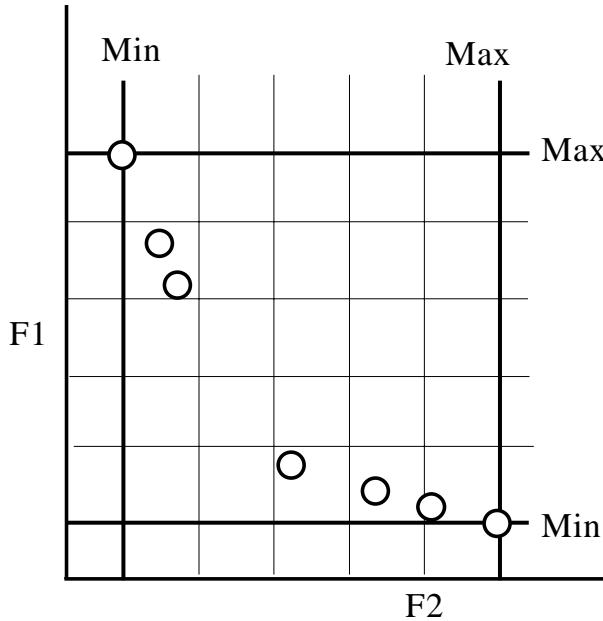


Fig. 3 Cover rate

2.5 探索の終了判定

従来までの多目的 GA に関する研究のほとんどが、あらかじめ設定された特定の世代、計算回数、計算時間を越えた段階で終了するという終了条件を用いていた。しかしながら単一目的、多目的に関わらず、探索の終了条件としてこれらの終了条件を用いるのは好ましくない。その最大の理由は、終了条件に探索の収束状況が反映されない点にある。また、個体数、交叉率、突然変異、移住率などのパラメータにより探索能力、探索の性質が大きく異なるため明示的に従来の終了条件パラメータを定めることができない点も挙げられる。

そこで、従来の世代、計算回数、計算時間とは異なる、アルゴリズムの探索状況より、終了を判断する新たな終了条件を提案する。この提案する手法では、明示的にパレート解の探索状況を扱っているため、より正確な手法の比較が行えるものと思われる。提案する終了条件を以下に記す。

任意の世代 N におけるパレート最適個体群（ランク 1 個体）を P_N 、世代 N から任意世代後の $N + T$ 世代におけるパレート最適個体群を P_{N+T} とする。この両者のパレート最適個体群を足し合わせ、その中よりパレート最適個体を抽出する。その上で、 P_N 全体の内、何個体が抽出されたのかその割合を計測し、一定以上の割合で P_N がランク 1 として抽出されていたならば、その時点により探索を終了する。

この終了条件における P_N が抽出される割合は、探索の進行状況を示す 1 つの指標であるといえる。故に、この提案する終了条件は、パレート個体の作り出すフロンティアの形成状況に基づく方法といえる。本研究では、この終了判定に用いる任意の割合を終了パラメータ（Termination parameter）として設定している。

3 多目的遺伝的アルゴリズムの並列化

GA のアルゴリズムは他の最適化手法と比較して、優れた並列性を持っている。評価、選択、交叉、突然変異などの遺伝的操作を全ての段階において並列処理することが可能である。

GA の並列化に関する研究は単一目的を中心に活発に行われており^{11, 13, 7, 15)}、多くのモデルが考案されその検証が行われている。本章では、それらのうち代表的な並列モデルを幾つか取り上げそれぞれの仕組みや特徴について述べる。

3.1 並列 GA モデル

本論文では、代表的な並列 GA モデルとして、

- ・マスタースレーブ型モデル
- ・分割母集団モデル
- ・近傍モデル

の 3 つをとりあげて説明を行う。それ以外の並列モデルについても、そのほとんどが上記の 3 つの並列モデルを変形したモデルとなっている。以下、代表的な 3 つのモデルについて説明する。

3.1.1 マスタースレーブ型モデル (Master Slave model)

このモデルでは母集団の分割は行わず、選択と交叉を大域的に行い、個体の評価のみ複数のプロセッサ (Processing Element: PE) を使用する。そのため、最終的に得られる結果は単一母集団モデルと差がなく、用いるスレーブマシンの台数による解への影響も全くない。しかし、マスタースレーブ間の通信負荷が非常に大きく、評価計算の負荷が軽い場合にはマスターのみに計算負荷が集中してしまうという欠点がある。

そのため、対象問題としては評価計算負荷が非常に高い問題に向いており、用いるアーキテクチャ環境としては複数の PE が単一のメモリを共有可能な、共有メモリー (Shared Memory) 型の並列計算機に向いていると言える。Fig. 1 に概念図を示す。

3.1.2 分割母集団モデル (Distributed GA model : DGA)

このモデルでは母集団は、用いるプロセッサ分の分割母集団 (Sub-population) に分割される。そのため分割母集団 GA もしくは島モデル (Island Model) と呼ばれることがある。

分割母集団は並列計算機上の各 PE に分配され、各々の母集団において GA が行われる。そして早熟収束の回避と多様性の維持を行うために、一定期間ごとにプロセッサ間で個体の交換を行う。一般にこの操作は移住 (Migration) と呼ばれ、プロセッサ間での個体の交換周期を移住間隔 (Migration interval) と言い、分割母集団内全体に対する移住する個体の割合を移住率 (Migration rate) と言う。移住の概念図を Fig. 2 に示す。

粗粒度並列化モデルにおいて「移住の頻度」、「移住先の選択法」、「移住個体の選択法」等、移住に関わるパラメータ、スキームが重要となる「移住の頻度」については各分割母集団が一斉に移住を行う「同期移住」や、各母集団が個別の判断により移住を行う「非同期移住」などがある。この他にも様々な移住スキームが提案されている^{9, 8)}。

本モデルは、通信負荷が低いため、プロセッサ数が多い並列機や PC クラスタのようなネットワー

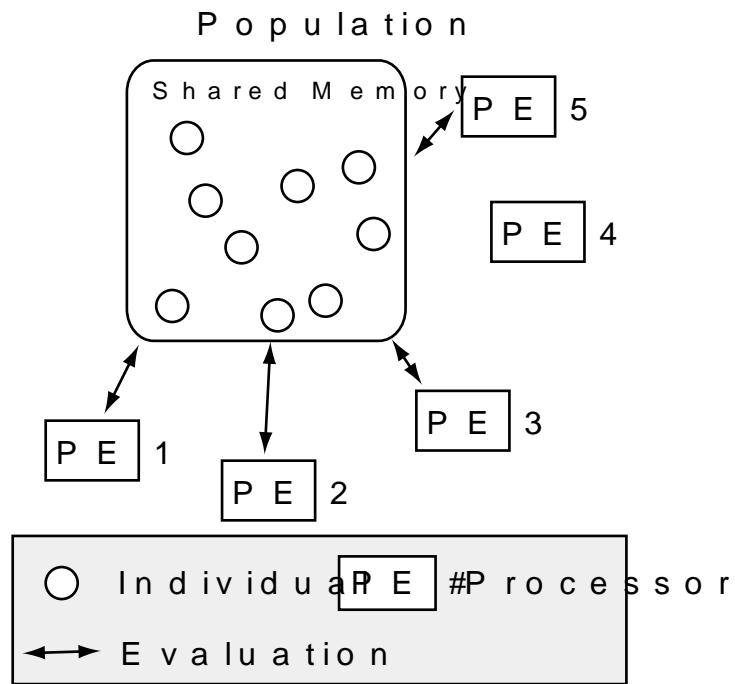


Fig. 1 Global parallelization model

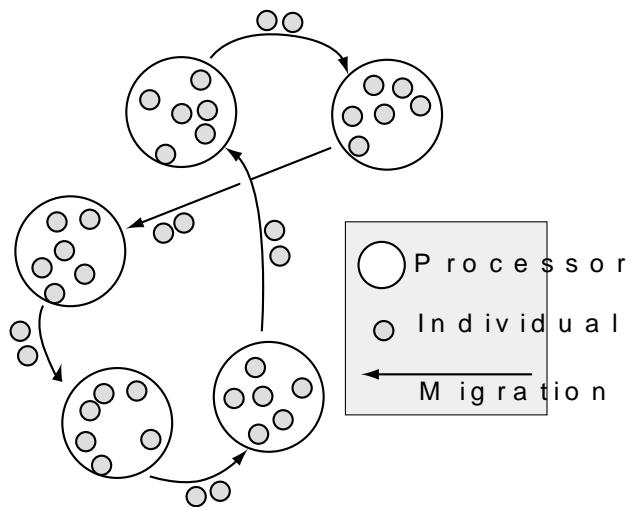


Fig. 2 Coarse-grained parallel model

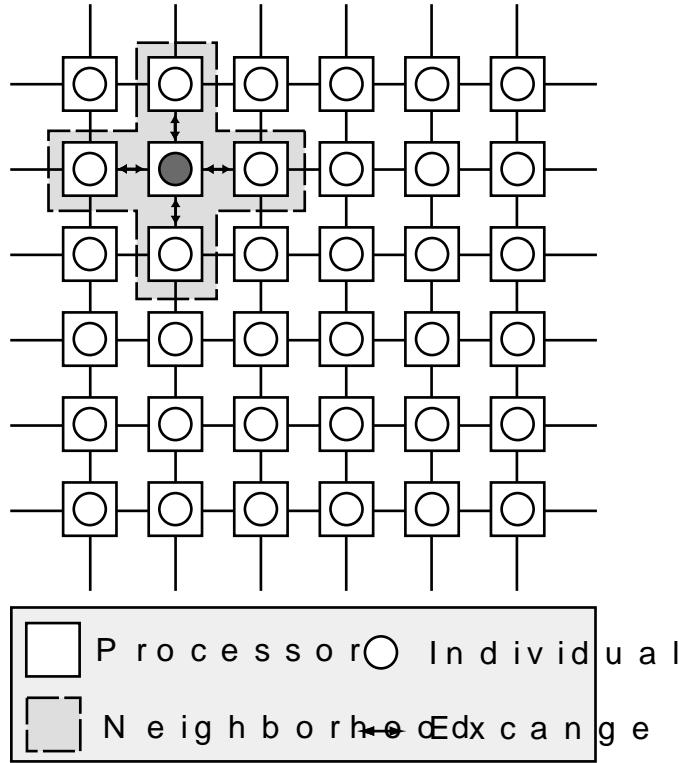


Fig. 3 Fine-grained parallel model

ク性能の低い並列計算機には適している。

3.1.3 近傍モデル (Neighborhood model)

このモデルでは、プロセッサ 1 つに割り当てる個体は 1 つまたは比較的少数であり、プロセッサ間通信のオーバーヘッドを減らすために、近傍のプロセッサの個体とのみ交叉を行う。このモデルは、個体の配置状態が 2 次元格子状空間の区分単位であるセル上に丁度配置されているように見えることから、セルラー GA とも呼ばれる。このモデルは、低い性能のプロセッサを極めて多数用いる計算機に多く見られる。

Fig. 3 にその概念図を示す。このモデルは極めて効率的ではあるが、部分集団の適当な近接構造の設定が困難であること、また、その近接構造の決定の際に並列計算機のアーキテクチャに依存することが問題とされている。

3.2 多目的 GA における並列化

一方で、多目的最適化 GA の並列化に関する研究は单一目的に比べその数は多くない。また、ここで使用されている計算モデルの多くは单一目的における GA の並列化と大差はなく、例としてマスター/スレーブ型モデルや⁷⁾、分割母集団モデル¹⁵⁾などが挙げられる。唯一の例外として、鬼頭らによるセルラー熱力学的 GA という多目的に特化した並列モデルが発表されている程度である¹⁸⁾。

しかし、单一目的の場合における探索と多目的における探索では GA の探索方法において幾つか異なる点がある。单一目的における GA の場合、その探索初期段階では、各個体は探索領域全体に広がり大局的探索を行う。そして、探索が進むにつれて各個体は 1 つの解に収束し、局所探索を行う。そ

のため，少ない個体数の場合には初期収束を起こし，個体数が必要以上に多い場合にはその分計算が必要となる．

それに対して，多目的における GA では，多くの場合，パレート解集合が探索領域全体に広がっている場合が多いため，探索の全ての段階において，探索領域全体における大局的探索が必要となる．同時に，各個体は真のパレート解へ近付く必要性があるので，局所探索も必要となる．これにより，個体数は多ければ多いほど，より広い範囲での精度の良いパレート解が求められることになる．すなわち，单一目的の場合と多目的の場合とでは，GA に求められる性能が異なる．

分割母集団モデルを多目的問題に適用した場合，各母集団内の個体数は单一母集団のものと比較すると減少するため，各島で求められるパレート解の質は悪くなる．また，各島内では優越した解であっても，全体の解集合と比較した場合には，優越していないような場合が考えられるため，計算効率も良いとは言えない．分割母集団モデルと单一母集団モデルとの比較による優位性および有効性については，第 5 章の数値実験において詳細に検討している．

4 提案する並列アルゴリズム

4.1 背景

前章において説明したように，多目的最適化では单一目的 GA で適用されてきた分割母集団モデルは单一母集団モデルと比較してあまり良好な結果を得ることができなかった．そこで，本研究では多目的 GA の特徴を活かした新たな並列分散アルゴリズムを幾つか提案しその有効性の検証を行った．

提案したアルゴリズムは 2 種類あり，分割母集団モデルの一種である領域分割型多目的遺伝的アルゴリズム (Divided Range Multi-Objective Genetic Algorithm: DRMOGA)，そして大域的並列モデルの一種である局所的培養型マスタースレーブモデル (Master-Slave model with Local Cultivation model: MSLC) である．以下，それぞれについて説明する．

4.2 領域分散型多目的 GA

多目的最適化 GA においては広範囲のパレート解を効率よく求めるためには前節で触れたように，

- 得られたパレート最適個体の近傍探索を行う能力がある
- パレート最適個体の近傍探索を必要以上に行い計算の無駄を生じない

ことが求められる．この 2 点を考慮していない通常の領域分散型モデルでは，通常の单一母集団モデルと比較しても，あまり良好な結果を得ることができない．また，通常の領域分散型モデルを用いて单一母集団モデルと同程度のパレート解集合を求めるためには，必要となる個体数が増大し，場合によつては並列処理を行う方が計算時間が長くなる場合がある．

そこで本研究では次に示すように，得られているパレート最適個体群を目的関数に沿って領域で分割し，その領域ごとに多目的最適化 GA を行う手法を提案する．

以下に提案する領域分割型多目的遺伝的アルゴリズムの流れを説明する．下記の流れの中で GA の総個体数を N ，分割数を m とし，目的関数は f_1 から f_L まで L 個存在するものとする．

- ステップ 1

N 個の個体をランダムに生成する。これらの個体が表現する設計変数は全ての制約条件を満足するものとする。

- ステップ 2

得られた個体のうちランク 1 (個体の中で優越されない個体 , すなわちパレート最適個体) のものだけを選択する。

- ステップ 3

基準となる目的関数 f_i の値に従って各個体のソートを行う。本研究では , 基準とする目的関数 f_i はランダムではなく f_1 から f_L まで順に変更することとしている。また , 基準とする目的関数の最大値 $f_i(x)$ から目的関数値順に N/m 個の個体を選択し , m 個のサブ母集団を形成する。

- ステップ 4

サブ母集団ごとに多目的最適化 GA を行う。本研究で行う多目的最適化 GA は次節で詳しく説明する。また , 各世代ごとに終了判定を行い , 条件を満たす場合には終了する。終了判定で , 条件を満たさない場合は , ステップ 5 に進む。

- ステップ 5

各母集団で多目的最適化 GA が k 世代行われたらステップ 3 に戻る。この世代数をソート間隔と呼ぶ。

本研究では , 分割数 m およびソート間隔 k はあらかじめ決定しておくものとする。Fig. 1 には 2 目的の場合に目的関数 f_1 に沿って 3 分割している概念図を示す。

この DRMOGA モデルに対して分散メモリ型並列計算機を用いて並列処理を行うことにより , 島モデルと同等の処理速度の向上や単一母集団モデルと同等 , もしくはそれ以上の解の精度を持った解集合を求めることが期待できる。

4.3 局所的培養型マスタースレーブモデル

DRMOGA は分割母集団モデルの多目的に特化したモデルであるのに対して , 局所的培養型マスタースレーブモデルは大域的並列モデルを多目的に特化させたモデルであるといえる。一般的なマスタースレーブモデルでは , マスター ノードは基本的に個体評価以外の GA オペレータ全てを担い , スレーブ ノードは個体の評価のみに専念する。そのため , このモデルは評価関数計算の負荷が高い場合には有効であるが , 逆に負荷の低い場合には通信による負荷が高くなる上 , マスター ノードに負荷が集中するためあまり有効でないことが指摘されている。しかし , 我々の提案する局所的培養型マスタースレーブモデルは , 基本的に GA オペレータは全てスレーブ ノードで行うため , マスター ノードは個体の管理のみの役割を担っている。

局所的培養型マスタースレーブモデルは , 佐藤ら ¹²⁾ により提案された MGG モデル (Minimal Generation Gap model) の考え方を取り入れ個体集団の多様性を特に考慮したモデルとなっている。

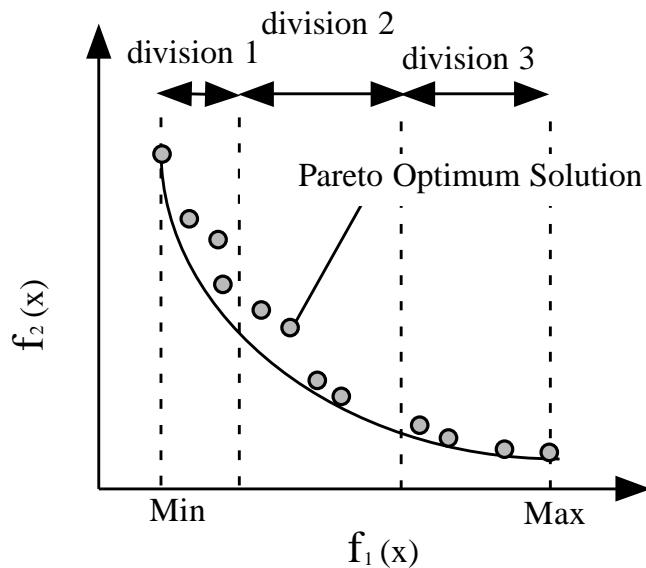


Fig. 1 DRMOGA (two objectives)

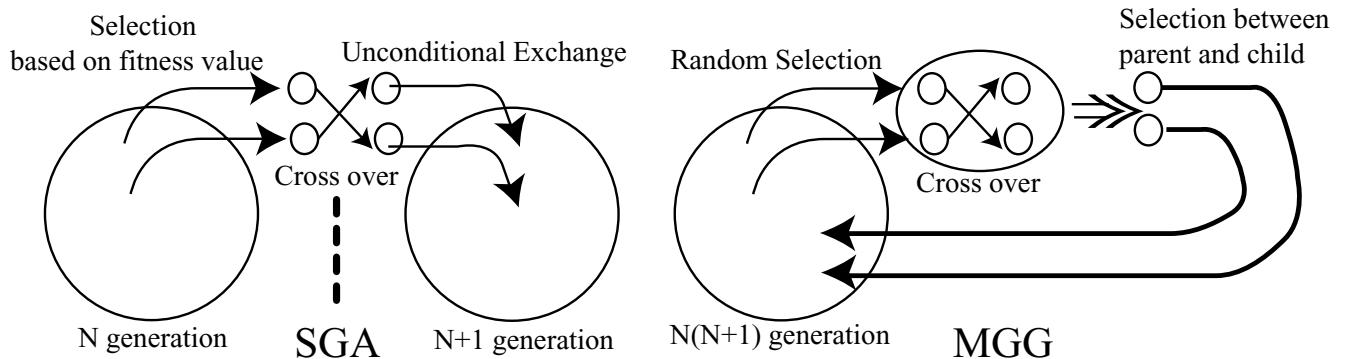


Fig. 2 Concept of MGG model

MGG モデルは、世代毎の個体の変化を 2 個体のみに限定し個体の変動をより緩やかに行うことにより、高い多様性維持機能を実現させようとするものである。単一目的においては、その有効性が既に検証されている。MGG モデルの概念図を Fig. 2 に示す。

MGG モデルにおける高い多様性の維持は、個体の選択を局所的に行うことにより実現されているため、多目的 GA においても十分その性能を発揮することが期待できる。しかも、MGG モデルは、個体の交叉、突然変異、選択といったオペレータが全て 2 個体のペアにより独立に行われるため並列性も高いと思われる。

そこで、我々は MGG をマスタースレーブ型並列モデルとして並列化し、多目的問題へ適用しその有効性を検証した。

本研究では、対象を多目的としたため個体の評価、すなわちランク付けにおいて本モデルをそのまま適用することはできない。なぜならば、本モデルにおける各スレーブノードの個体情報は基本的にマスター ノードより送られてきた 2 個体、さらに送られてきた個体を基に交叉によって生成した子個体の 2 個体しかない。そのためスレーブノードにおける個体だけでは、適切なランク付けを行うこと

ができないのである。そこで適切なランク付けのため、各ノードは個体全体もしくは個体全体を近似した情報を持っておく必要がある。そこで本提案手法では、毎世代ごとマスター ノードにおいて全個体からランク 1 の個体を抽出し、ランク 1 個体の持っている評価関数値のみを集約し各ノードへ配信するという作業を行っている。

具体的なアルゴリズムをマスター ノードとスレーブ ノードの場合に分けて以下に示す。

- マスター ノード

- ステップ 1

N 個の個体をランダムに生成する。これらの個体が表現する設計変数は全ての制約条件を満足するものとする。

- ステップ 2

生成した個体を全て評価した上で、各スレーブごとに生成、評価した染色体を全スレーブ ノードから受信して集める。

- ステップ 3

全個体よりランク 1 の個体の目的関数値のみを集約し、各ノードに分配する。これは、各ノードにおいて選択時に個体比較に基づいた選択を行うために必要だからである。

- ステップ 4

個体を 2 個体ずつペアで、非復元抽出し各ノードに配る。

- ステップ 5

各ノードから 2 個体のペアを受け取り、ステップ 4 で選んだ 2 個体のペアと入れ替える。全ての個体が配り終えるまでステップ 4、ステップ 5 を繰り返す。

- ステップ 6

終了条件を満たすかどうか判定を行う。終了条件を満たさなければステップ 3 へ戻る。

- スレーブ ノード

- ステップ 1

N 個の個体をランダムに生成する。これらの個体が表現する設計変数は全ての制約条件を満足するものとする。

- ステップ 2

生成した個体を全て評価した上で全ての染色体をマスター ノードへ送る。

- ステップ 3

全個体の内、ランク 1 の個体の目的関数値の情報をマスターより受け取る。

- ステップ 4

2 個体の個体ペアを受け取る。

- ステップ 5

受け取った 2 個体を用いて交叉、突然変異、選択などの各種 GA 操作を行う。

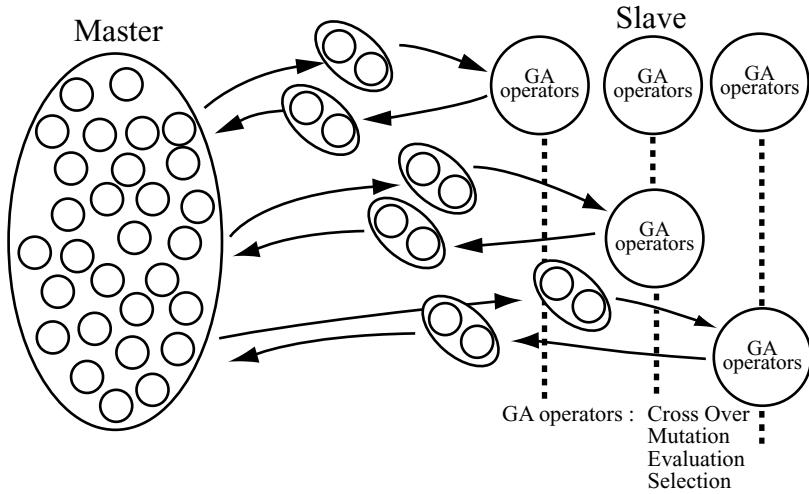


Fig. 3 Master-Slave model with Local Cultivation model

– ステップ 6

選択によって選ばれた 2 個体のペアをマスターへ送り返す。マスターから送信終了のメッセージが届くまでステップ 4, ステップ 5 を繰り返す。

– ステップ 7

終了条件を満たすかどうか判定を行う。終了条件を満たさなければステップ 3 へ戻る。

上記のように、大部分の GA オペレータをスレーブノードが行うため、従来のマスタースレーブ型に比べマスター ノードの高負荷を軽減することができる。本提案手法における主な特徴を以下に示す。

- プロセス数による影響が少ない。
- GA オペレータは基本的に全てスレーブノードが行う。
- 従来のマスタースレーブモデルに比べマスター プロセスに対する負荷が軽い。
- 個体の棲み分けを実現することができる。

上記におけるプロセス数による影響が少ないというのは、例えば分割母集団モデルでは用いるプロセス数によって島内における個体数が変化するため解へ大きく影響する可能性がある。対して局所的培養型マスタースレーブモデルでは、基本的に何プロセス用いても総仕事量およびその内容に変化がないため解への影響は無い。本モデルの概念図を Fig. 3 に示す。

5 数値実験

本章では、提案した手法を実際に幾つかの対象問題へ適用し、従来手法との比較を通じて提案手法の有効性の検証を行う。まず、本実験における対象問題として取り上げた多目的 0/1 ナップザック問題について説明する。多目的 0/1 ナップザック問題は 0/1 ナップザック問題を単純に多目的化した問題となっており、問題自体は非常に単純な問題である。今回は、荷物数の異なる 3 つの多目的 0/1 ナップザックに対して実験を行った。

次に、本数値実験に用いた GA の構成、シミュレーション環境と設定する GA パラメータについて説明を行う。本実験では計算環境として 16 ノードからなる PC クラスタ環境を用いた。また、各手法の比較を行う前に局所的培養型マスタースレーブモデルにおける適切な選択手法についての考察を行っている。局所的培養型マスタースレーブモデルは、他のアルゴリズムと異なり局所選択を行っている。そのため、他のアルゴリズムと異なり選択の中に確率的要素を取り入れているルーレット選択では探索の無駄が生じてしまう可能性が高いのである。

その上で、各アルゴリズムにおける数値結果とそれに基づいた考察を行い、多目的並列 GA における提案手法の有効性の検証を行っている。具体的には、次の 4 つの手法

- ・ 単一母集団 GA (SGA)
- ・ 分割母集団モデル (DGA)
- ・ 領域分散型 GA (DRMOGA)
- ・ 局所的培養型マスタースレーブモデル (MSLC)

をそれぞれ対象問題に対して実験し、得られた結果の考察を行った。提案する手法のうち領域分散型 GA を DRMOGA モデル、局所的培養型マスタースレーブモデルを MSLC モデル、また、単一母集団モデルを SGA モデル、島モデルを DGA モデルとそれぞれ表記する。提案手法の有効性の検証の後、分割母集団モデルにおけるプロセス数と解との関係および多目的 GA の並列手法である DGA、DRMOGA、MSLC の並列化効率について考察を行う。

5.1 ナップザック問題

5.1.1 問題の定式化

一般に、0/1 ナップザック問題は、荷物 (item) のセットから成り立っている。各荷物には重さと利益が付随している。そして上限制約としてナップザックの容量がある。その目的は、荷物全体を総和した利益が最大になるような荷物の組み合わせを見つけることである。ただし、選んだ荷物の組み合わせは全てナップザック内に収まる。つまりトータルの重さは、ナップザックに定められた許容量を超えてはいけない。

この単一目的問題は、ナップザックの数を任意数にすることによって直接的に多目的問題へ拡張することができる。ここで多目的 0/1 ナップザック問題は、以下の式によって定式化される。

$$\begin{aligned}
 p_{i,j} &= \text{ナップザック } i \text{ に関する荷物 } j \text{ の利益} \\
 w_{i,j} &= \text{ナップザック } i \text{ に関する荷物 } j \text{ の荷物の重さ} \\
 c_i &= \text{ナップザック } i \text{ の許容重量,}
 \end{aligned}$$

設計変数値であるベクトル $\mathbf{x} = (x_1, x_2, \dots, x_m) \in 0, 1^m$ (m は任意の正の整数) は以下の制約条件を満たさなければならない。

$$\forall i \in 1, 2, \dots, n : \sum_{j=1}^m w_{i,j} \cdot x_j \leq c_i \tag{7}$$

その上で，次式によって求まる $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ の最大化を目的とする．

$$f_i(x) = \sum_{j=1}^m p_{i,j} \cdot x_j \quad (8)$$

(尚，上式において $x_j = 1$ ならば荷物 j は選択された荷物であることを意味する)

多目的ナップザック問題は多目的における多くの研究に用いられている代表的なテスト関数の1つであり，特に離散的な問題における良質なテスト関数として知られている^{17, 16)}．

尚，本研究では使用例題として100荷物2目的，250荷物2目的，750荷物2目的の3つのナップザック問題を用いた．今回用いた例題の内，100荷物2目的問題をTable 1に，250荷物2目的問題をTable 2,3に示す．

Table 1 100item 2 Knapsack problems

1 Knapsack($capacity \leq 2732$)

weight	94	74	77	74	29	11	73	80	81	82	75	42	44	57	20	20	99	95	52	81
profit	57	94	59	83	82	91	42	84	85	18	94	18	31	27	31	42	58	57	55	97
weight	68	16	79	30	16	90	21	49	70	78	77	21	84	19	65	38	25	43	99	75
profit	79	10	34	100	98	45	19	77	56	25	60	22	84	89	12	46	20	85	42	94
weight	80	10	44	26	21	74	20	22	81	89	15	35	24	16	43	75	25	76	48	75
profit	20	65	27	34	27	91	17	56	23	89	18	11	91	79	14	99	45	73	81	96
weight	15	23	10	81	81	67	58	77	49	16	65	74	14	41	74	74	17	12	95	29
profit	51	96	63	40	93	87	71	54	74	15	32	57	70	62	12	71	57	97	48	33
weight	75	61	59	37	75	90	17	79	15	88	76	93	98	80	33	39	96	71	39	49
profit	42	15	39	91	17	63	81	49	60	90	87	25	15	30	76	76	53	59	40	59

2 Knapsack($capacity \leq 2753$)

weight	55	10	97	73	69	23	62	47	90	62	96	88	95	61	94	16	91	61	27	18
profit	20	19	20	66	48	100	13	87	62	73	53	79	17	93	78	22	85	86	56	56
weight	96	77	14	36	17	56	83	41	52	69	97	45	94	45	47	28	82	13	82	39
profit	44	86	94	93	57	31	20	35	70	79	58	24	84	12	17	43	35	47	92	38
weight	28	41	55	12	50	32	97	87	36	11	20	37	87	91	19	22	89	54	20	78
profit	93	50	27	100	36	30	23	22	56	73	55	32	75	42	82	80	55	48	93	28
weight	52	35	18	96	54	10	39	17	51	40	25	84	54	31	97	37	63	39	60	87
profit	26	42	96	93	16	39	46	80	24	87	37	73	81	38	98	13	91	85	17	59
weight	63	37	13	31	84	90	84	57	21	64	63	21	95	83	81	45	68	89	100	14
profit	58	56	93	66	64	17	10	33	28	97	25	42	17	23	37	46	52	33	26	90

Table 2 250item 2 Knapsack problems ~ 1 Knapsack($capacity \leq 6536$) ~

weight	79	25	99	41	94	75	40	59	95	99	95	93	74	83	14	89	41	53	75
profit	100	49	54	12	78	30	65	31	90	50	80	84	53	72	94	60	74	58	17
weight	70	49	19	84	16	25	66	68	79	16	93	14	20	99	93	100	56	62	69
profit	24	45	19	33	39	91	69	48	56	84	66	60	95	49	63	22	71	83	36
weight	91	52	86	43	76	45	21	57	45	41	58	33	68	22	66	83	50	64	35
profit	15	64	28	49	41	14	17	28	65	27	65	28	21	17	33	68	51	91	97
weight	42	33	66	38	57	69	90	79	89	28	70	38	71	46	41	49	43	36	68
profit	28	91	82	54	12	47	33	30	65	56	57	91	88	77	99	29	23	39	86
weight	92	33	84	90	22	93	92	89	47	23	13	11	34	11	60	82	10	16	81
profit	12	85	22	64	33	95	32	10	44	56	86	38	72	10	57	30	28	92	68
weight	20	85	52	92	31	24	83	76	15	38	63	32	99	39	97	28	42	80	21
profit	54	34	93	35	41	46	82	66	15	89	70	46	43	60	36	63	18	67	13
weight	99	11	91	24	45	28	89	31	81	78	48	72	36	77	84	40	65	43	49
profit	12	94	19	65	59	28	91	37	24	86	23	100	38	21	42	58	91	26	69
weight	23	82	41	22	67	96	58	56	81	42	55	31	38	33	10	43	84	52	96
profit	72	89	34	84	30	24	62	20	45	11	24	54	61	14	97	79	65	68	93
weight	92	51	50	63	70	46	30	45	41	15	82	52	25	69	62	26	64	93	78
profit	54	36	41	62	90	10	58	57	46	85	19	15	15	44	70	17	25	24	37
weight	82	71	65	40	25	20	24	19	10	12	31	22	96	79	31	98	80	10	31
profit	45	97	85	64	46	74	86	89	12	83	24	40	99	27	28	65	57	66	75
weight	42	99	26	21	83	78	66	54	100	22	48	16	25	97	95	70	22	20	16
profit	50	42	34	12	59	100	12	61	98	14	68	64	46	87	68	33	43	28	24
weight	17	19	49	90	44	53	18	42	26	87	55	58	32	61	32	10	11	12	61
profit	25	29	13	59	20	66	84	16	40	69	55	73	85	25	17	81	32	74	90
weight	96	59	94	25	30	70	53	35	54	91	51	20	87	70	50	52	31	43	32
profit	88	11	98	59	12	93	16	97	37	59	57	16	73	74	65	12	33	28	66
weight	25	40	78																
profit	17	88	86																

Table 3 250item 2 Knapsack problems ~ 2 Knapsack($capacity \leq 6489$) ~

weight	17	63	65	19	79	11	83	88	29	75	83	47	77	19	43	30	89	70	48	39
profit	40	16	42	50	99	12	46	43	97	28	41	15	36	96	69	24	36	21	54	38
weight	17	54	28	73	71	99	50	19	55	25	43	10	49	83	57	41	42	80	33	19
profit	43	17	19	48	81	50	28	34	51	98	24	61	11	78	86	89	45	20	16	74
weight	78	61	50	56	58	36	44	82	49	90	46	37	31	98	51	48	14	94	36	58
profit	11	92	99	92	15	50	54	30	97	85	19	11	78	73	47	96	47	74	96	19
weight	44	52	65	98	39	22	12	21	17	88	93	16	86	41	10	10	32	45	13	87
profit	12	34	78	26	30	95	99	68	84	28	79	61	56	99	95	27	16	38	28	38
weight	39	70	49	45	11	16	55	15	42	80	43	42	31	61	41	21	51	55	43	41
profit	85	62	62	18	65	83	65	86	76	55	74	94	80	91	71	20	42	90	100	75
weight	15	76	72	24	32	11	90	78	20	21	11	68	31	18	14	19	65	14	11	83
profit	99	32	17	59	10	88	55	95	44	41	69	10	41	83	10	62	75	24	26	55
weight	22	60	90	21	44	93	50	66	41	90	88	63	42	89	50	53	39	64	39	41
profit	34	25	60	72	78	64	62	93	97	92	21	36	99	94	65	27	96	69	76	32
weight	33	79	46	39	78	95	56	41	24	68	90	12	10	33	30	72	74	13	71	77
profit	75	43	55	77	13	64	65	50	16	73	96	66	43	15	76	58	86	29	26	20
weight	66	30	87	69	16	76	94	33	94	23	92	59	66	99	88	38	54	46	32	76
profit	31	26	54	38	75	21	30	93	58	82	92	24	32	34	41	36	25	54	55	77
weight	67	27	15	84	73	84	78	81	56	45	23	59	91	70	86	44	35	83	26	87
profit	67	11	56	74	51	32	53	54	92	53	23	34	91	27	25	75	13	65	47	52
weight	57	55	80	22	62	20	47	69	11	38	59	17	35	25	39	58	67	31	83	17
profit	51	55	27	78	74	73	85	10	41	70	92	67	30	11	54	34	96	32	87	65
weight	82	53	53	55	81	16	16	39	34	60	95	56	85	82	47	88	100	77	82	75
profit	12	39	21	98	24	19	60	77	10	18	96	23	93	19	73	93	16	60	77	54
weight	95	74	84	18	17	17	58	36	60	59										
profit	78	50	94	15	90	54	78	90	36	36										

5.2 多目的遺伝的アルゴリズムの構成法

本研究では対象問題として多目的 0/1 ナップザック問題を用いた。本適用問題は、荷物の数によってその難易度が大きく異なるものの、問題自体は非常にシンプルで制約も緩いため最も一般的と思われる GA の構成を用いた。

5.2.1 個体の表現方法

通常の GA と同様、0, 1 からなるビット列を用いた。具体的には、問題の荷物の数分のビット列を用意し、遺伝子のビット位置とアイテム番号の位置を適合させることにより遺伝子の持つアイテム情報を表現した。

5.2.2 交叉

本適用問題は、0, 1 のビット列により個体が表現される非常にシンプルな問題であるため、最も一般的な 1 点交叉を用いた。ただし、親個体同士のビット異なりが 1 点以下の場合には、交叉を行っても生成される子個体は親個体と全く同一になってしまふため交叉を行わないものとした。

5.2.3 突然変異

突然変異には、ビット反転を用いた。また、ランクが 1 である個体に対しては突然変異を行わないように設定した。これは個体の適合度改悪を防ぐためである。

5.2.4 選択

選択では、各世代におけるランク 1 の個体のみを選択するパレート保存戦略を用いた⁵⁾。しかし各世代における個体数は一定であるため、この一定数を超えた場合、シェアリングにより各個体に適合度を与えルーレット選択を行い、この一定数以下の場合、ランクに基づくルーレット選択を行った。また、手法によってはルーレット選択ではなくトーナメント選択を用いた。ルーレット選択とトーナメント選択の比較については、5.4 節にて比較実験を行っている。

5.2.5 制約外への個体の対応

本手法では、初期個体発生時において定義域外の個体は認めていない。つまり初期生成の段階において制約を満たす個体数が必要個体数となるまで個体の生成を行っている。しかし、交叉もしくは突然変異といった GA オペレータにより個体は定義域を越える場合もある。今回用いたナップザック問題では重量の制約があるため、この重量を超えるような目的関数値を持つ個体については何らかの変更が必要となる。そこで、今回は問題の特性を考え、制約を満たしていない個体については詰め込むべき荷物をランダムに 1 つずつ減らすという方法を用いた。

5.3 シュミレーション環境と設定するパラメータ

数値計算例で使用した並列計算機は Table 4 に示すような PC クラスタである。ネットワークは一般的な FastEthernet および安価な Switching Hub を使用している。

Table 5 に本研究で提案する DRMOGA, MSLC およびそれと比較する SGA と DGA において使用したパラメータをまとめて示す。分散モデルの場合、使用するプロセッサ数は分割数(島数)と等

Table 4 Cluster system

CPU	Pentium II (Deschutes) 500MHz *16
Memory	128 Mb
OS	Debian GNU/Linux 2.2 (kernel 2.2.17)
Network	FastEthernet
	TCP/IP
Communication library	MPICH1.2.1

Table 5 Used parameters

	SGA	DGA	DRMOGA	MSLC
Crossover rate		1.0		
Mutation rate		0.0		
Number of islands	-	16		
Migration interval (sort interval)	-	10		-
Migration rate	-	0.1	-	-

しい。

多目的最適化 GA において得られるパレート最適個体の精度に影響するパラメータは使用する個体数とシェアリング半径と考えられる。これらのパラメータは多くの手法で解に大きく影響することが報告されており²⁾、最適な値を求ることは非常に難しい。そこで、本数値計算例では個体数として 400, 4000 の値を、また、シェアリングレンジとして、それぞれ 100, 500 を使用している。シェアリングレンジとはシェアリング半径を決定するサブパラメータであり、パレート最適個体のうちでユークリッド距離の最も離れた 2 個体の距離をシェアリングレンジで分割することによりシェアリング半径を求めている。すなわち、シェアリングレンジが大きくなるとシェアリング半径は小さくなる。

5.4 MSLC に対するルーレット選択とトーナメント選択の比較

各アルゴリズムの結果について考察を行う前に、MSLC における適切な選択手法についての考察を行う。MSLC は、4 章で説明したように全ての GA オペレータはマスターから渡された 2 個体間で行われる。そのため選択は、親 2 個体、子 2 個体を合わせた 4 個体から 2 個体が選択されるという局所選択が行われる。

このような選択対象の数が限られた局所選択においては、トーナメント選択のような個体の適合度をそのまま直接反映する選択方法の方が、ルーレット選択のような選択の仕組みに確率的な要素を含む手法よりも良いと思われる。特に MSLC では、アルゴリズム自体に選択の多様性が保たれる仕組み

が組み込まれているため、これ以上確率的な要素を含め多様性を保持する必要はないと思われる。

そこで、以下では MSLC におけるルーレット選択を用いた場合とトーナメント選択を用いた場合の結果を比較し、選択手法の与える解への影響について考察する。

本数値実験では対象問題として、100 荷物 2 目的、250 荷物 2 目的、750 荷物 2 目的の 3 種類の多目的 0/1 ナップザック問題に対して、個体数として 400 個体、4000 個体を用いて数値実験を行った。得られた結果の内、400 個体を用いた場合の被覆率の結果を Fig. 1 に、4000 個体を用いた場合の結果を Fig. 2 に示す。また、各目的関数ごとの最大値と最小値の平均についてまとめたものを Table 6 に示す。さらに、実際に得られたパレート解のプロットを各問題ごと、用いた個体数ごとに Fig. 3 ~ Fig. 8 に示す。尚、これらのパレート解のプロットは、10 試行平均のうち最も被覆率が平均被覆率に近かった場合におけるパレート解のプロット図である。また、今回適用した多目的 0/1 ナップザック問題は、真の解が未知であるため、評価項目の内、誤差に関してのデータは得られなかった。

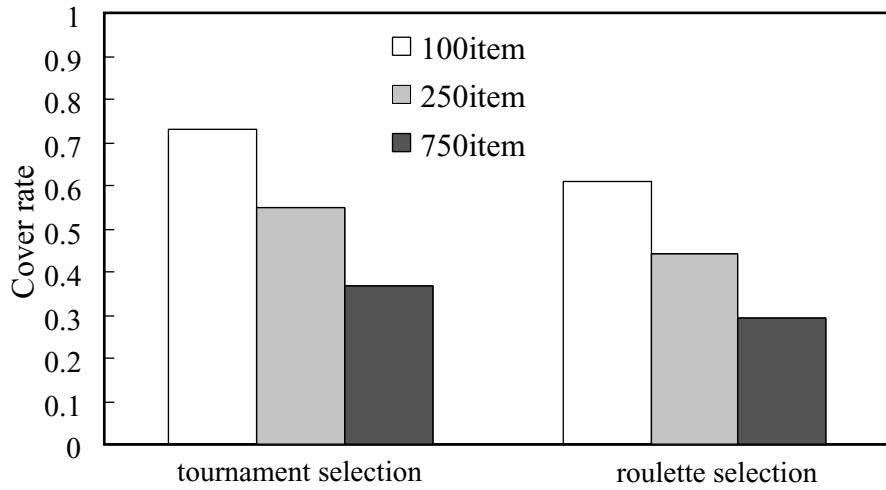


Fig. 1 Cover rate (comparison of MSLC's selection methods, population size=400)

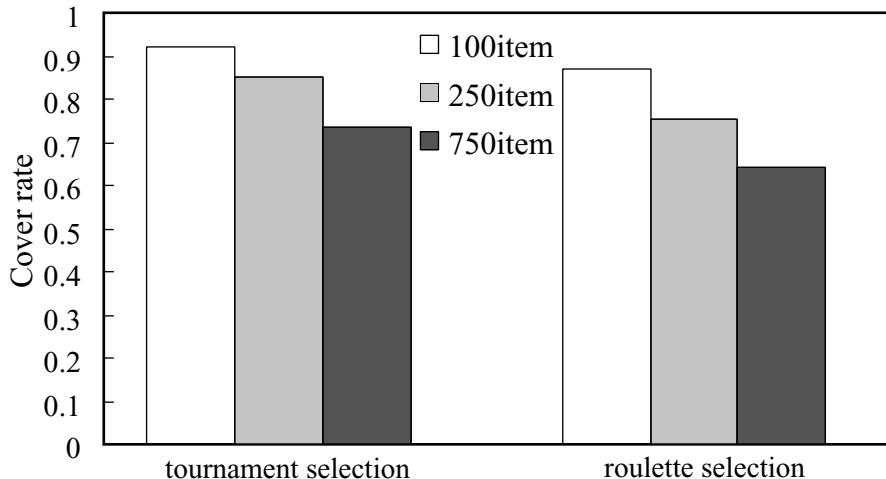


Fig. 2 Cover rate (comparison of MSLC's selection methods, population size=4000)

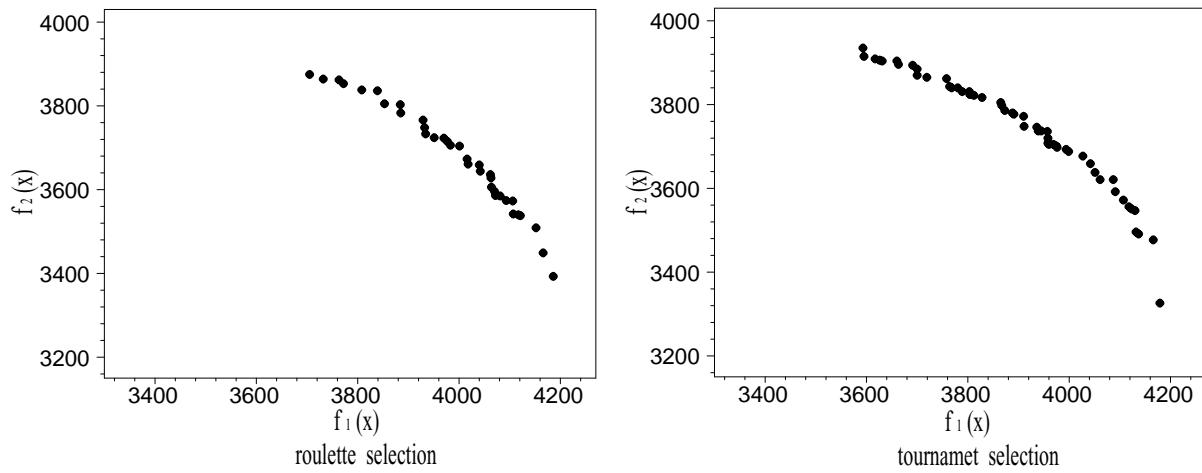


Fig. 3 Pareto optimum individuals (100items problem, population size=400)

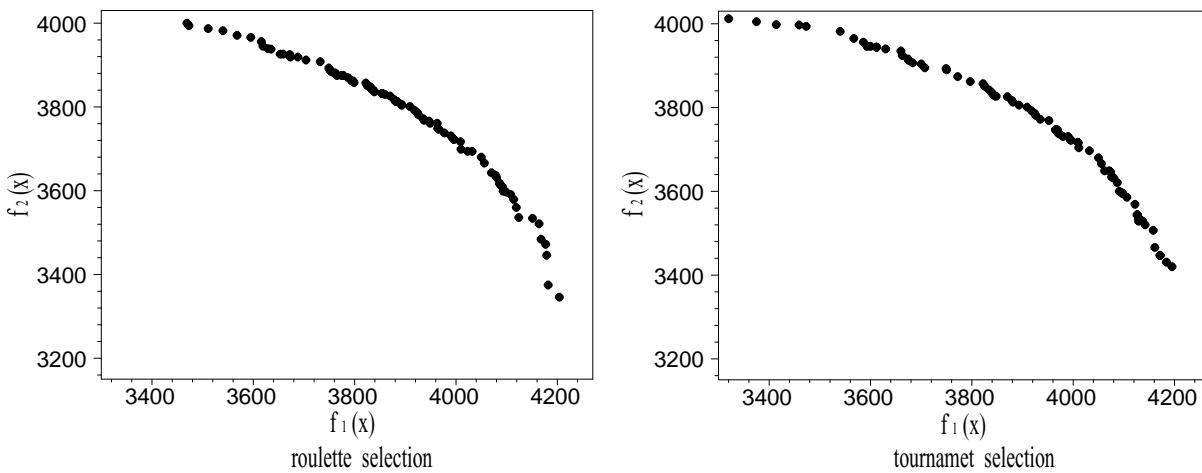


Fig. 4 Pareto optimum individuals (100items problem, population size=4000)

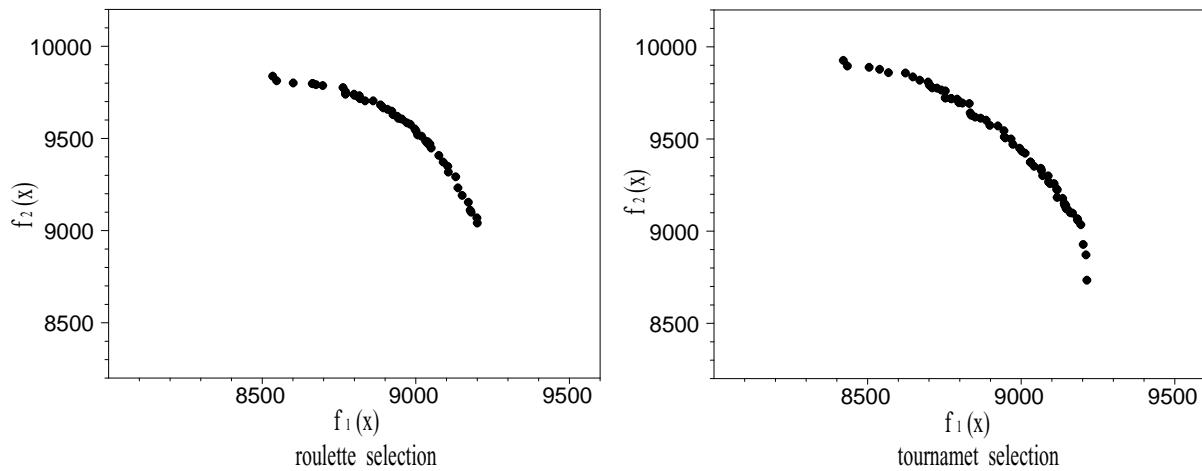


Fig. 5 Pareto optimum individuals (250items problem, population size=400)

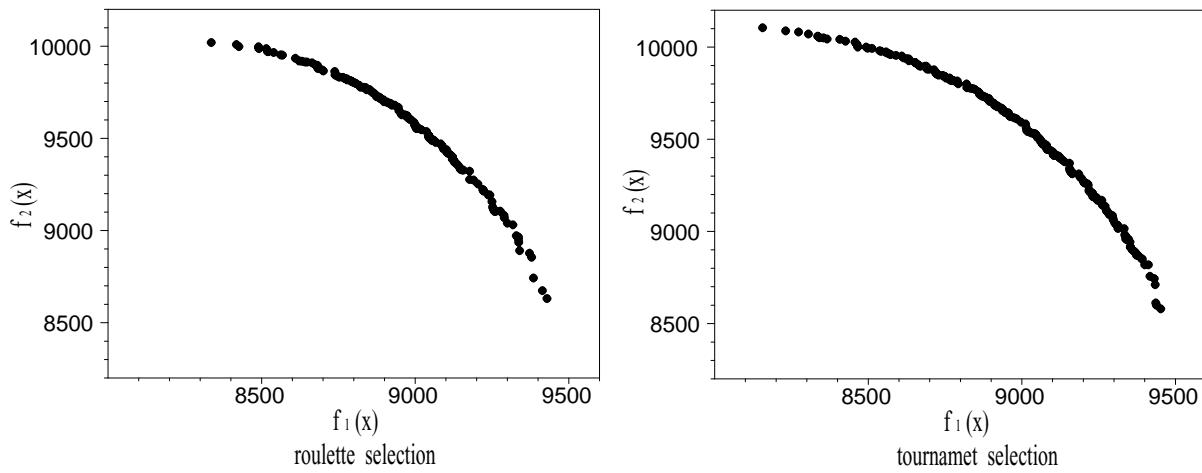


Fig. 6 Pareto optimum individuals (250items problem, population size=4000)

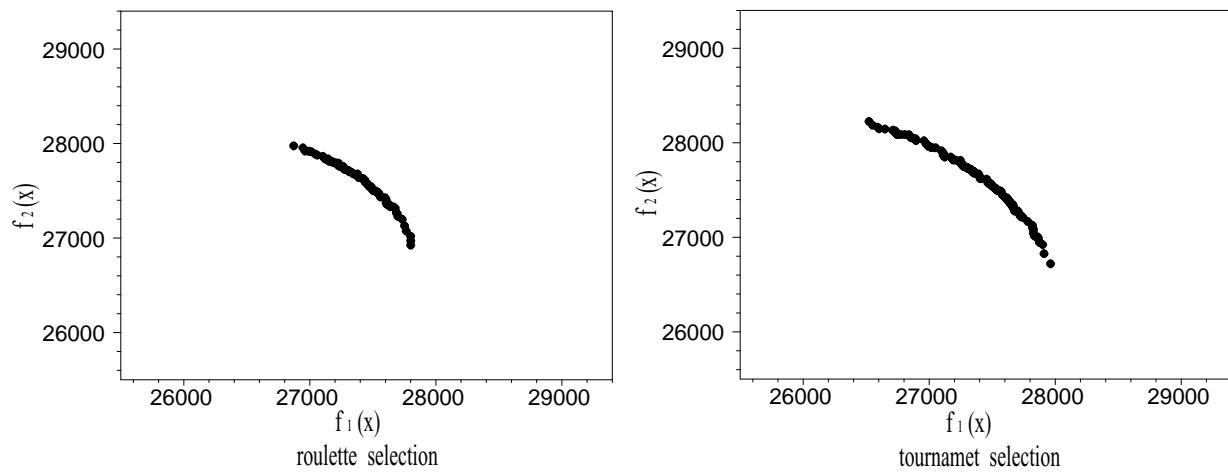


Fig. 7 Pareto optimum individuals (750items problem, population size=400)

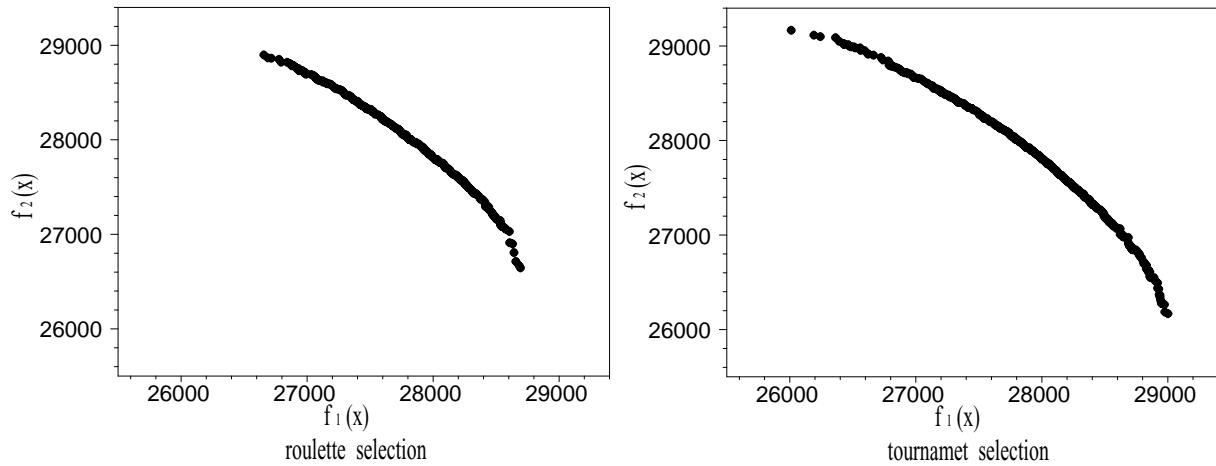


Fig. 8 Pareto optimum individuals (750items problem, population size=4000)

まず，解の幅広さについて見た場合，特に個体数が少ない場合には，Fig. 1 から分かるように全ての問題に対してトーナメント選択を用いた場合の方が大きく勝っている．また，個体数が多い場合についても，Fig. 2 から分かるように比較的問題の難しい 250 荷物，750 荷物の問題では，トーナメント選択はルーレット選択に比べ非常に良好な結果を示している．これらの結果は，各目的関数ごとの最大値と最小値の平均について示す Table 6 とほぼ同様の傾向を示しており，トーナメント選択を用いた場合の方がルーレット選択を用いた場合に比べより幅広い解が得られていると言える．

次にパレート解の進行具合，すなわち真のパレート解へどれだけ近づいているかという観点から考察を行う．得られたパレート解 Fig. 3 ~ Fig. 8 を見た場合，全ての問題，個体数においてトーナメント選択を用いた場合の方がより探索の進行方向へ進んでいる様子が分かる．この傾向は特に 250 荷物，750 荷物といったより難易度の高い問題においてより顕著に現れている．

以上より，パレート解の幅広さ，解への進行具合の両側面から MSLC では，ルーレット選択よりもトーナメント選択を用いた方が良好な結果が得られることが分かった．

また，本論文では述べていないものの他のアルゴリズムでは，MSLC の場合におけるような傾向は一切認められなかった．これは，通常のアルゴリズムでは個体の選択は一島内における全ての個体を対象に行われるためである．トーナメント選択を用いた場合には，ルーレット選択と異なり僅かな個体ランクの違いも厳密に選択に反映される．そのため，一島内における全ての個体を対象に選択を行うアルゴリズムの場合，個体集団が世代ごとにあまり変化しなくなっている．

対して MSLC では，個体の選択は 2 個体のペアによって行われるため，トーナメント選択を用いても個体集団全体が世代ごとにあまり変化しないという事はない．逆に，個体ランクの僅かな違いも選択に反映させるトーナメント選択を用いることにより，探索の無駄が減少すると共に，探索が進む機会（良好な遺伝子情報を持った個体）を逃すことが無くなるため解の精度は良くなる．

5.5 数値計算結果

本実験では，先ほどの MSLC における選択手法の比較の場合と同様，対象問題として，100 荷物 2 目的，250 荷物 2 目的，750 荷物 2 目的の 3 種類の多目的 0/1 ナップザック問題をとりあげた．この 3 つの対象問題に対して SGA，DGA，DRMOGA，MSLC の 4 つの手法を個体数 400 個体，4000 個体の場合について数値実験を行った．

得られた結果の内，被覆率について，400 個体の場合を Fig. 9 に 4000 個体の場合を Fig. 10 に示す．各目的関数ごとの最大値と最小値の平均に関しては，各問題の各目的関数ごと，用いた個体数別に分けて Fig. 11 ~ Fig. 21 に示す．さらに，最終的なパレート解の図を各問題，用いた個体数ごとに Fig. 23 ~ Fig. 28 に示す．また，平均終了世代数に関しては Table 7 に示す．尚，先ほどの場合と同様，これらのパレート解のプロットは，10 試行平均のうち最も被覆率が平均被覆率に近かった場合のパレート解のプロット図であり，全ての数値計算結果は 10 試行の平均を取ったものである．

Table 6 The Max-Min values (MSLC)

population number	problem	function	method	max value	min value	difference
400 populations	100 item	f_1	tournament	4190	3413	777
			roulette	4169	3571	598
		f_2	tournament	3945	3293	652
			roulette	3901	3350	551
	250 item	f_1	tournament	9229	8312	917
			roulette	9174	8426	748
		f_2	tournament	9878	8764	1114
			roulette	9840	9005	835
	750 item	f_1	tournament	28078	26677	1401
			roulette	27840	26833	1007
		f_2	tournament	28123	26659	1464
			roulette	27933	26909	1024
4000 populations	100 item	f_1	tournament	4235	3336	899
			roulette	4260	3366	894
		f_2	tournament	3978	3193	785
			roulette	4021	3251	770
	250 item	f_1	tournament	9454	8122	1332
			roulette	9404	8272	1132
		f_2	tournament	10083	8553	1530
			roulette	10005	8587	1418
	750 item	f_1	tournament	28954	26149	2805
			roulette	28722	26441	2281
		f_2	tournament	29017	26114	2902
			roulette	28850	26603	2248

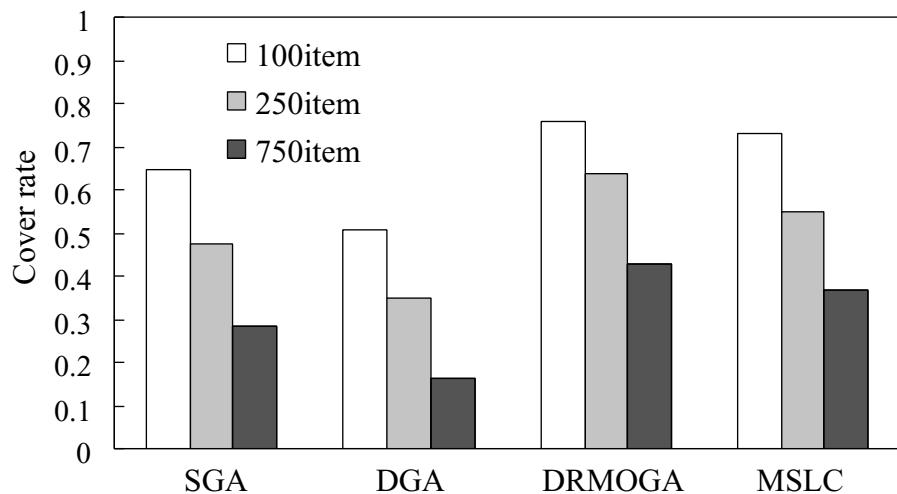


Fig. 9 Cover rate (population size=400)

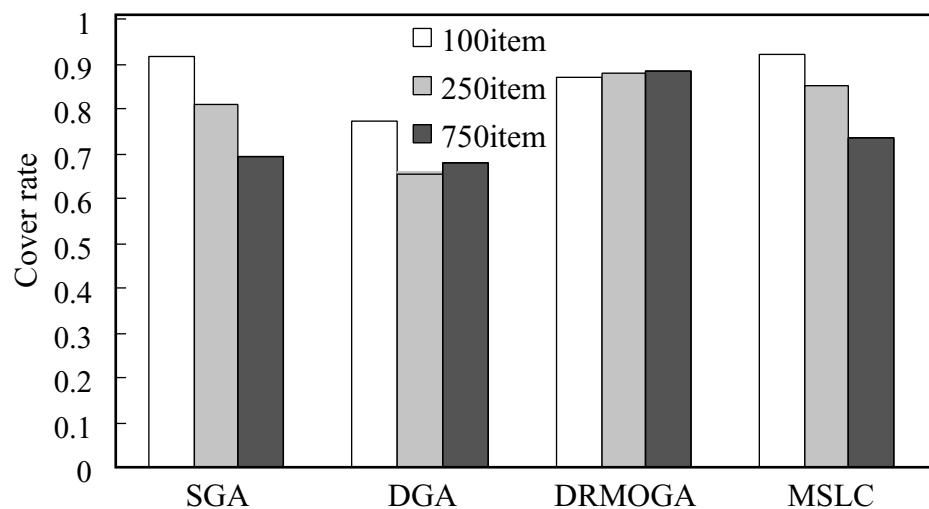


Fig. 10 Cover rate (population size=4000)

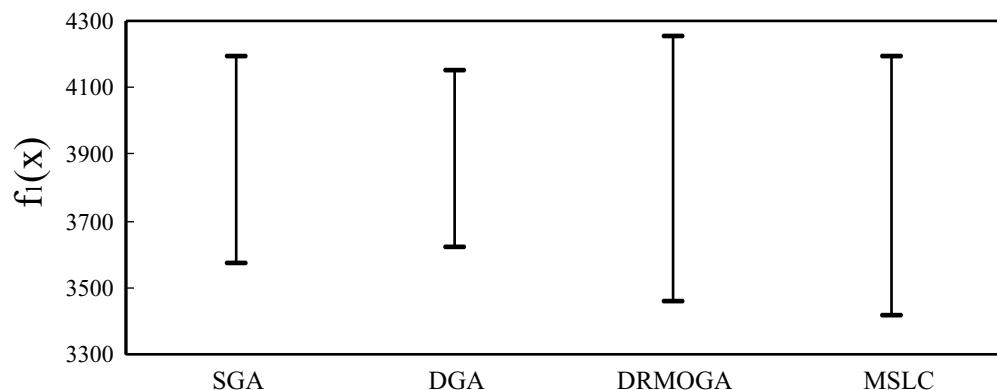


Fig. 11 The Max-Min values of f_1 (100 items problem, population size=400)

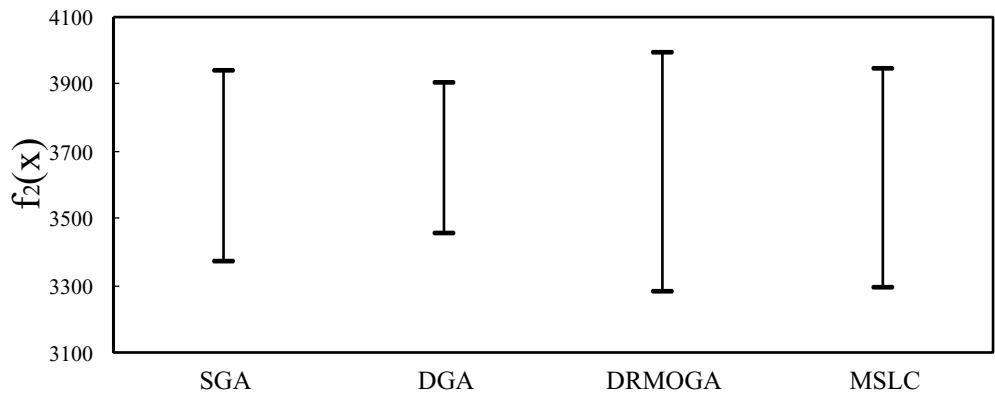


Fig. 12 The Max-Min values of f_2 (100 items problem, population size=400)

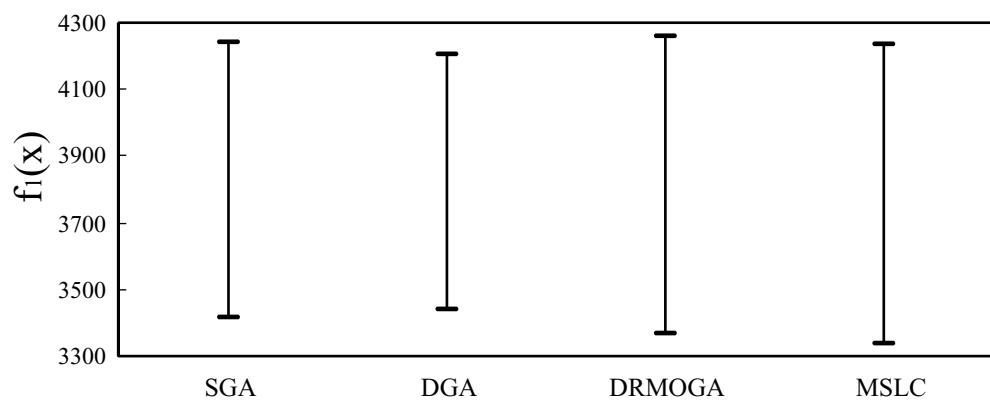


Fig. 13 The Max-Min values of f_1 (100 items problem, population size=4000)

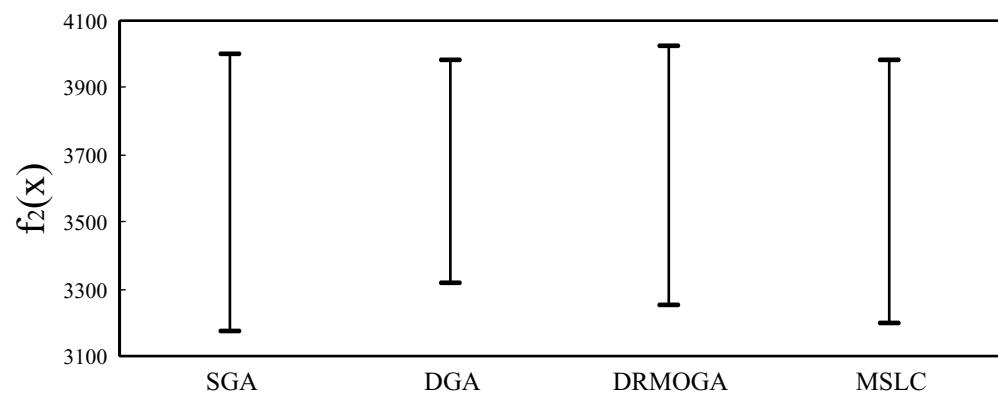


Fig. 14 The Max-Min values of f_2 (100 items problem, population size=4000)

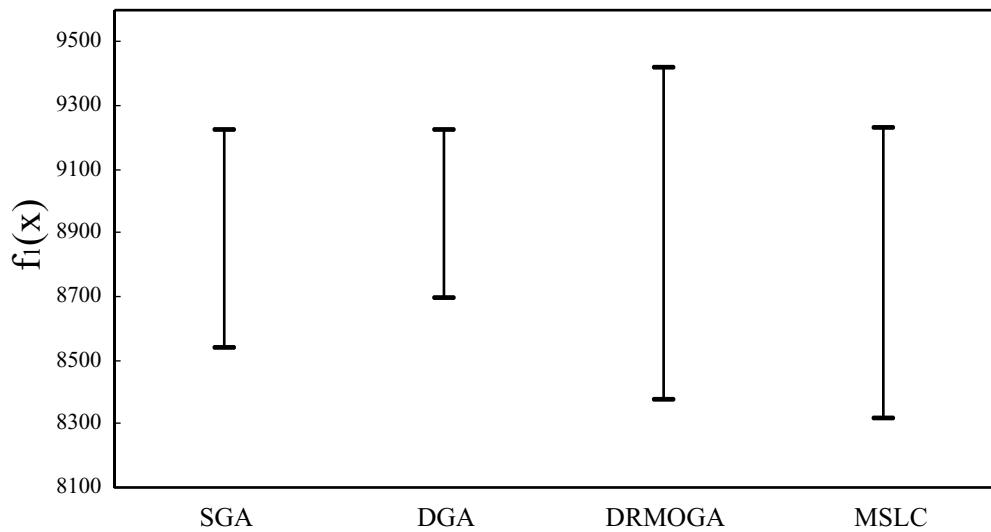


Fig. 15 The Max-Min values of f_1 (250 items problem, population size=400)

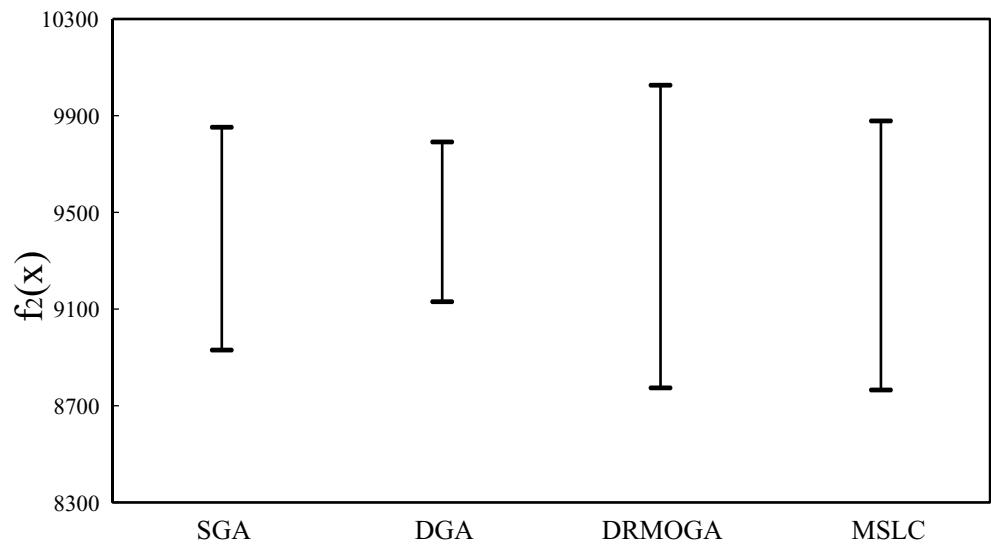


Fig. 16 The Max-Min values of f_2 (250 items problem, population size=400)

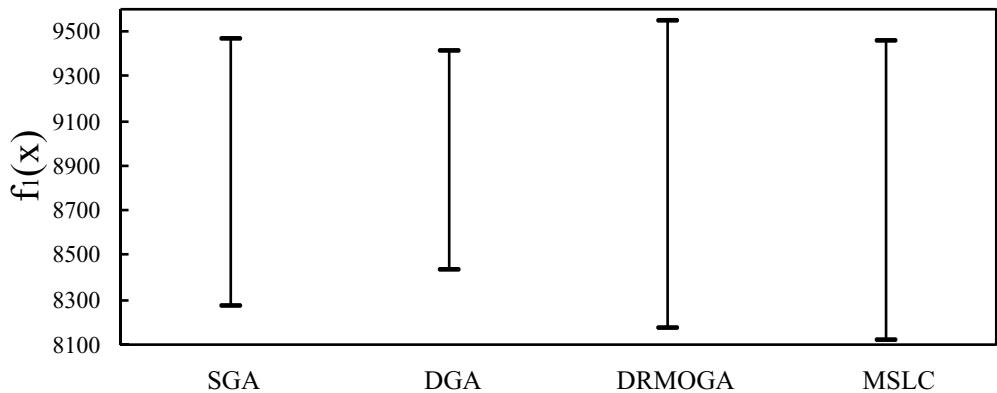


Fig. 17 The Max-Min values of f_1 (250 items problem, population size=4000)

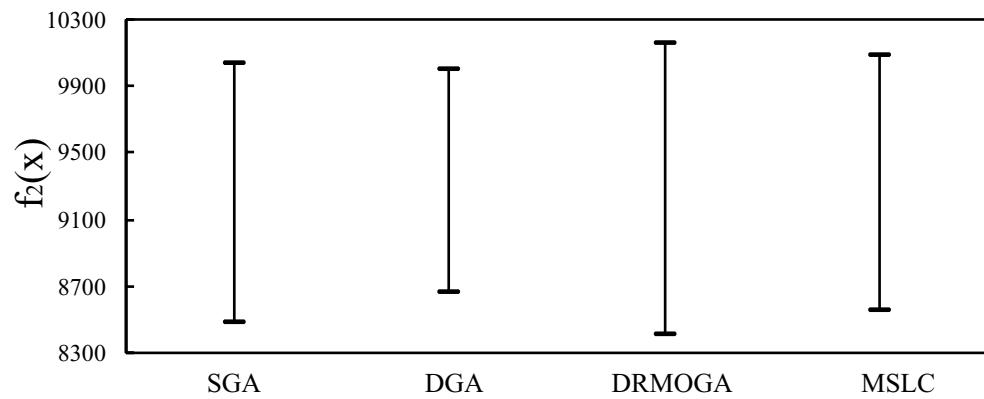


Fig. 18 The Max-Min values of f_2 (250 items problem, population size=4000)

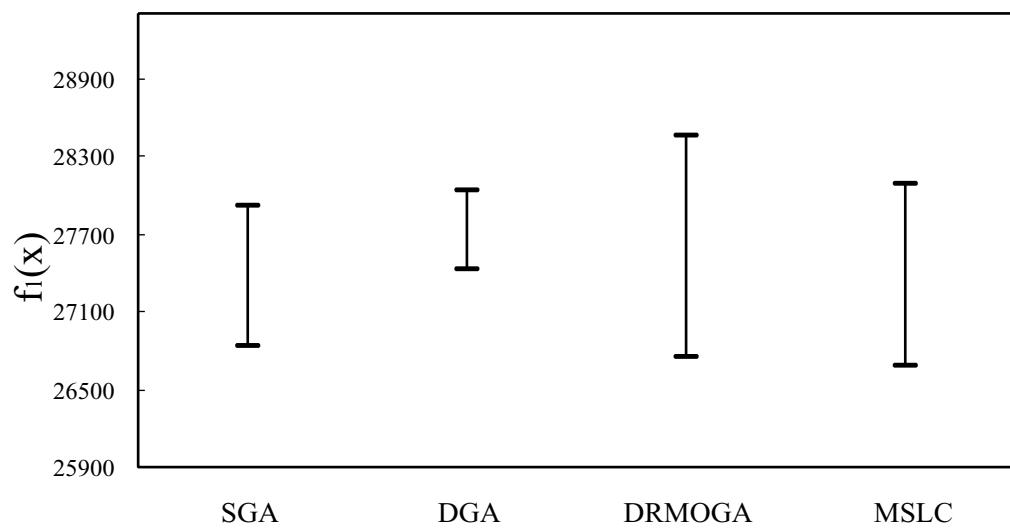


Fig. 19 The Max-Min values of f_1 (750 items problem, population size=400)

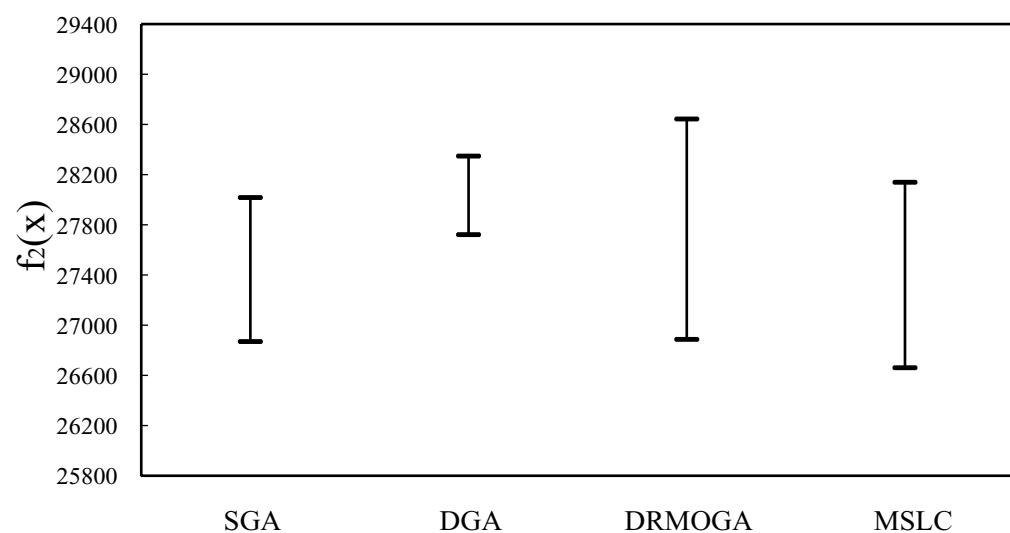


Fig. 20 The Max-Min values of f_2 (750 items problem, population size=400)

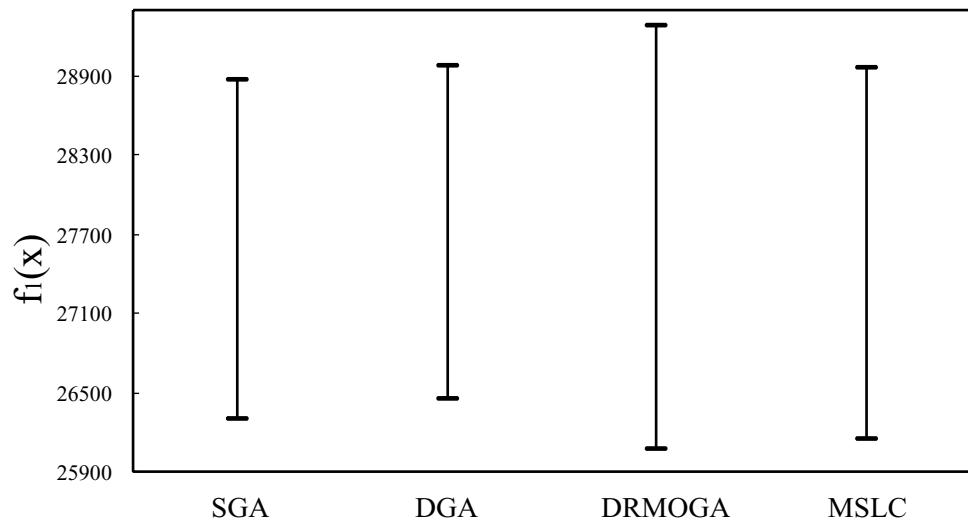


Fig. 21 The Max-Min values of f_1 (750 items problem, population size=4000)

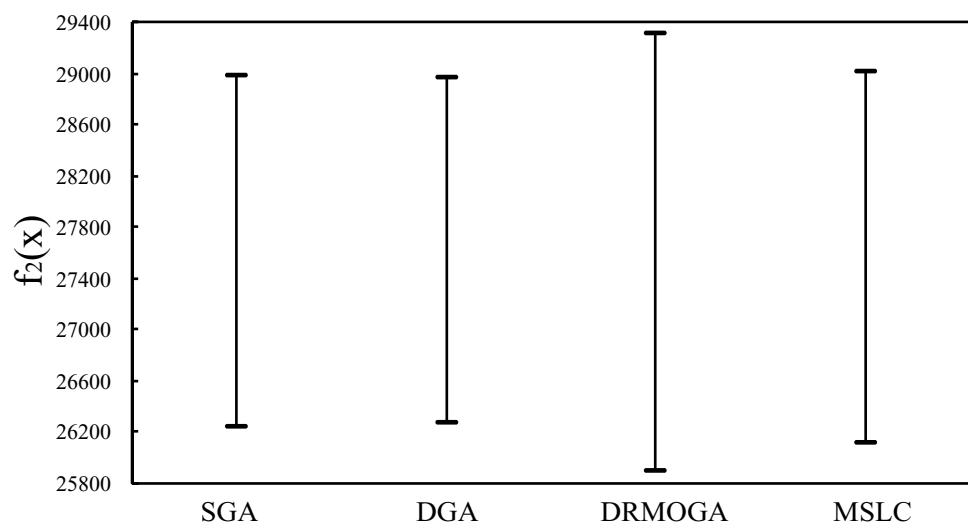


Fig. 22 The Max-Min values of f_2 width of 750 items problem (population size=4000)

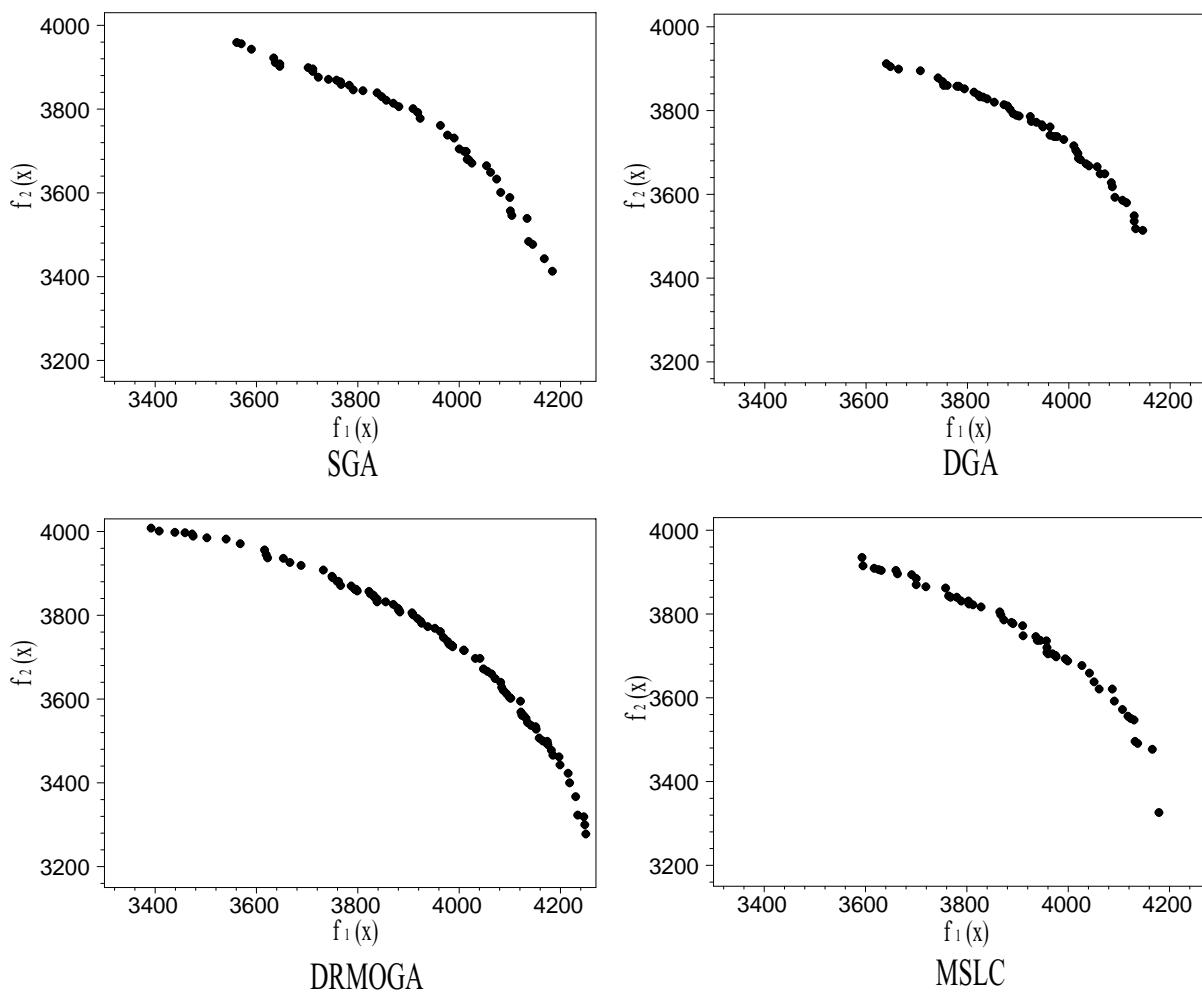


Fig. 23 Pareto optimum individuals (100items problem,population size=400)

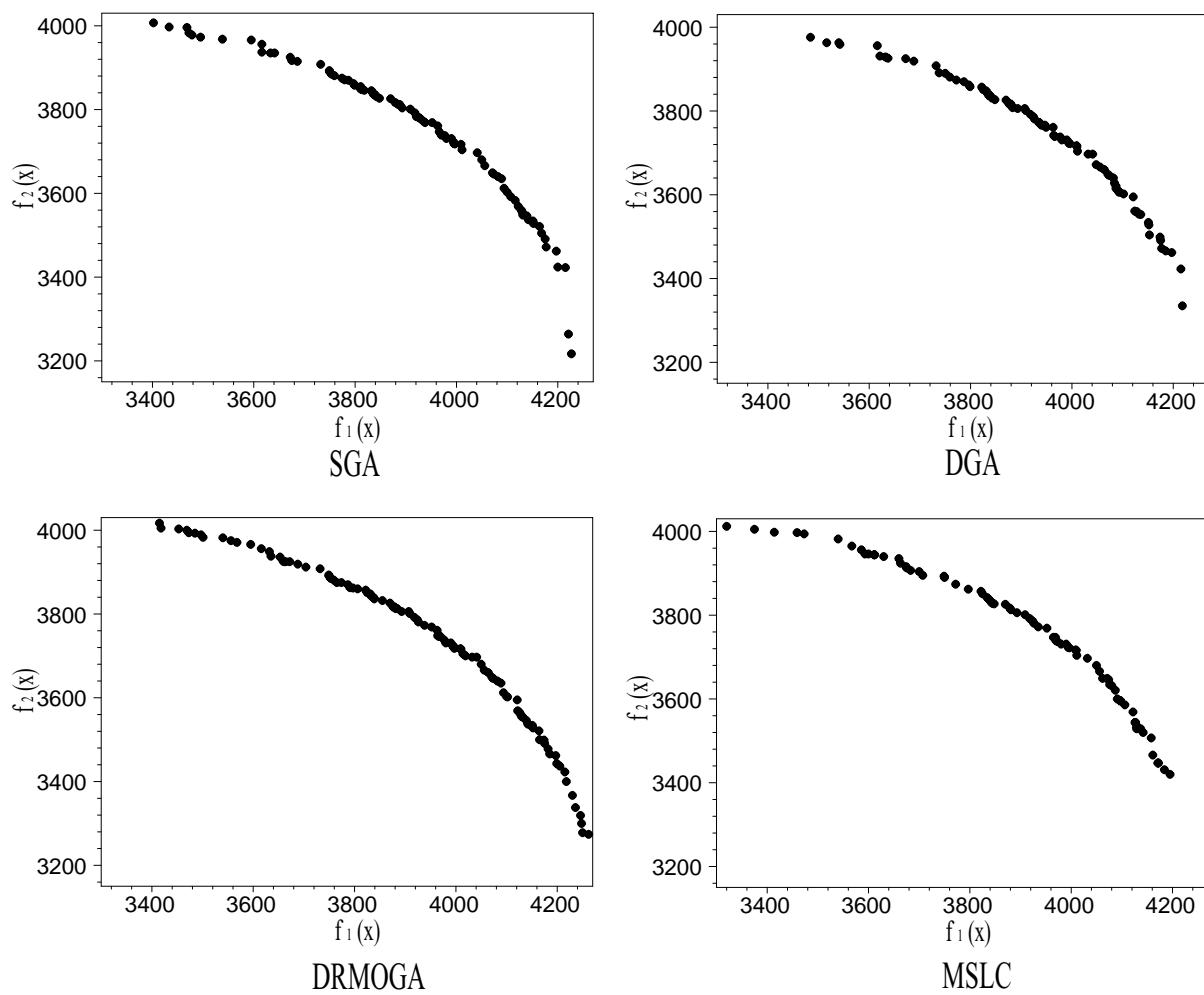


Fig. 24 Pareto optimum individuals (100items problem, population size=4000)

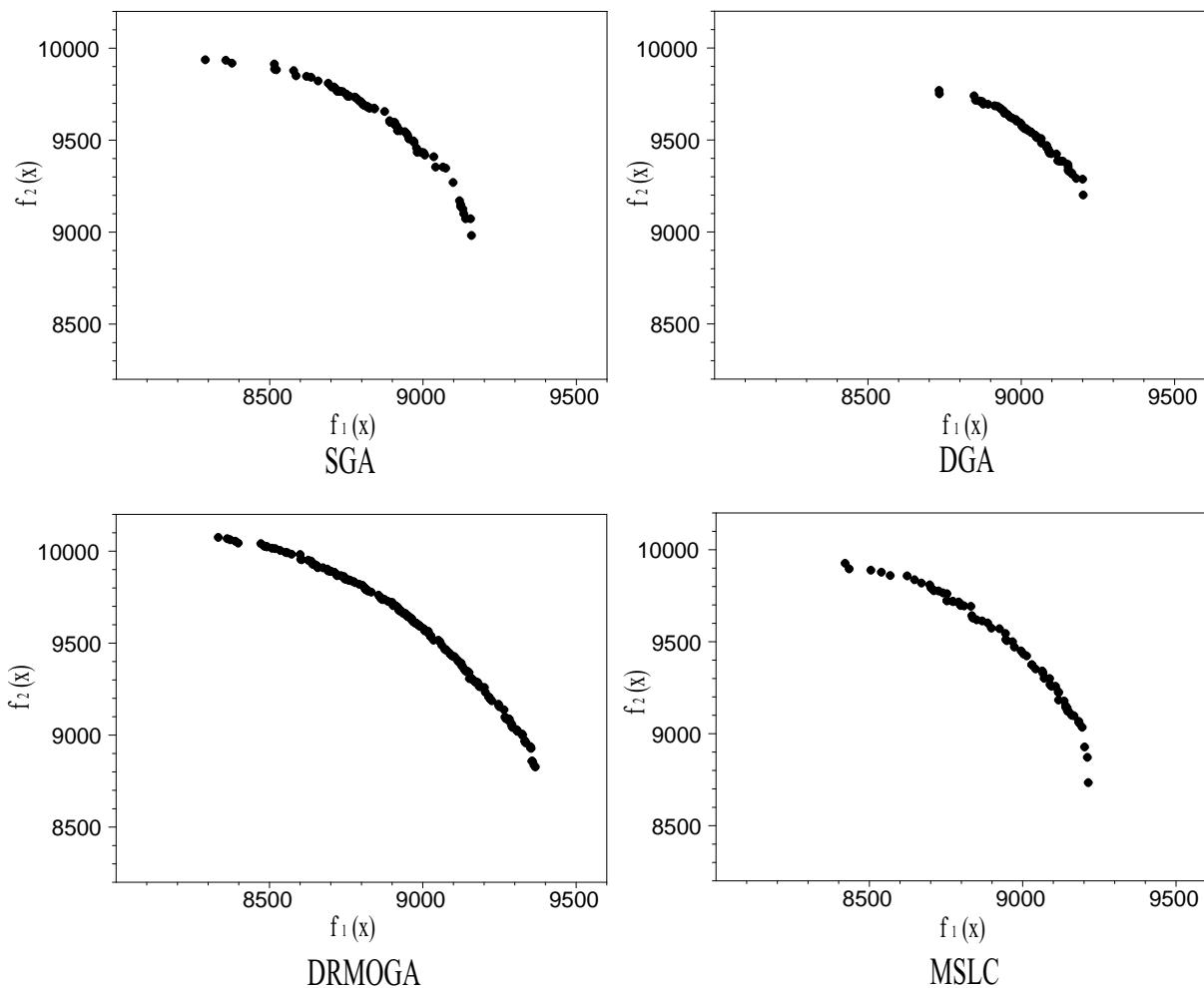


Fig. 25 Pareto optimum individuals (250items problem, population size=400)

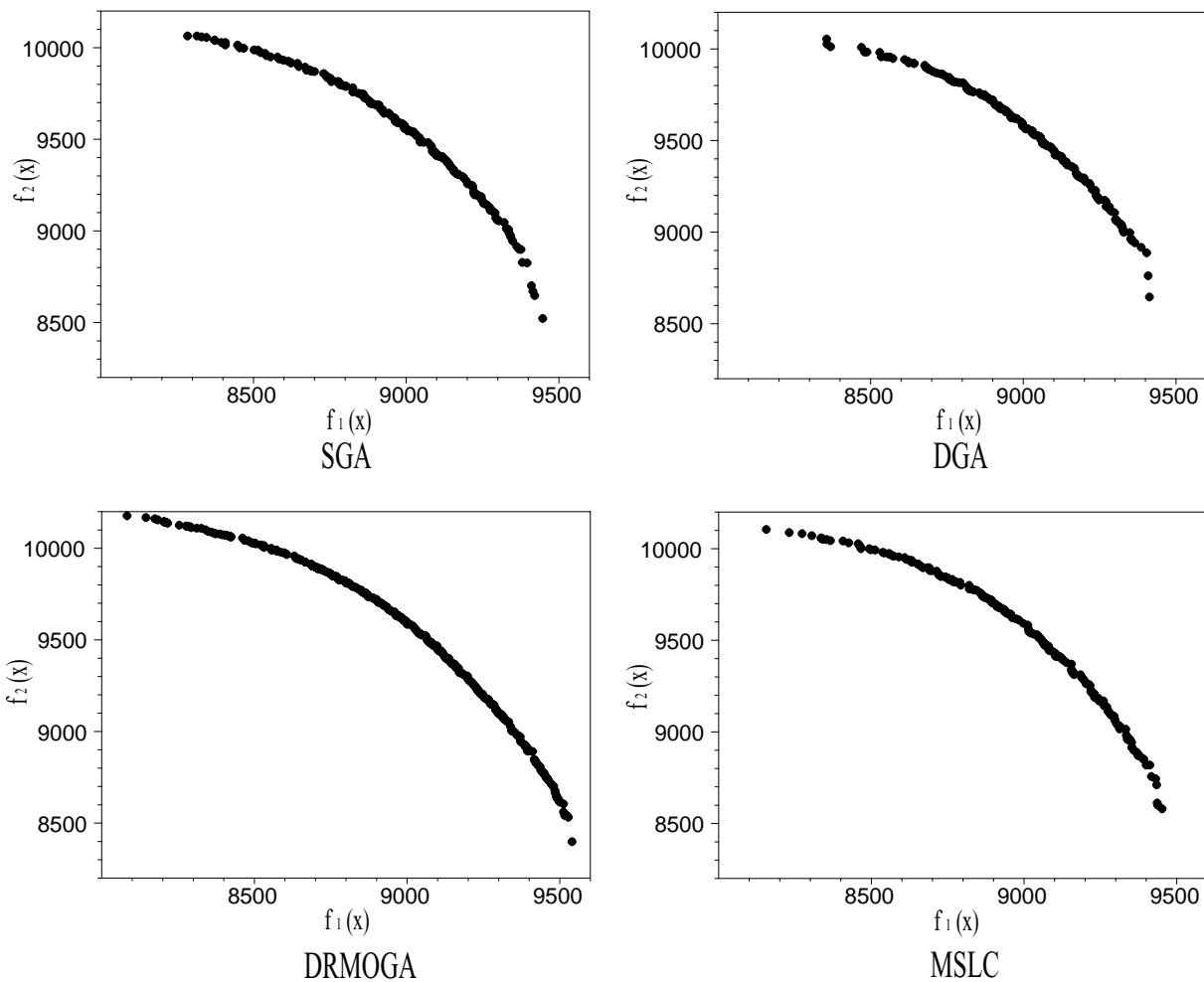


Fig. 26 Pareto optimum individuals (250items problem, population size=4000)

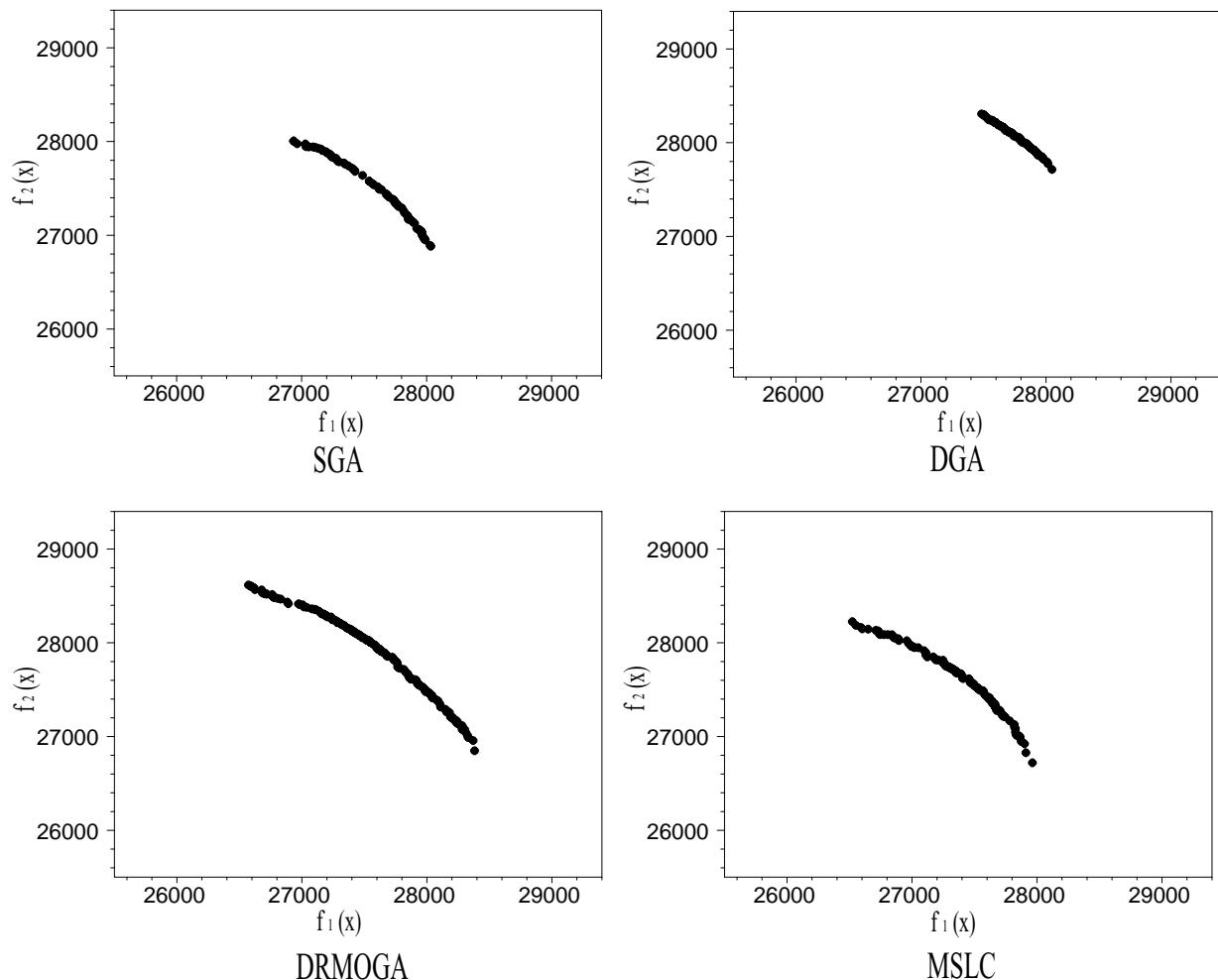


Fig. 27 Pareto optimum individuals (750items problems, population size=400)

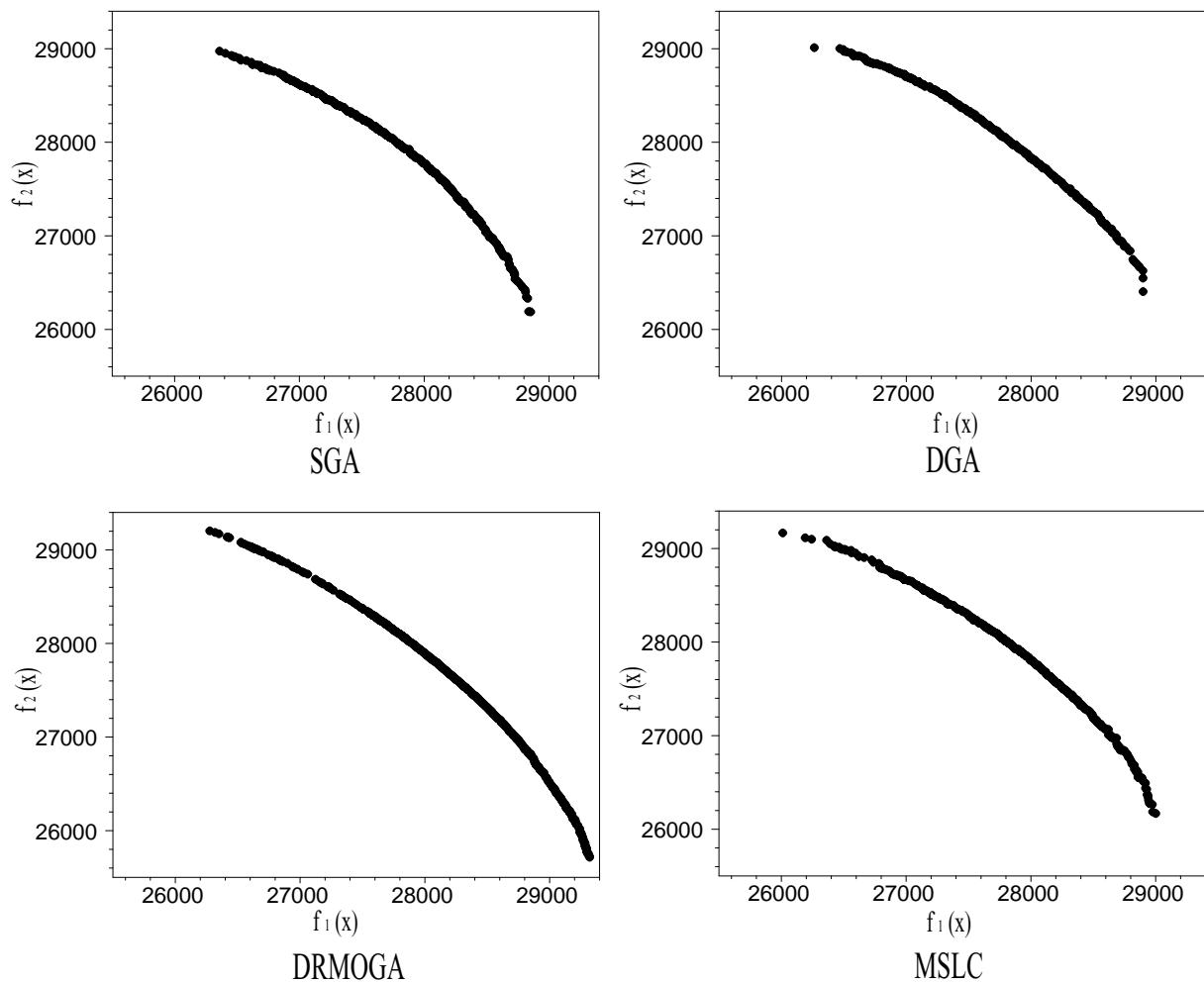


Fig. 28 Pareto optimum individuals (750items problem, population size=4000)

Table 7 Comparison of necessary generation with respect to the problems and methods

population number	problem	method	generations
400	100 itmes	SGA	135.0
		DGA	226.2
		DRMOGA	10002.0
		MSLD	174.3
	250 itmes	SGA	285.2
		DGA	305.9
		DRMOGA	5725.8
		MSLD	355.1
	750 itmes	SGA	990.0
		DGA	2317.1
		DRMOGA	3783.6
		MSLD	1258.9
4000	100 itmes	SGA	116.9
		DGA	172.5
		DRMOGA	10002.0
		MSLD	143.6
	250 itmes	SGA	252.9
		DGA	250.9
		DRMOGA	10002.0
		MSLD	368.3
	750 itmes	SGA	802.9
		DGA	787.6
		DRMOGA	8216.0
		MSLD	1129.4

5.5.1 SGA と DGA の比較

3 章でも述べていたように，DGA は多目的最適化問題の特性を考慮していないため，SGA と比較してもあまり良好な結果が得られないと思われる。そこで，本節では SGA と DGA の数値結果に注目し，両者の解の較を行う。

まず，解の幅広さについて考察する。被覆率について，Fig. 9 および Fig. 10 から分かるように全ての場合において SGA は DGA より良好な結果を示しているのが分かる。特に，個体数が少ない場合にはより両者の差が顕著に現れているのが分かる。ただし，個体数が多く，問題が難しい場合，すなわち 4000 個体，750 荷物のような場合には両者の差はほとんど見られない。その傾向は，Fig. 11 ~ Fig. 22 に示した各目的関数ごとの最大値，最小値の分布位置からも読みとることができる。

次に，パレート解の進行具合から結果を考察する。実際に得られたパレートフロントの図，Fig. 23 ~ Fig. 28 から，個体数が少ない場合には DGA は SGA に比べてより真のパレート解に近い位置に分布しているのが分かる。これは特に，問題の難しい 250 荷物 (Fig. 25)，750 荷物 (Fig. 27) の場合に強くその傾向が現れている。しかし，個体数が増加した場合には SGA も真のパレート最適解付近に分布しており，パレート解の進行具合に関して両手法に大きな差はみられない (Fig. 26, Fig. 28 参照)。

以上の結果より次のことが言える。

- SGA に比べ DGA は多様な解が得られにくい。
- 個体数が少ない場合には特に DGA は幅広い解が得られにくい。
- 個体数が少ない場合，DGA は SGA に比べて精度が優れている。
- 個体数が多い場合，DGA と SGA に大きな差は生じにくい。

DGA では各島ごと独立に探索を行っているため 1 島辺りの個体数が SGA に比べて少ない。そのため，個体数が少ない場合にはその影響が出やすくなり，SGA に比べて多様な解を得ることができない。反面，SGA は個体数が少ない場合，局所的に固まりやすい傾向があるため，解の精度という面では SGA と比較して良好な結果を示していると思われる。

しかし，個体数が増加すると SGA も真のパレート解付近の探索を行い，DGA もより広範囲の探索を各島協調しながら探索を行うため，個体数が少ない場合に比べ両者の差はあまり見られなかった。しかし，個体数が増加した場合でも全ての問題において SGA は DGA とほぼ同精度の解をより幅広く求めていることから SGA は DGA よりも有効な手法であるといえる。

5.5.2 提案モデルと従来の並列モデルの比較

4 章にて提案した，DRMOGA と MSLC の有効性について SGA，DGA との比較を通じてその検証を行う。

まず，解の幅広さについて考察する。Fig. 9 および Fig. 10 から分かるように DRMOGA は，全ての場合において最も高い被覆率を得ているのが分かる。特に，個体数が 4000 個体の場合 (Fig. 10) においては，他のアルゴリズムが問題の荷物数の増加に伴って被覆率の値を下げているのに対して，DRMOGA のみが平均的に非常に良好な値を保っているのが分かる。対して MSLC は，DRMOGA ほどではないものの，他の手法と比べ良好な値を示している。一方，各目的関数ごとの最大値，最小値の分布位置を示した Fig. 11 ~ Fig. 22 を見た場合にも，DRMOGA，MSLC ともに良好な値を示し

ている。100 荷物の場合には、問題が簡単であるためどのアルゴリズムにも大きな差は見られないが、250 荷物、750 荷物の場合には DRMOGA と MSLC の優位性がはっきりと現れている。

次に実際のパレート解のプロット図から結果の考察を行う。その他の結果と同様、100 荷物の個体数が多い場合にはどの手法も真のパレート解付近に幅広く解が得られていてどの手法にも差が見られない (Fig. 23)。しかし、その他の場合には、DRMOGA が他の手法に比べ非常に幅広く解が得られている様子が分かる。特に、個体数が多い場合には、非常に幅広くかつ隙間無く真のパレート解が得られている (Fig. 26, Fig. 28)。MSLC も、DRMOGA ほどでは無いものの SGA, DGA と比較して良好な結果を示している。しかし、個体数が多く、問題が比較的難しい場合には、SGA, DGA, MSLC には大きな違いが見られなかった。

また精度面 (解への近さ) から見た場合、個体数が少ない場合には、DGA が最も真のパレート解へ近い解分布をしているのが分かる (Fig. 25, Fig. 27)。逆に、個体数が少ない場合には SGA は精度面では他の手法と比べて若干悪い結果を示している。しかし、個体数が 4000 個体の場合には、各手法ごとの精度面での差はほとんど無くなり、どの手法も精度面においては良好な結果を示している。

以上の結果より次のことが言える。

- ・総合的に見た場合、DRMOGA が最も優れた手法である。
- ・MSLC は DRMOGA には劣るものの全ての問題において SGA, DGA より良好な解を示した。
- ・個体数が 10 倍になることにより全ての手法において解の精度が大きく向上した。
- ・MSLC は、DRMOGA に比べ多様性の面で劣るが SGA, DGA と比較して良好な結果が得られた。

DRMOGA は、多様性に関して対象問題、用いる個体数に関わらず最も良好な結果が得られた。これは、DRMOGA の持つ目的関数値を基準に個体を分割する効果が非常に効果的に探索へ影響したものと考えられる。すなわち、任意の目的関数値を基準に個体分割を行うことにより、

- ・個体集団は、全体として幅広く保存される。
- ・近傍個体同士での探索となるため、効果的な交叉が行われる。
- ・従来の DGA と違い、全体的な視野に立った個体の多様性保持機能。

などの効果が探索へ影響したものと思われる。一方、MSLC についても DRMOGA ほどではないが、SGA, DGA と比べて良好な結果が得られた。これは、MSLC における局所的選択の多様性保持機能が上手く機能しているからだと思われる。特に、SGA に比べても明確に優位な結果が得られたのは、SGA におけるシェアリング機能以上に局所的選択における多様性の保持が有効であった証拠である。

5.5.3 分割母集団モデルにおけるプロセス数と解との関係

DGA および DRMOGA のような分割母集団モデルは、基本的に全ての個体を各島 (各プロセス) で等分に分割し探索を行うため、用いる総個体数とプロセス数の関係が解へ大きく影響する。そこで、ここでは DGA と DRMOGA に対して用いるプロセス数と解への影響について考察する。

設定としては、4000 個体を用いて 16 プロセスの場合と 128 プロセスの場合について比較を行った。そのため、16 プロセスを用いた場合には 1 島辺り 250 個体、128 プロセスを用いた場合には 1 島辺り 32 個体となっている。使用したパラメータは、128 プロセスの場合において 128 島で計測を行ったこと以外は、前節の数値実験で用いたものと同じである (Table 5)。

得られた結果の内，被覆率について Fig. 29 に示す．各目的関数ごとの最大値と最小値の平均に関しては，各問題の各目的関数ごと，用いた個体数別に分けて Fig. 30～Fig. 34 に示す．さらに，最終的なパレート解の図を各問題，用いた個体数ごとに Fig. 36～Fig. 38 に示す．また，平均終了世代数に関しては Table 8 に示す．尚，先ほどの場合と同様，全ての数値計算結果は 10 試行の平均を取ったものである．

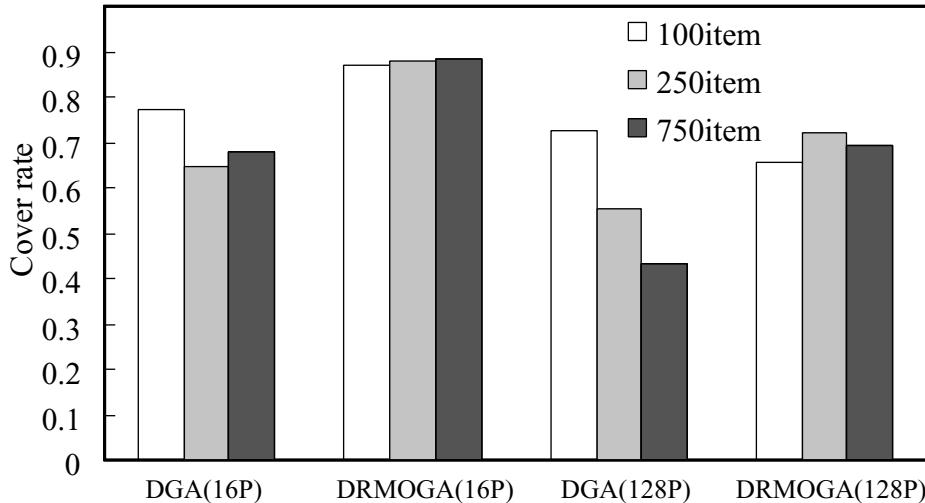


Fig. 29 Cover rate (population size=4000)

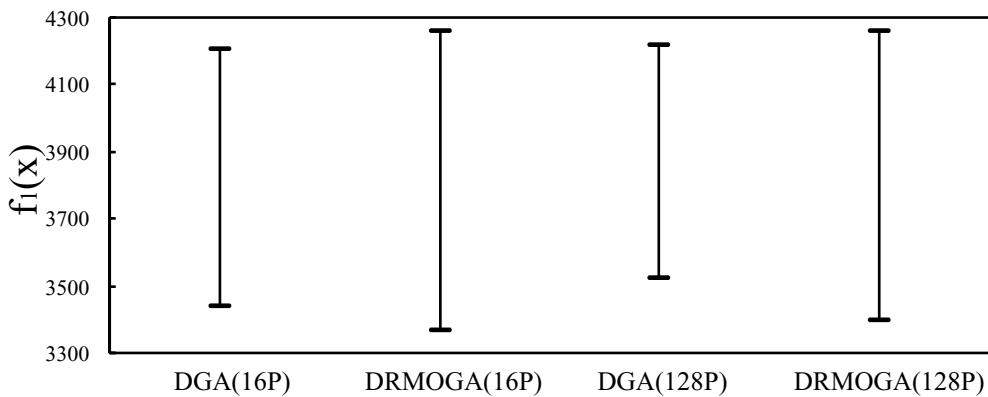


Fig. 30 The Max-Min values of f_1 (100items problem, population size=4000)

幅広さに関しては，Fig. 29 より，DGA，DRMOGA 共に全ての問題に対して 16 プロセスを用いた場合の方が 128 プロセスを用いた場合より良好な結果が得られているのが分かる．この傾向は，Fig. 30～Fig. 34 に示した各目的関数ごとの最大値，最小値の分布位置からも読みとることができる．

また，実際のパレート解のプロット図を比較しても全ての場合において 16 プロセスを用いた場合の方が 128 プロセスを用いた場合よりも幅広く，かつ隙間のない良好な結果が得られていることが分かる (Fig. 36～Fig. 38) ．

以上の結果より次のことが言える．

- ・ 1 島辺りの個体数が少ない場合，あまり良好な結果が得られない．
- ・ 用いるプロセス数によって得られる解が大きく異なる．

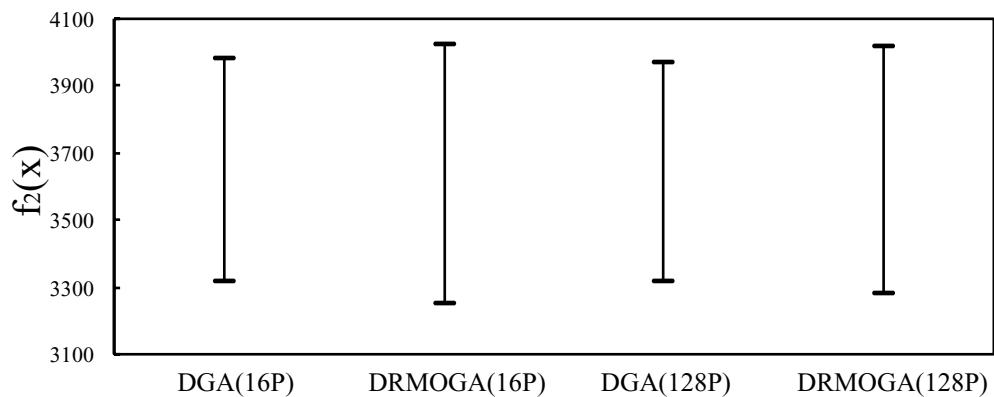


Fig. 31 The Max-Min values of f_2 (100items problem, population size=4000)

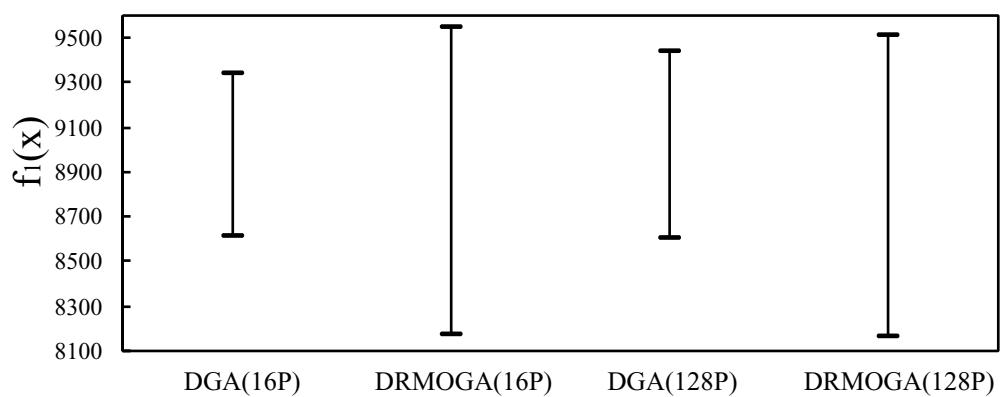


Fig. 32 The Max-Min values of f_1 (250items problem, population size=4000)

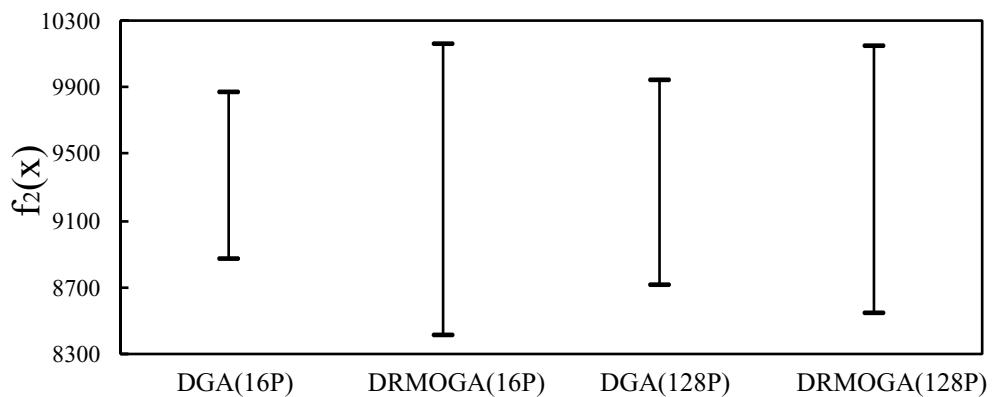


Fig. 33 The Max-Min values of f_2 (250 items problems, population size=4000)

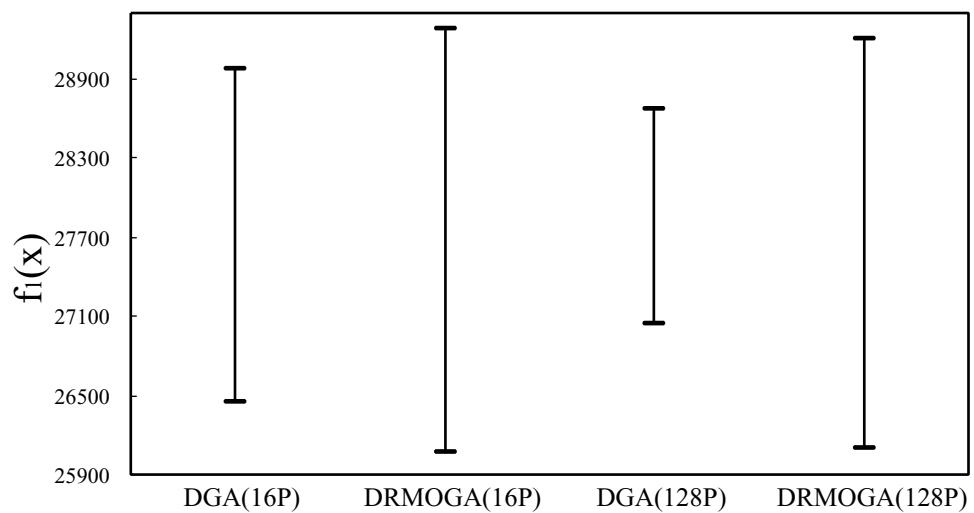


Fig. 34 The Max-Min values of f_1 (750items problems, population size=4000)

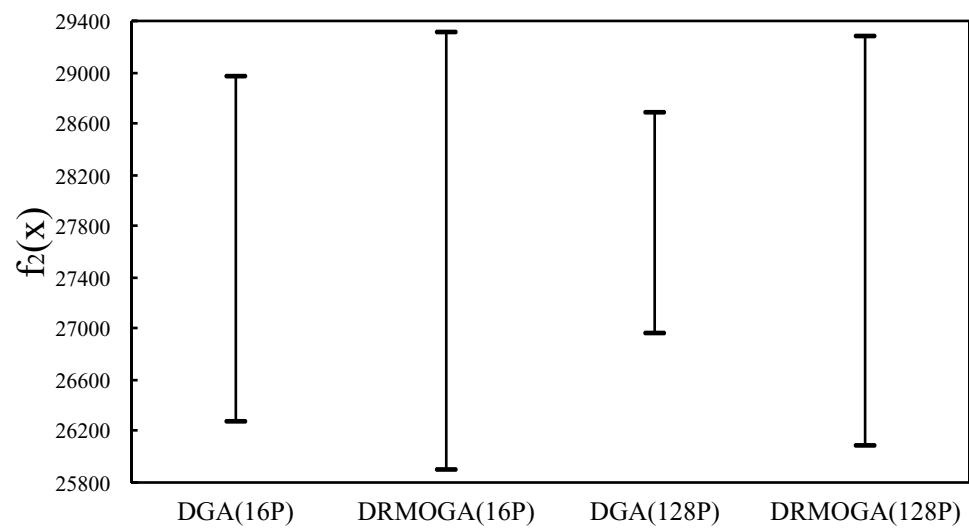


Fig. 35 The Max-Min values of f_2 (750items problem, population size=4000)

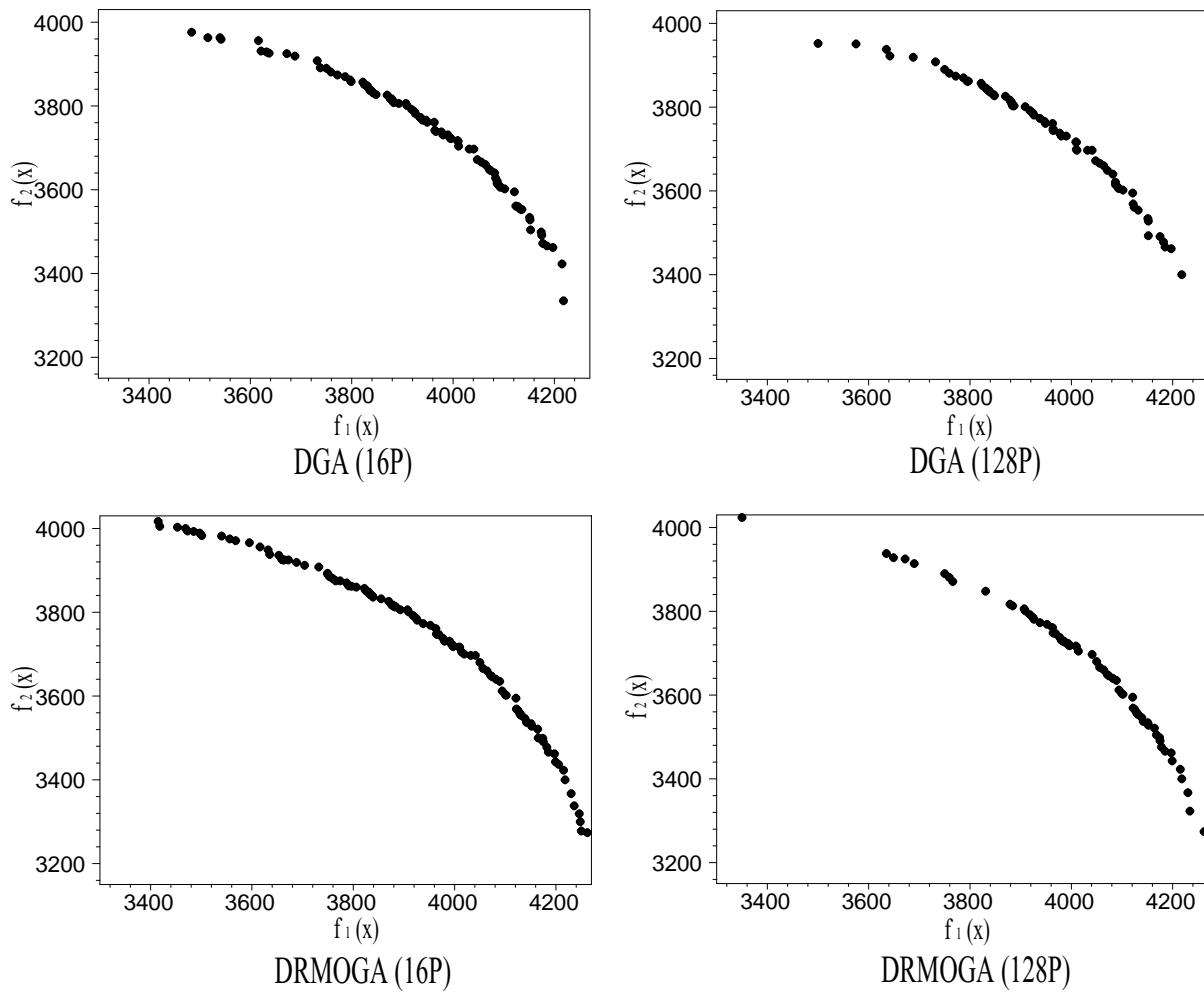


Fig. 36 Pareto optimum individuals (100items problem, population size=4000)

Table 8 Comparison of necessary generation with respect to the problems and methods

problem	method	generations
100 itmes	DGA (16p)	226.2
	DGA (128p)	270.8
	DRMOGA (16p)	10002.0
	DRMOGA (128p)	10002.0
250 itmes	DGA (16p)	250.9
	DGA (128p)	10002.0
	DRMOGA (16p)	10002.0
	DRMOGA (128p)	10002.0
750 itmes	DGA (16p)	787.6
	DGA (128p)	10002.0
	DRMOGA (16p)	8216.0
	DRMOGA (128p)	10002.0

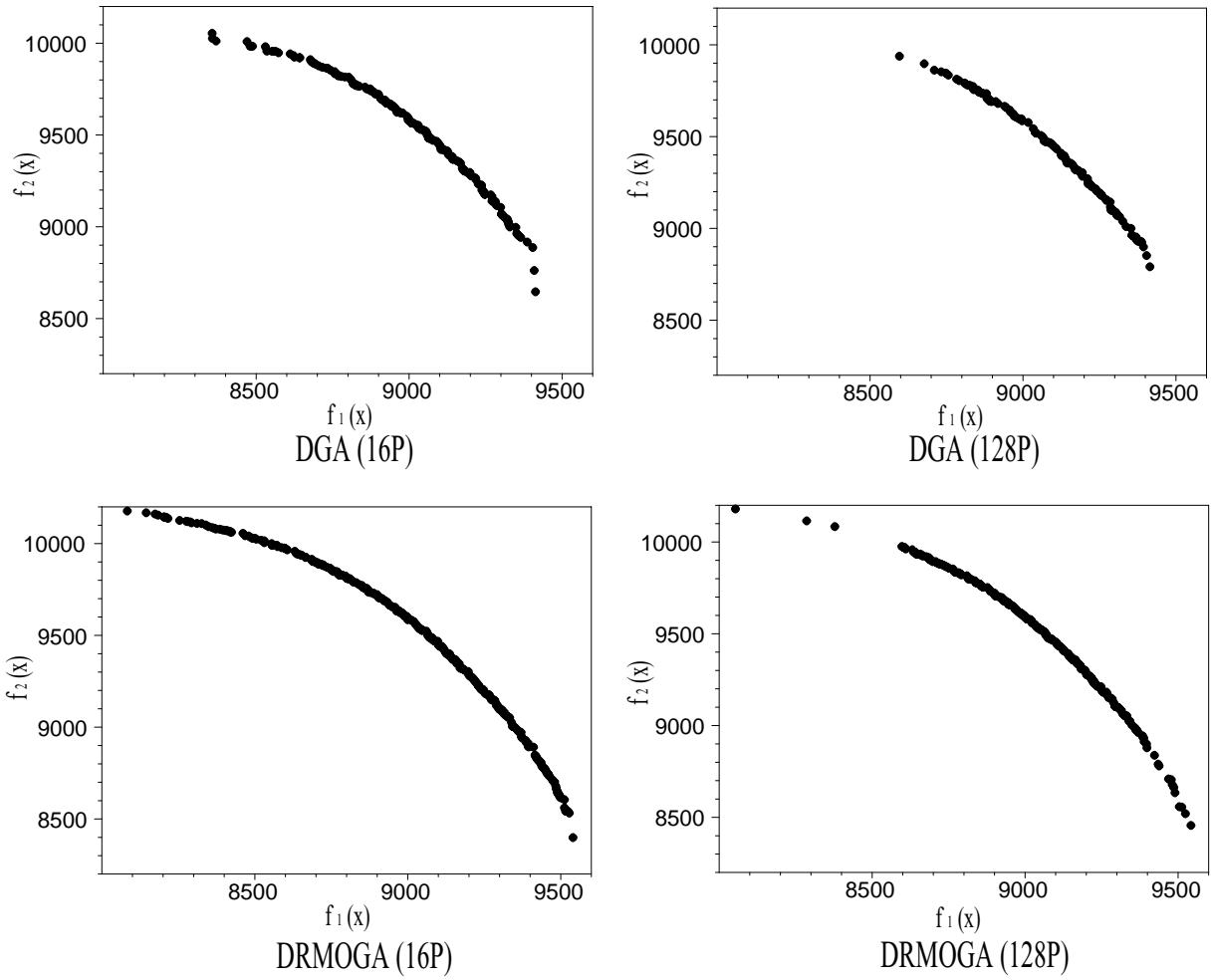


Fig. 37 Pareto optimum individuals (250items problem, population size=4000)

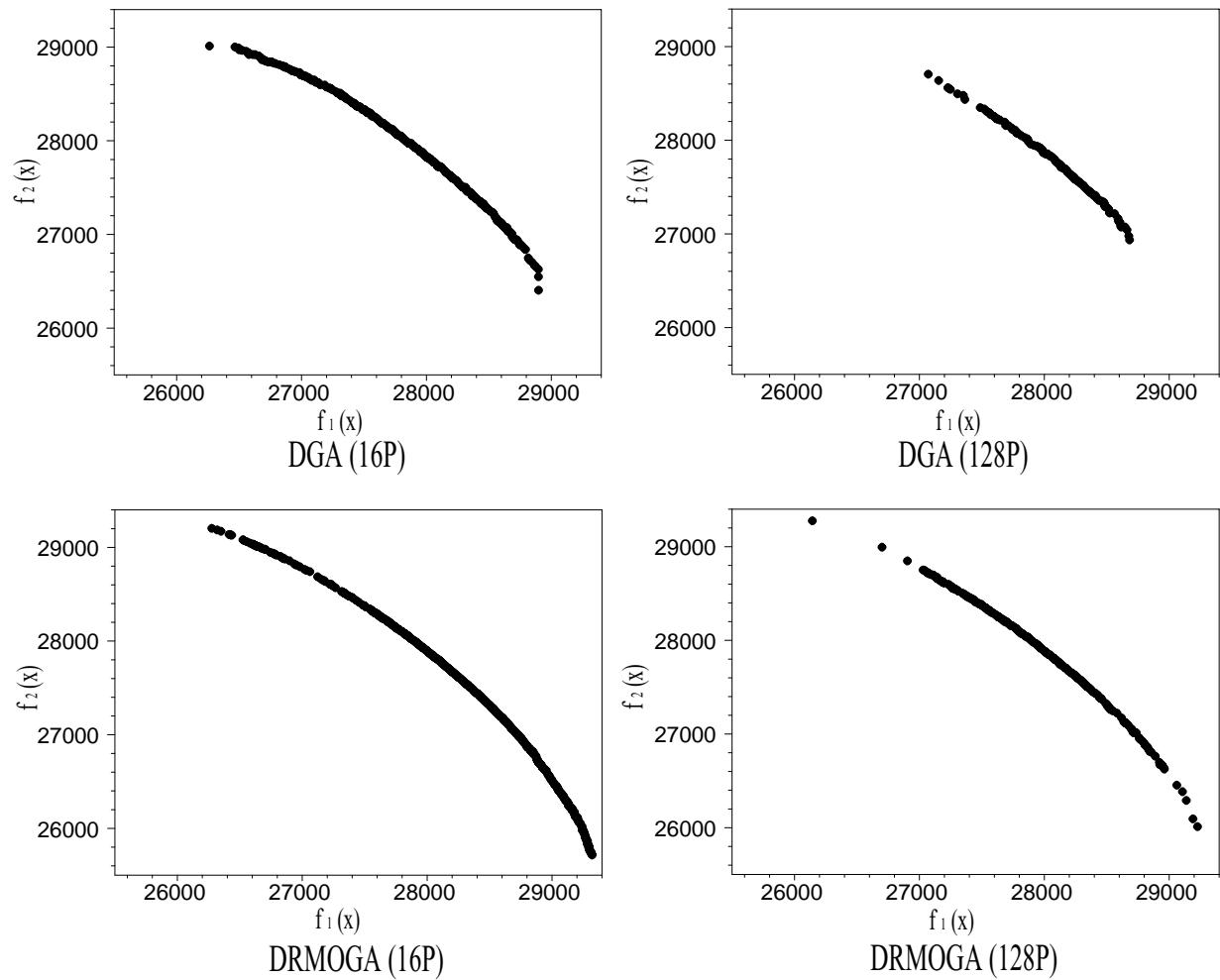


Fig. 38 Pareto optimum individuals (750items problem, population size=4000)

Table 9 Calculation time

method	Number of Processors	Times(sec)
SGA	1	16657.2
DGA	1	16657.2
	2	4011.3
	4	1318.3
	8	477.3
	16	213.4
DRMOGA	1	16657.2
	2	4271.1
	4	1500.3
	8	583.2
	16	264.3
MSLC	1	4042.4
	2	5160.8
	4	2371.0
	8	1675.9
	16	1685.1

用いるプロセス数によって解が異なる最も大きな原因是、1島辺りの個体数の変化であると思われる。今回の結果より、分割母集団モデルを用いる場合には、1島辺りの個体数がある程度存在しない場合には良好な結果が得られないことが分かった。

マスタースレーブ型モデルでは、用いるプロセス数による解への影響は一切ないためこの点において MSLC は、DGA や DRMOGA のような母集団分割モデルと比較して優れていると言える。

5.5.4 並列化効率について

本節では、各並列アルゴリズムの並列化効率について考察する。各モデルの総評価回数を一定とするため、4000 個体を 1000 世代まで計算した場合の計算時間について計測した。使用したパラメータは、前節の数値実験で用いたものと同じである (Table 5)。

得られた結果を Table 9 に示す。また、DGA における並列化効率を示した図を Fig. 39, DRMOGA における同様の図を Fig. 40 に、MSLC における図を Fig. 41 に示す。

Table 5 より 1 プロセスを用いる場合には、SGA にくらべ MSLC の方が約 4 倍ほど早く終了しているのが分かる。これは、GA オペレータにおける選択にかかる時間の差が結びついたものと思われる。多目的 GA では、選択の際に必要となる個体の適合度を決定するため全ての個体を用いて個体間比較を行っている。SGA においてこの個体間比較は個体数の階乗回行う必要があるため、個体数が膨大である場合には個体比較にかかる時間が大きくなる。一方、MSLC では個体比較は全世代におけるランク 1 の個体と現世代における親 2 個体、子 2 個体の 4 個体を加えた数で行われるため SGA に

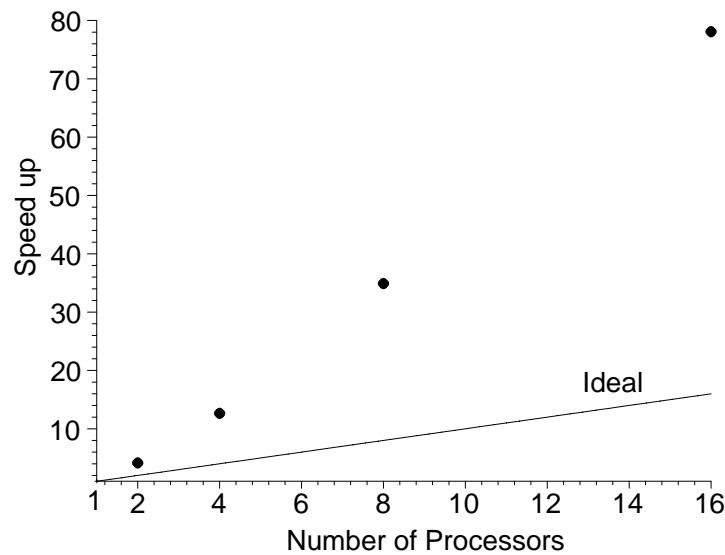


Fig. 39 Speed Up (DGA)

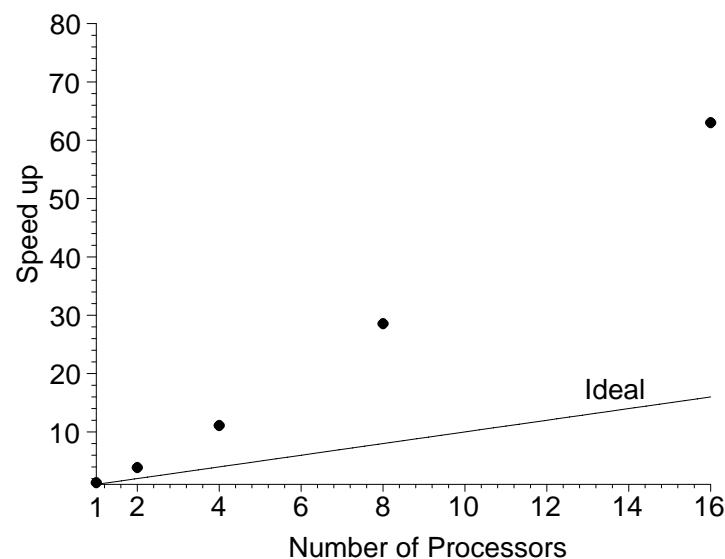


Fig. 40 Speed Up (DRMOGA)

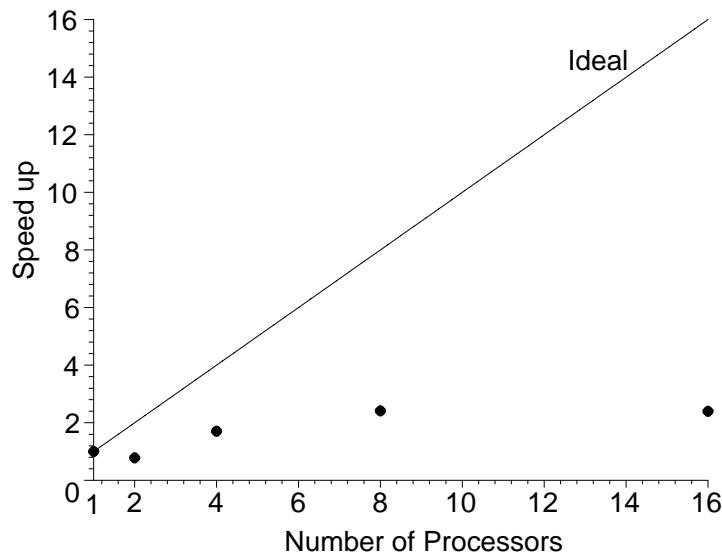


Fig. 41 Speed Up (MSLC)

比べて個体比較にかかる時間が短くなる。

また, Fig. 39 および Fig. 40 より DGA, DRMOGA では線形以上の並列化効率が得られているのが分かる。これは、分散化することにより 1 島辺りの個体数の数が島数分の 1 となり、個体比較にかかる計算時間が短くなっているためである。上記で述べたように個体比較には個体数の階乗回の比較を行う必要があるため、個体数の減少は大幅な計算時間の短縮につながるのである。

一方、MSLC では 8 プロセッサーで約 2 倍と DGA, DRMOGA と比較して大きな速度向上は求められなかった (Fig. 41)。また、8 プロセッサーの場合と 16 プロセッサーの場合では計算時間にあまり変化がないという結果が得られた。これは、DGA や DRMOGA に比べ通信量が多いことが一番の原因だと考えられる。そのため、高速通信を行える環境の場合には、本結果と比べかなり良い並列化効率が得られるものと思われる。

6 結言

本論文では、多目的 GA における新たな並列アルゴリズムとして DRMOGA と MSLC の 2 つの手法を提案し、多目的ナップザック問題を用いた数値実験を通してその有効性の検証を行った。その結果得られた結論を以下に示す。

- (1) 従来の DGA では、多目的最適化問題において良好な解を得ることができなかった。これは DGA には多目的最適化問題の特性を考慮に入れていないため、より広範囲の探索が行えないためである。
- (2) 提案した DRMOGA, MSLC は、従来までの DGA, SGA と比較して良好な結果を得ることができた。このことより、提案した手法は多目的最適化問題に充分有効であることが確認できた。
- (3) 分割母集団モデルである DGA, DRMOGA では、用いるプロセスによって大きく解が異なる。特に、1 島辺りの個体数が充分にない場合には、良好な結果を得ることができない。

(4) MSLCにおいては , アルゴリズムの仕組みとして多様性の保持機能が組み込まれているため , ルーレット選択と比べより厳密な適合度に基づいた選択を行うトーナメント選択の方が非常に良い結果を得ることができた .

(5) DRMOGA や DGA では , 線形以上の速度向上が得ることができる . これは , 1島あたりの個体数の減少が選択における個体比較の計算時間短縮につながるためである . 対して MSLC は分割母集団モデルと比べて , 良好な速度向上を得ることができなかつた . これは , 分割母集団モデルとマスタースレーブ型モデルにおける通信量が大きく異なるためである .

謝辞

本研究を遂行するにあたり，多大なる御指導，そしてご協力を頂きました同志社大学 工学部 知識工学科 三木光範教授，廣安知之助手に心より感謝いたします。

また，本研究の実施にあたりご協力を頂きました同志社大学大学院 工学研究科 知識工学専攻 知的システムデザイン研究室 近藤 健史君に深く感謝いたします。さらに，本論文作成に関して惜しみない協力をしてくれた同志社大学 知識工学科 知的システムデザイン研究室 奥田 環君，岡田 靖男君に感謝します。

そして，知的システムデザイン研究室の研究室の皆さん，特に畠中一幸 さん（現 富士通）を初めとする1期生の方々，同回生である2期生の皆さんには，研究生活全般に関してお世話になり，この3年間を非常に有意義に過ごすことができました。ここに感謝の意を表したいと思います。

最後に，3年間の研究室生活を含む6年間の学生生活を理解と寛容を持って見守り，さらなる進学に關しても承諾してくれました両親に感謝いたします。

参考文献

- 1) E. Cantu-Paz. Topologies, migration rates, and multi-population parallel genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, Vol. 1, pp. 91–98, 1999.
- 2) C. A. Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In *Proceedings of Congress on Evolutionary Computation*, pp. 1–11, 1999.
- 3) C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international conference on genetic algorithms*, pp. 416–423, 1993.
- 4) D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- 5) 北野宏明. 遺伝的アルゴリズム 2. 産業図書, 1995.
- 6) 比屋根. 並列遺伝的アルゴリズムによる多目的最適化問題のパレート最適解集合の生成法と定量的評価法. 第 9 回自律分散システムシンポジウム, pp. 295–300, 1997.
- 7) B.R. Jones, W.A. Crossley, and A.S. Lyrintzi. Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 1–11, 1998.
- 8) Trelles-Salazar O. & Sndoval F. Marin, F.J. Genetic algorithms on lan-message passign architectures using pvm. In *Application to the routing problem. In,Davidor, Y.,Schwefel, H.-P.,anner,R(Eds.) , Parallel Problem Solving from Nature, PPSN III, Berlin:Springer-Verlag.*, pp. 534–543, 1994.
- 9) Y.Takai M.Munemoto and Y.Sato. An efficient migration scheme for subpopulation-based asynchronously parallel genetic algorithms. In *in Proceedings of the Fifth International Conference on Genetic Algorithms*, p. 649, 1993.
- 10) T. Murata and H. Ishibuchi. Moga: Multi-objective genetic algorithms. In *Proceedings of the 2nd IEEE International Conference on Evolutionary Computing*, pp. 289–294, 1995.
- 11) L. Nang and K. Matsuo. A survey on the parallel genetic algorithms. *J.SICE*, Vol. 33, No. 6, pp. 500–509, 1994.
- 12) Yamamura-M. Satoh, H. and S. Kobayashi. Minimal generation gap model for gas considering both exploration and expolation. In *Proceedings of the 4th International Conference on Juzzy Logic, Neural Nets and Soft Computing*, pp. 734–744, 1997.

- 13) H. Sawai and S. Adachi. Parallel distributed processing of a parameter-free ga by using hierarchical migration methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, Vol. 1, pp. 579–586, 1999.
- 14) J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of 1st International Conference on Genetic Algorithms and Their Applications*, pp. 93–100, 1985.
- 15) D.Q. Vicini. Sub-population policies for a parallel multiobjective genetic algorithm with applications to wing design. In *Proceedings of International Conference on Systems, Man, and Cybernetics*, pp. 3142–3147, 1998.
- 16) E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. *Parallel Problem Solving from Nature - PPSN-V*, pp. 292–301, 1998.
- 17) E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999.
- 18) 鬼頭宏誌, 森直樹, 松本啓之亮. 近傍モデルを用いたセルラー熱力学的遺伝的アルゴリズムの提案. システム制御情報学会研究発表講演会予稿集, pp. 81–84, 1999.

付 図

1	Pareto optimal solution	4
2	Sharing	6
3	Cover rate	8
1	Global parallelization model	10
2	Coarse-grained parallel model	10
3	Fine-grained parallel model	11
1	DRMOGA (two objectives)	14
2	Concept of MGG model	14
3	Master-Slave model with Local Cultivation model	16
1	Cover rate (comparison of MSLC's selection methods, population size=400)	23
2	Cover rate (comparison of MSLC's selection methods, population size=4000)	23
3	Pareto optimum individuals (100items problem, population size=400)	24
4	Pareto optimum individuals (100items problem, population size=4000)	24
5	Pareto optimum individuals (250items problem, population size=400)	24
6	Pareto optimum individuals (250items problem, population size=4000)	25
7	Pareto optimum individuals (750items problem, population size=400)	25
8	Pareto optimum individuals (750items problem,population size=4000)	25
9	Cover rate (population size=400)	28
10	Cover rate (population size=4000)	28
11	The Max-Min values of f_1 (100 items problem, population size=400)	28
12	The Max-Min values of f_2 (100 items problem, population size=400)	29
13	The Max-Min values of f_1 (100 items problem, population size=4000)	29
14	The Max-Min values of f_2 (100 items problem, population size=4000)	29
15	The Max-Min values of f_1 (250 items problem,population size=400)	30
16	The Max-Min values of f_2 (250 items problem, population size=400)	30
17	The Max-Min values of f_1 (250 items problem, population size=4000)	30
18	The Max-Min values of f_2 (250 items problem, population size=4000)	31
19	The Max-Min values of f_1 (750 items problem, population size=400)	31
20	The Max-Min values of f_2 (750 items problem, population size=400)	31
21	The Max-Min values of f_1 (750 items problem, population size=4000)	32
22	The Max-Min values of f_2 width of 750 items problem (population size=4000)	32
23	Pareto optimum individuals (100items problem,population size=400)	33
24	Pareto optimum individuals (100items problem, population size=4000)	34
25	Pareto optimum individuals (250items problem, population size=400)	35
26	Pareto optimum individuals (250items problem, population size=4000)	36
27	Pareto optimum individuals (750items problems, population size=400)	37

28	Pareto optimum individuals (750items problem, population size=4000)	38
29	Cover rate (population size=4000)	42
30	The Max-Min values of f_1 (100items problem, population size=4000)	42
31	The Max-Min values of f_2 (100items problem, population size=4000)	43
32	The Max-Min values of f_1 (250items problem, population size=4000)	43
33	The Max-Min values of f_2 (250 items problems, population size=4000)	43
34	The Max-Min values of f_1 (750items problems, population size=4000)	44
35	The Max-Min values of f_2 (750items problem, population size=4000)	44
36	Pareto optimum individuals (100items problem, population size=4000)	45
37	Pareto optimum individuals (250items problem, population size=4000)	46
38	Pareto optimum individuals (750items problem, population size=4000)	47
39	Speed Up (DGA)	49
40	Speed Up (DRMOGA)	49
41	Speed Up (MSLC)	50

付 表

1	100item 2 Knapsack problems	18
2	250item 2 Knapsack problems ~ 1 Knapsack($capacity \leq 6536$)~	19
3	250item 2 Knapsack problems ~ 2 Knapsack($capacity \leq 6489$)~	20
4	Cluster system	22
5	Used parameters	22
6	The Max-Min values (MSLC)	27
7	Comparison of necessary generation with respect to the problems and methods	39
8	Comparison of necessary generation with respect to the problems and methods	45
9	Calculation time	48