多目的最適問題における進化的アルゴリズムの並列化

New Models of Parallel Genetic Algorithms in Multi Objective Optimization Problems

渡邉 真也, 廣安 知之, 三木 光範（同志社大学 工学部）
Watanabe Shinya,Tomoyuki Hiroyasu,Mitsunori Miki (Faculty of Engineering，Doshisha Univ)
sin@mikilab.doshisha.ac.jp, tomo@is.doshisha.ac.jp, mmiki@mail.doshisha.ac.jp

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.

1 はじめに

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 [2, 3, 11]．これらの研究は一般に多目的 GA と呼ばれ，Schaffer らの VEGA [11] に始まり，Goldberg のランキング法 [3] や Fonseca らの MOGA [2] などが代表的な研究としてあげられる．

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．

単一目的における GA の並列化に関する研究は近年活発に行われている [8, 10]．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．

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, 12]．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．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，領域分割型多目的遺伝的アルゴリズム (Divided Range Multi-Objective Genetic Alogrithm: DRMOGA) を，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う．

2 GA による多目的最適化への応用

2.1 多目的最適化問題

多目的最適化とは「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

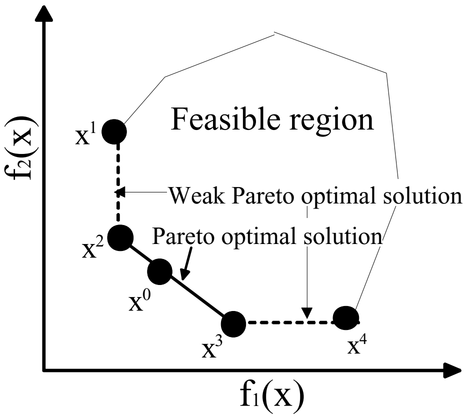

Fig. 1　Pareto optimal solution

問題」と定義される．目的関数が互いに競合し合っているため，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．パレート解の概念図を Fig1 に示す．

2.2　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評価方法である．以下，多目的 GA における個体の評価方法であるランキングついて説明する．

ランキング手法

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．

パレートランキングによる方法とは，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．パレートランキング法は，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

$$r(X_i) = 1 + n_i \tag{1}$$

のように定めることにしている．

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．

3　提案する並列アルゴリズム

多目的最適化 GA の並列化に関する研究は単一目的に比べその数は多くない．また，そこで使用されている計算モデルの多くは単一目的における 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 [12)] 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 [15)]．

しかし，単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．単一目的における GA の場合，基本的に探索の初期段階においてのみ個体の多様性が重要になる．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．

そこで，本研究では多目的 GA の特徴を活かした新たな２つの並列分散アルゴリズムを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，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 Alogrithm: 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．

3.1　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，通常の単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単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，場合によっ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. 2 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．

3.2 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．

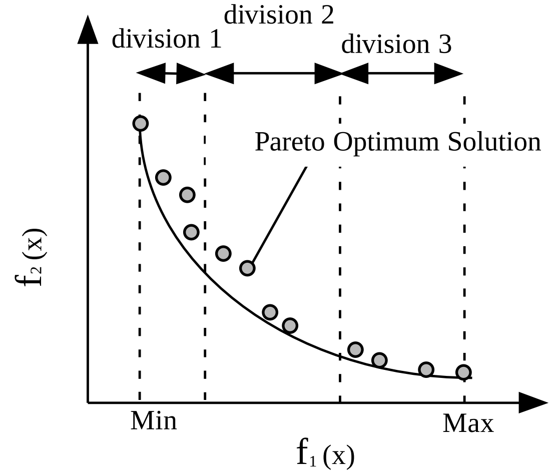

Fig. 2　DRMOGA (two objectives)

局所的培養型マスタースレーブモデルは，佐藤ら [9] により提案された MGG モデル（Minimal Generation Gap model）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．

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．すなわち，各ノードは個体評価を行うために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

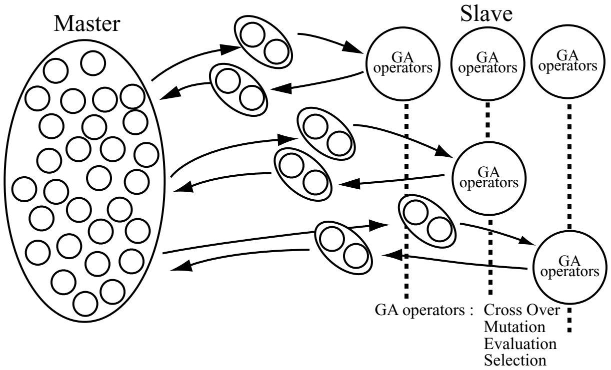

Fig. 3 Master-Slave model with Local Cultivation model

おいて全個体からランク1の個体を摘出し，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．

本提案手法では，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．

4 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．本研究では，主な対象問題として離散的テスト問題として最も代表的な多目的0/1ナップザック問題を取り上げた．

以下では，多目的0/1ナップザック問題および対象問題に対する GA の構成とパラメータについて説明する．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も考察を行った．さらに，より評価計算の負荷が高く，より現実的な携帯電話におけるネットワーク設計問題についても各手法を適用し，得られた結果について考察を行う．

4.1 0/1 多目的ナップサック問題

多目的ナップサ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 [14, 13]．ここでは多目的0/1ナップサック問題の定義，この問題に対する GA の構成と GA パラメータについて概説する．

4.1.1 0/1 ナップサック問題の定義

一般に0/1ナップサック問題は，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，上限制約としてナップサックの容量がある．この問題の目的は，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

この単一目的問題は，ナップサックの数および付随する荷物のセットを複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．ここでの多目的0/1ナップサック問題は，以下の式によって定式化される．

$p_{i,j}$ = ナップサック i に関する荷物 j の利益
$w_{i,j}$ = ナップサック i に関する荷物 j の荷物の重さ
c_i = ナップサック i の許容重量,

設計変数値であるベクトル $\mathbf{x} = (x_1, x_2, \cdots, x_m) \in 0, 1^m$（$m$ 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．

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

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

$$f_i(x) = \sum_{j=1}^{m} p_{i,j} \cdot x_j \tag{3}$$

（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）

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 つのナップサック問題を用いた．

4.1.2　ナップサック問題における多目的 GA の構成法

本研究では対象問題の 1 つとして多目的 0/1 ナップサック問題を用いた．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ってくるものの，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．

4.1.3　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．

4.1.4　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．

4.1.5　突然変異

突然変異には，ビット反転を用いた．また，個体の適合度改悪を防ぐために，ランクが 1（各世代におけるパレート最適解）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．

4.1.6　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．

4.1.7　制約外への個体の対応

今回用いたナップサ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対して何らかの対処が必要となる．本数値実験では，初期個体発生時において定義域外に存在する個体は認めず，制約を満たす個体数が必要個体数となるまで，個体の生成を行うものとする．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が定義域を越える場合もある．このような場合，この問題の特性を考え，制約を満たしていない個体の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．

4.2　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 1 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 1　Cluster system

CPU	Pentium II 500MHz *16
Memory	128 MB
OS	Debian GNU/Linux 2.2
Network	FastEthernet TCP/IP
Communication library	MPICH1.2.1

Table 2　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	-	-

Table 2 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 [1]，最適な値を求めることは非常に難しい．

そこで本研究では，各問題ごとに幾つかの個体数を用いて計算を行った．ナップザック問題においては，個体数として 400，4000 の値を使用した．

4.3　結果

4.3.1　ナップザック問題の結果

本研究では，対象とするナップザック問題として，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．

最終的なパレート解のプロット図の幾つかを Fig. 4〜Fig. 6 に示す．

結果より，解の幅広さという観点について考察する．DRMOGA は，全ての場合において最も良い結果が得られた．特に，個体数が 4000 個体の場合においては，他のアルゴリズムが問題の荷物数の増加に伴って改悪しているのに対して，DRMOGA のみが幅広い解を得ることができた．この傾向は，個体数が多い場合に特に顕著である．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,

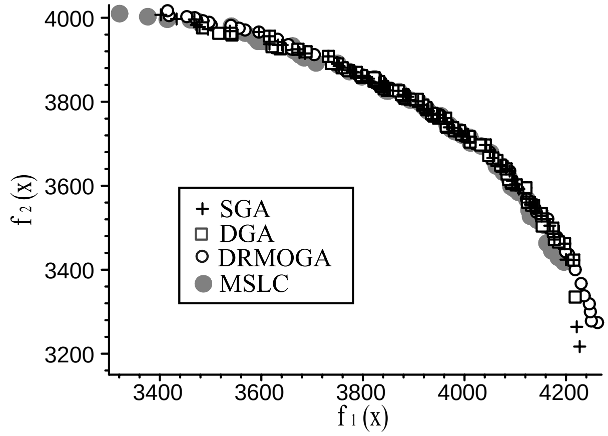

Fig. 4　100items problem (population size = 4000)

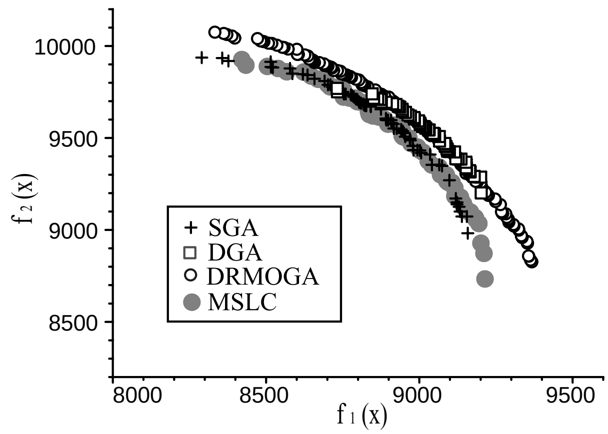

Fig. 5　250items problem (population size = 400)

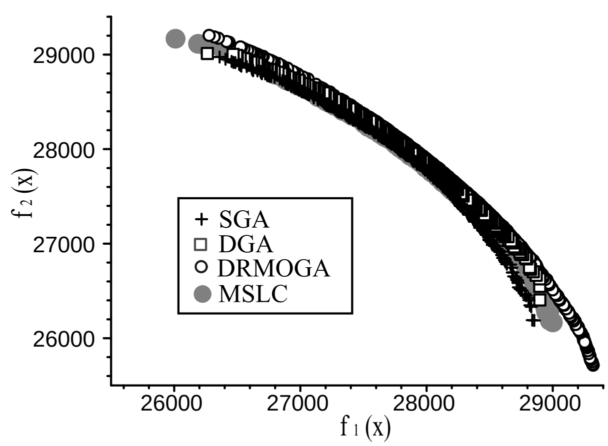

Fig. 6　750items problem (population size = 4000)

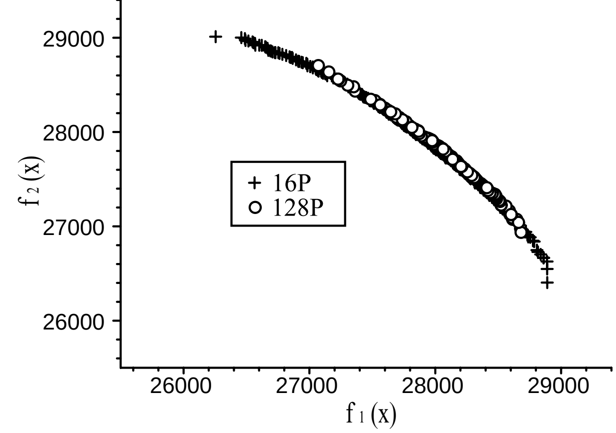

Fig. 7　DGA pareto optimum individuals (750items problem (population size = 4000)

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. 5）．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．

4.3.2　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 2）．

パレート解のプロット図の内，750 荷物の場合における結果を Fig. 7～Fig. 8 に示す．

Fig. 7～Fig. 8 より，DGA，DRMOGA 共に全ての

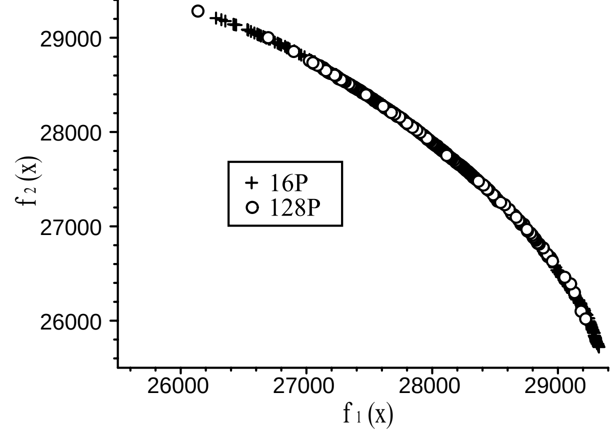

Fig. 8 DRMOGA pareto optimum individuals (750items problem (population size = 4000)

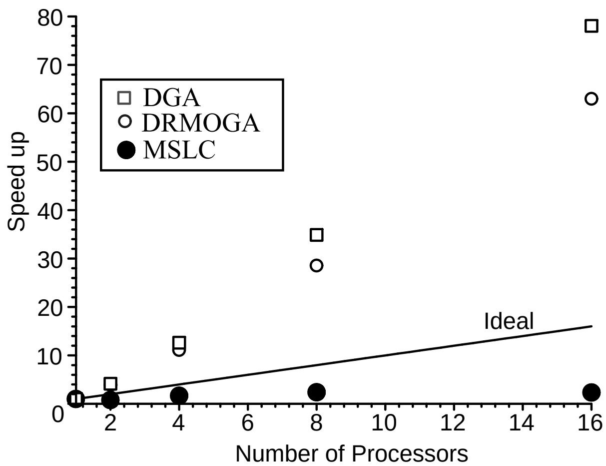

Fig. 9 Speed Up

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．

このこと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．

用いるプロセス数によって解が異なる最も大きな原因は，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．

並列化効率

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 2）．

Table 3 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

得られた結果を Table 3 に示す．また，各手法における並列化効率を示した図を Fig. 9 に示す．Table 9 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．

また，Fig. 9 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．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．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．

4.3.3 ネットワーク設計問題の結果

本研究では，上述のナップザック問題だけでなく，より複雑かつ計算負荷の高い携帯電話ネットワーク設計問題への適用も試みた．ネットワーク設計問題は，任意に定められた領域内において定められた候補サイトの中より実際にアンテナを設置するサイトを決定する問題であり，その目的は電波のカバー領域の最大化と設計コストの最小化として定義されている [4]．しかし，ネットワーク設計問題は問題自体が非常に複雑である上，遺伝的操作にも幾つかの特殊な操作を加える必要がある．そのため，ここでは問題に関する詳細な説明については割愛する．

ネットワーク設計問題では，80，160，320 の 3 通り

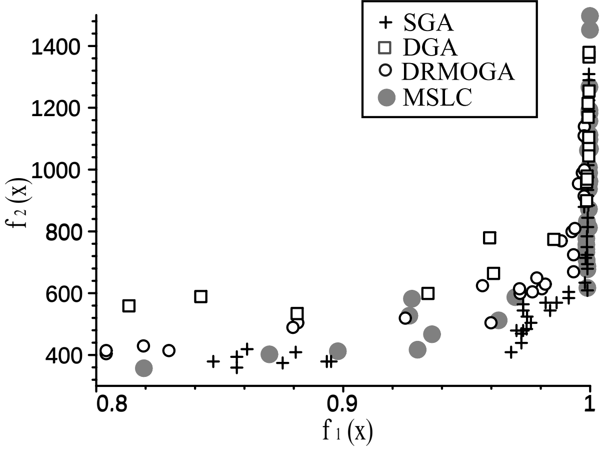

Fig. 10　Pareto optimum individuals (population size = 80)

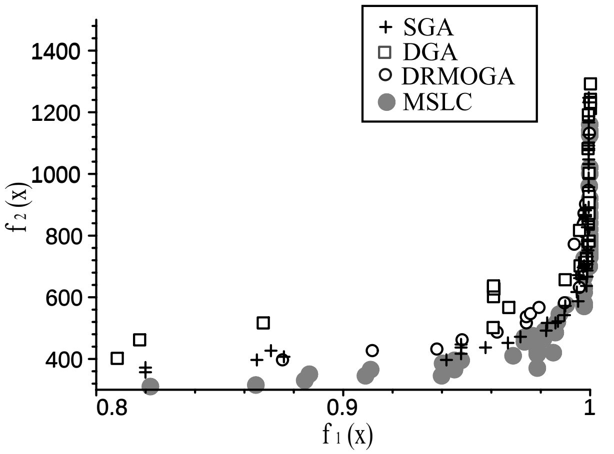

Fig. 11　Pareto optimum individuals (population size = 160)

の場合について実験を行った．ナップザック問題とアンテナ問題において用いる個体数が異なるのは，対象問題の評価計算部分における計算負荷が大きく異なるためである．

実験結果のプロット図を総個体数 80，160 個体の場合についてそれぞれを Fig.10，11 に示す．

結果より，上述のナップザック問題に比べ SGA, MSLC といった分割母集団モデルではない手法が良好な結果を示しているのが分かる．これは，用いている個体数が 80，160 個体と少ないため，DGA,DRMOGA といった分割母集団モデルでは 1 島内の個体数が極端に少なくなり，結果として効果的な探索が行えなかったためである．しかし，本適用問題は，評価計算負荷が非常に高いため 16 プロセス程度の並列環境においても 200 個体以上で探索することは現実的に難しい．

このように，評価計算負荷が高い問題に対しては，使用できる個体数が制限されてしまうため，領域分割型モデルでは十分な探索を行えない場合がある．そのような場合において，MSLC は非常に有効な手法の一つであると言える．

Table 4　Calculation time

method	Number of Processors	Times(sec)
SGA	1	19447.847378
DGA	1	19447.8
	2	8506.7
	4	4085.0
	8	2097.9
	16	1073.1
DRMOGA	1	19447.8
	2	9044.7
	4	4244.0
	8	2089.7
	16	1058.6
MSLC	1	23634.1
	2	23415.9
	4	7787.4
	8	3382.8
	16	1875.5

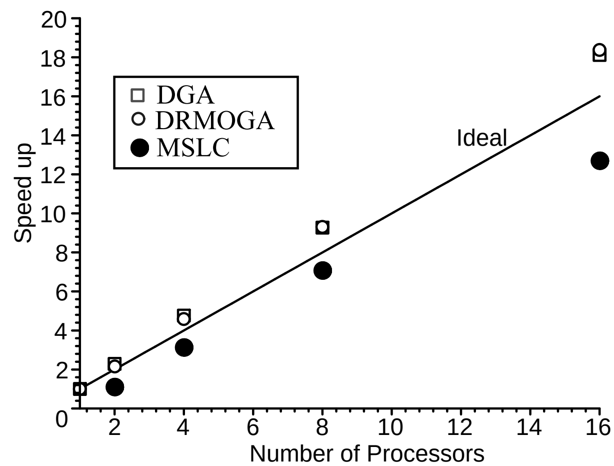

Fig. 12　Speed Up

並列化効率

ネットワーク設計問題における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，80 個体を 20 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 2）．

得られた結果を Table 4 に示す．また，各手法における並列化効率を示した図を Fig. 12 に示す．

結果より，各手法とも非常に線形に近い並列化効率が得られているのが分かる．これは，評価関数の計算負荷が非常に重いことに起因している．つまり，ネットワーク設計問題を解くための総時間における評価関数に必要となる計算時間の占める割合が非常に高いのである．そのため，各手法とも並列化効率という観点から見た場合，どの手法も線形に非常に近くなっているのである．

この結果より，MSLC はナップザック問題のような計算負荷が非常に軽い問題ではあまり良好な並列化効率が得られないものの，本適用問題のように計算負荷がある程度たかいような問題では良好な並列化効率が得られることが分かった．

5　結論

本研究では，多目的 GA を並列処理するための新たな分散モデルとして，DRMOGA と MSLC を提案し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．

- 従来の DGA に比べ DRMOGA は，どのような場合においても良好な結果を得ることができる．
- 分割母集団モデルである DGA,DRMOGA は，用いるプロセス数，島内の個体数によって大きく結果が異なる．特に，1 島内における個体数が充分に無い場合には良好な結果を得ることができない．
- 評価計算負荷が高いような問題においては，MSLC は最も良好な結果が得られた．逆に，用いる個体数に厳しい制限があるような場合，分割母集団モデルではあまり良好な探索が行えない．
- 評価計算負荷が軽い問題においては，DGA, DRMOGA は線形以上の並列化効率が得られた．一方，MSLC はあまり良好な並列化効率を得ることができなかった．しかし，評価計算負荷が高い問題ではどの並列手法においても線形に近い良好な並列化効率が得られた．

参考文献

1) C. A. Coello. An updated survey of evolutionary multiobjective optimization techiniques: State of the art and future trends. In *Proceedings of Congress on Evolutionary Computation*, pp. 1–11, 1999.

2) C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international coference on genetic algorithms*, pp. 416–423, 1993.

3) D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesly, 1989.

4) El-ghazali Talibi Herve Meunier and Philippe Reininger. A multiobjective genetic algorithm for radio network optimization. *In* Proceedings of International Workshop on Emergent Synthesis (IWES'99), pp. 317–324, 2000.

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 alogirithm. In *Proceedings of the 7th AIAI/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 1–11, 1998.

8) L. Nang and K. Matsuo. A survey on the parallel genetic algorithms. *J.SICE*, Vol. 33, No. 6, pp. 500–509, 1994.

9) 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.

10) 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.

11) 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.

12) 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.

13) 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.

14) 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.

15) 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.