

ヘテロ計算環境を想定した多目的遺伝的アルゴリズムの並列モデルの提案およびその検討

吉井 健吾[†] 廣安 知之^{††} 三木 光範^{††}

本研究ではヘテロ計算環境に対応した多目的遺伝的アルゴリズムの並列モデルを提案し、数値実験によりその有効性を検討する。提案モデルはマスタスレーブモデルを拡張し、交叉ペアとなる2個体をスレーブプロセスに送信する。そしてスレーブプロセスは計算資源の性能に適應して動的に生成子個体数を変化させることにより、すべての計算資源を最大限に利用することを実現する。また、目的関数空間において近接している2個体間で交叉を行う近傍交叉を取り入れ、マスタプロセスは近接した2個体をスレーブプロセスに送信することにより、探索性能の向上を図る。ヘテロな計算環境を使用した数値実験により、提案モデルは従来のマスタスレーブモデルと比較して高い並列度および優れた解探索性能を示した。また提案モデルではすべての計算資源を効率的に使用可能であること、そしてオーバヘッドの影響を軽減可能であることを確認した。

Discussion of Parallel Model of Multi-objective Genetic Algorithms on Heterogeneous Computational Resources

KENGO YOSHII,[†] TOMOYUKI HIROYASU^{††} and MITSUNORI MIKI^{††}

In this paper, a parallel model of multi-objective genetic algorithm supposing a heterogeneous environment is discussed. In this proposed parallel model, we extended master-slave model, and 2 individuals as a crossover pair are transmitted to each slave process. Then the number of offspring generated by crossover is changed dynamically adapting to the performance of the each calculation resource. This mechanism is effective for heterogeneous computational resources. Moreover, we incorporated the neighborhood crossover, in which the crossover is performed between individuals that are close to each other in the objective space. Therefore, 2 individuals which are close to each other are sent to each slave process. This neighborhood crossover improves the search ability. Computational experiments indicated that the proposed model has high search ability, and was able to utilize the maximum performance of all calculation resources and reduce the overhead time.

1. はじめに

実最適化問題の多くには複数の評価基準が存在し、評価基準が互いにトレードオフの関係にあることが多い。このような問題を多目的最適化問題としてとらえ、遺伝的アルゴリズム (Genetic Algorithm: GA) を応用した多目的 GA に関する研究がさかに行われている^{1)~9)}。多目的 GA はどの解にも劣らない解の集合であるパレート最適解集合を一度に求めることが可能であることから、有効であるとされている。一方、多目的 GA の問題点として高い計算負荷があげられ

る。実問題の多くは膨大な計算時間を必要とすることから、並列処理により計算時間を短縮させることは重要な課題であるといえる。多目的 GA の並列に関する研究も数多くなされているが、その並列度は小さい場合が多い^{10),11)}。一方、世界中の計算資源を仮想的に統合したグリッドコンピューティングの技術の進歩により、数多くの資源を容易に使用可能になりつつある。そのためグリッドのようなヘテロな計算環境を利用した並列モデルについても考慮する必要がある。

本論文の目的はヘテロ計算環境に対応した多目的 GA の並列モデルを提案し、その有効性を検証することである。ヘテロ計算環境で並列化を行う際の問題点を整理し、その問題点を解決できる仕組みを考慮した並列モデルの提案を行う。そして代表的な多目的 GA の手法である NSGA-II³⁾ を提案モデルに組み込み、従来の並列モデルとの比較検討を行う。

[†] 同志社大学工学部大学院
Graduate School of Engineering, Doshisha University

^{††} 同志社大学工学部
Knowledge Engineering Department, Doshisha University

2章では多目的GAの並列化について説明し,3章で提案並列モデルであるグリッド環境における並列多目的GAについて述べる.その後,4章で提案モデルのアルゴリズムの性能調査を数値実験により行い,5章でヘテロ計算環境における数値実験により提案モデルの有効性を検証する.最後に6章で結論を述べる.

2. 並列多目的遺伝的アルゴリズム

2.1 多目的遺伝的アルゴリズムにおける並列処理の必要性

実問題の多くは最適解を求めるまでに膨大な計算量を必要とし,単目的GAと同様に多目的GAにおける並列処理の必要性も高い.特に多目的GAにおいては,その必要性は以下の理由により単目的GAと比較してより高くなると考えられる.

- 評価する目的関数が複数存在する.
- 目的関数の増加にともない,パレート最適フロントの次元も高くなるため,良好なパレート最適解集合を得るためには,母集団サイズを大きく設定する必要がある.

多目的最適化問題は複数の目的関数を有しているため,目的関数の増加にともない1回の評価に要する時間も大きくなるといえる.また,目的関数の増加にともない,パレート最適フロントの次元も高くなるため探索も困難となる.あるテスト関数における目的関数の数と初期母集団に存在する非劣解の割合の関係を図1に示す.図1において N は母集団サイズを示している.図1から目的関数の増加にともない,初期母集団内に存在する非劣解の割合が高くなるのが分かり,そしてその割合は母集団サイズ N の増加にともない軽減されていることが分かる.初期母集団から非劣解の割合が高くなるにともない,探索が効果的に行われなくなるため,良好なパレート最適解集合を形成させるためには,母集団サイズを大きく設定する必要がある.母集団サイズを大きく設定すると,1世代あたりの評価計算回数も増加するため,多目的GAにおける並列処理の必要性はますます高くなるといえる.

2.2 多目的遺伝的アルゴリズムにおける並列モデル

多目的GAの並列処理に関する議論は多目的GAの研究の初期の頃から比較して近年増えつつあり¹⁰⁾⁻¹⁵⁾,並列モデルとしてマスタスレーブモデルと分割母集団モデルが多く用いられている.本節では並列モデルについて述べる.

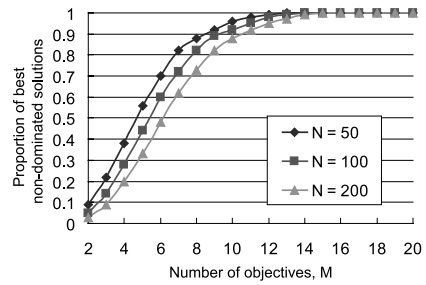


図1 目的関数の数と初期母集団に存在する非劣解の割合(出典:参考文献10))

Fig.1 The proportion of the non-dominated solutions in initial population versus the number of objective functions (Source: Ref.10)).

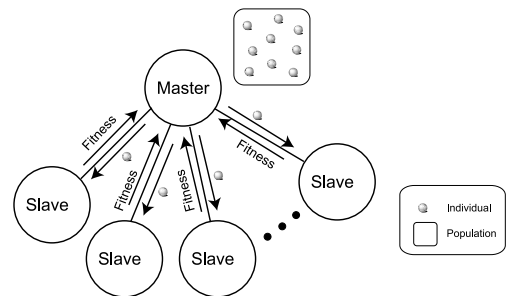


図2 マスタスレーブモデル

Fig.2 Master-slave model.

2.2.1 マスタスレーブモデル

マスタスレーブモデルは,マスタプロセスがGAの遺伝的オペレータを実行し,全体の計算の中で負荷の高い評価計算を複数のスレーブプロセスで並列化するモデルであり,図2に示すトポロジをとる.マスタスレーブモデルは, P プロセッサを使用した場合約 P 倍の計算時間の短縮が可能であり,並列度が高いモデルであるといえる.また逐次モデルをそのまま適用可能であるため実装も容易であり,得られる結果も逐次モデルと同等であることが特徴である.一方,マスタプロセスとスレーブプロセス間の通信頻度は高いため,通信負荷が分割母集団モデルよりも高いという欠点がある.

2.2.2 分割母集団モデル

分割母集団モデルは島モデルとも呼ばれ,1つの母集団を複数のサブ母集団に分割し,サブ母集団単位にプロセッサを割り当てて並列化を行うモデルであり,図3に示すトポロジをとる.このモデルでは,それぞれのプロセッサでGAが逐次に行われ,定期的いくつかの個体がプロセッサ間で交換されること(移住)により探索空間における多様性の維持を図る仕組みが用いられている.分割母集団モデルの長所として,

一般に多目的GAの探索段階における,他のどの解にも優越されない解のことを非劣解と呼ぶ.

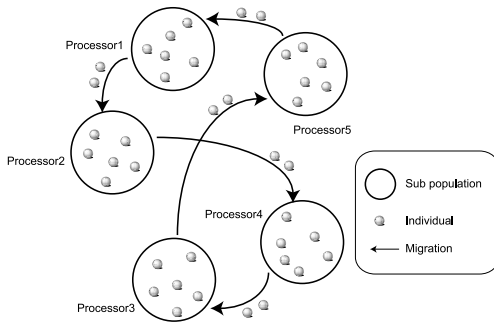


図 3 鳥モデル
Fig. 3 Island model.

定期的にプロセッサ間で通信が行われるため、マスタスレーブモデルと比較して通信負荷が少ないことや、対象問題によっては求まる解の精度が向上することがあげられる。一方、サブ母集団の数によりプロセッサ数が制約されるため、マスタスレーブモデルと比較して並列度は低いことが短所としてあげられる。

2.3 並列多目的遺伝的アルゴリズムに関する研究の現状

多目的 GA の並列化に関する議論は、アルゴリズムの改良に関する議論とともに増えつつあるが、多くの研究が分割母集団モデルを用いた並列化について議論している^{10)~13),15)}。Deb らは分割母集団モデルを利用し、プロセッサごとに目的関数ベクトルを変化させ、各プロセッサが持つサブ母集団がパレート最適フロントのそれぞれ異なる部分フロントを探索することを可能としている¹⁰⁾。Branke らは Cone Separation という方法により探索過程で得られた非劣解集合を目的関数空間において複数に分割し、分割母集団モデルにより並列化を行っている¹⁵⁾。また Streichert らは、分割母集団モデルにおけるサブ母集団の生成に、クラスタリングのアルゴリズムを用いることにより、適切にサブ母集団を分類して並列化を行うことを検討している¹¹⁾。

これらのように、多くの研究が、通信負荷を軽減かつ探索性能を向上させる目的で分割母集団モデルを利用している。しかし、分割母集団モデルでは用いるプロセッサ数が島数に制限されるため、並列度は高いとはいえない。一方、近年では安価な PC の高性能化、ネットワークの高速化にともない、ネットワークにつながれた数多くの PC を計算資源として容易に使用可能となりつつある。特に、世界中のインターネット上に存在する PC やサーバおよびストレージ等を仮想的に統合したグリッドコンピューティングの技術により、グリッドミドルウェアを利用して容易に計算資源を利

用できる環境が整いつつある。しかし、これらの大規模な計算環境を想定した、またはヘテロな計算環境を想定した並列多目的 GA の研究は行われておらず、このような環境を想定した並列モデルの検討を行う必要性は高いと考えられる。

本論文の目的はヘテロ計算環境の特徴を洗い出し、ヘテロ計算環境で並列化を行う際の問題点を整理し、その問題点を解決できる仕組みを考慮した並列モデルを検討することである。次章以降では、検討モデルの説明および数値実験による検討モデルの有効性について議論を行う。

3. ヘテロ計算環境における並列多目的遺伝的アルゴリズム

本章では、ヘテロ計算環境を想定した多目的 GA の並列モデルについての提案を行う。ヘテロ計算環境の特徴を洗い出し、従来のモデルを用いて並列化を行う場合の問題点を整理した後、その問題点を考慮した並列モデルの要求定義を行う。その後、提案モデルの説明を行う。

3.1 ヘテロ計算環境の特徴と従来の並列モデルにおける問題点

ヘテロ計算環境の特徴を以下に示す。

- それぞれ性能が異なる計算資源が存在
- オーバヘッドの増大
- 大規模な計算環境

ヘテロ計算環境に存在する計算資源はそれぞれ性能が異なるため、計算資源の性能に適応した計算負荷を考慮する必要がある。またヘテロ計算環境に存在する計算資源はインターネット上に存在していることが考えられるため、通信時間などのオーバヘッドは PC クラスタ内で並列処理を行う場合と比較して大きくなるといえる。そしてヘテロ計算環境では計算資源の規模は無敵大であり、並列度の高いマスタスレーブモデルが有効であると考えられる。しかし、ヘテロ計算環境にマスタスレーブモデルをそのまま適用すると以下の問題が生じる。

- ヘテロ計算環境に存在する計算資源はそれぞれ性能が異なり、評価計算時間も異なる。そのため毎世代同期をとる必要がある GA においては、性能の劣る計算資源が世代交代を遅延させる原因となる。そのため、性能の優れた計算資源ほどアイドル時間が増加し、結果として並列度の低下につながる。
- マスタスレーブモデルではマスタプロセスとスレーブプロセス間の通信頻度が多いため、イン

ターネット上に存在するグリッド環境では通信時間も大きくなり、オーバーヘッドの影響も大きくなる。

したがって、これらの問題を考慮した並列モデルを検討する必要がある。

3.2 要求定義

3.1 節で示した問題点をふまえ、ヘテロ計算環境における多目的 GA の並列モデルの要求定義を行う。

- 高い並列度
- ヘテロな計算環境を効果的に利用可能
- オーバヘッドの影響を軽減可能
- 並列モデルの変更にもなう解探索性能低下の阻止

ヘテロ計算環境には膨大な計算資源が存在するため、高い並列度を有するモデルを考慮することが必須である。次にヘテロな計算資源を無駄なく最大限に使用できる仕組み、およびオーバーヘッドの影響を軽減できる仕組みを考慮してモデルに組み込む必要がある。最後に逐次モデルから変更した際に、解探索性能が改悪しないことが最低条件である。

本研究では、これらの要求定義をすべて満たす多目的 GA の並列モデルを提案し、その有効性を検討する。

3.3 基本モデル

本節では提案モデルの基本モデルについて説明を行う。提案モデルの主な特徴を以下に示す。

- 高い並列度を有するマスタスレーブモデルを拡張
- ヘテロな計算環境およびオーバーヘッドの軽減を考慮した、計算資源の性能に適應した生成子個体数の変化
- 探索性能を考慮した、近傍交叉

提案並列モデルの概念図を図 4 に示す。まず、提案モデルは並列度の高いマスタスレーブモデルを拡張する。通常のマスタスレーブモデルは評価する個体をスレーブに送信するのに対し、提案モデルでは交叉ペアとなる 2 個体をスレーブに送信する。そして 2 個体を受信したスレーブは交叉、突然変異および評価を行う。この際、計算資源の性能に適應して動的に交叉回数を変化させ、生成子個体数を変化させる仕組みを取り入れる。すなわち、性能の優れた計算資源ほど数多くの子個体を生成し、評価を行うことにより探索に貢献することが可能である。この仕組みにより、ヘテロな計算環境を最大限に利用することができると考えられる。また、スレーブにおける処理を増大させることにより、全体の通信回数を削減し、オーバーヘッドの影響が軽減可能となる。このとき重要となるのは、マスタがスレーブに送信する 2 個体をどのように決定する

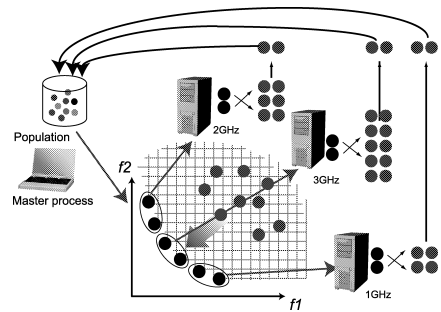


図 4 提案並列モデル

Fig. 4 Proposed parallel model.

かである。提案モデルでは探索性能を考慮し、個体間の近接度合いを考慮して目的関数空間において近接している個体どうして交叉をさせる近傍交叉を取り入れている。すなわち、近接した 2 個体をスレーブに送信し、各スレーブは近接した 2 個体間で交叉を行うことにより、探索性能の向上を図る。

3.4 近傍交叉

一般に代表的な多目的 GA の手法では、交叉ペアとなる個体はランダムに選ばれ、個体間の設計変数空間および目的関数空間における距離が大きく離れ、効果的な探索ができないという問題点が存在する。そのため、目的関数空間で近接している個体どうして交叉を行う近傍交叉を取り込んだ多目的 GA が提案され、他の手法と比較して優れた性能を示している^{16),17)}。近傍交叉のアルゴリズムを以下に示す。また近傍交叉の概念図を図 5 に示す。

- (1) ある目的関数を基準に最も適合度の高いまたは低い個体を選択し、その個体から目的関数空間において近接している順にソートを行う。この際、ソートの基準となる目的関数は毎世代異なるものとする。
- (2) ソート後の探索個体群に対して母集団サイズの十分小さいある一定の幅の間隔において近傍シャッフルを行う。近傍シャッフルとは、母集団内の個体を 1 割程度の幅の間隔においてランダムに並べ替えるものであり、繰り返し同じペアで交叉を行うことを防ぐために行う。近傍シャッフルの様子を図 6 に示す。
- (3) 隣り合う 2 個体を交叉ペアとして母集団サイズ/2 の数の交叉ペアを作成し、各交叉ペアに対して 1 点交叉もしくは多点交叉を行う。

3.5 計算資源の性能に適應した生成子個体数の増加
一般的な多目的 GA では、2 個体の親個体から 1 回の交叉により 2 個体の子個体を生成する。一方、提案モデルでは計算資源の性能に適應して交叉回数を動的

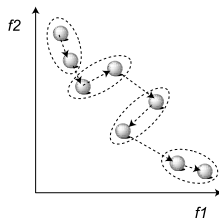


図 5 近傍交叉

Fig. 5 Neighborhood crossover.

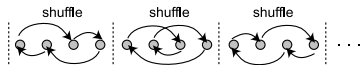


図 6 近傍シャッフル

Fig. 6 Neighborhood shuffle operator.

に変化させ、生成される子個体の数を変化させる仕組みを取り入れている。つまり性能の優れた計算資源では多くの子個体を生成し、性能の劣る計算資源では少ない子個体を生成することになる。提案モデルでは、ユーザが与えた一定時間内で可能な限り子個体を生成し、突然変異、評価を行うものとする。一定時間内に可能な限り子個体を生成、評価させることにより、計算資源の性能に適應した動的な負荷の決定を実現している。一定時間後、各スレーブプロセスは生成された子個体集合から最も優れた 2 個体を選択し、マスタプロセスに返信する。以下にこのアルゴリズムの流れを示す。

- (1) スレーブプロセスはマスタプロセスから 2 個体を受信し、空の生成子個体集合 C を作成する。
- (2) 受信した 2 個体に対して交叉を行い、1 つの子個体 i を生成する。
- (3) 生成子個体 i に対して突然変異を行い、評価を行う。
- (4) 生成子個体 i を生成子個体集合 C に追加する。
- (5) マスタから 2 個体を受信してから与えられた一定時間に達していない場合は、 $i = i + 1$ としステップ (2) に戻る。
- (6) 生成子個体集合 C から非劣解集合 S を抽出する。
- (7) 非劣解集合 S からいずれかの目的関数値が最も良好である 2 個体を選択する。非劣解の数が 1 の場合は、その非劣解を除いた生成子個体集合 C から非劣解集合 S を再度抽出し、いずれかの目的関数値が最も優れた 1 個体を選択する。
- (8) 選択された 2 個体をマスタプロセスに返信する。
- (9) 上記ステップ (1) から (8) をすべてのスレーブプロセスが行い、すべてのスレーブプロセスが

ら 2 個体ずつ受信したマスタプロセスは、アーカイブ母集団を更新し、次世代の探索母集団を形成する。

なおこのアルゴリズムでは、生成子個体数が増加するほど、プロセスにおける 1 世代あたりの評価計算回数も増加する。また生成子個体集合のうち選択された 2 個体以外は淘汰されるため、精度の高い個体群のみ生き残る仕組みになっている。

3.6 提案モデルの特徴

本提案モデルの特徴を以下にまとめる。

- 高い並列度
- ヘテロな計算環境を最大限に利用可能
- オーバヘッドの影響を軽減
- アプリケーションユーザの負荷を軽減

まず、提案モデルはマスタスレーブモデルを拡張しているため高い並列度を有している。また評価だけでなく交叉、突然変異の遺伝的操作も並列化を行っているため、オリジナルマスタスレーブモデルよりもより並列度を向上させることが可能といえる。次に各スレーブプロセスの性能に適應した動的な負荷の決定を実現しており、ヘテロな計算環境を最大限に利用することが可能である。そして、スレーブプロセスにおける処理を通常よりも増大させているため、全体の通信回数の削減が可能となり、オーバヘッドの影響の軽減も期待できる。またアプリケーションユーザの負荷を軽減することも可能であるといえる。マスタプロセスの性能を把握する必要はなく、2 個体ずつ平等に各スレーブプロセスにジョブを投入するだけであり、スレーブプロセス側で動的に負荷を決定する仕組みとなっている。

3.7 提案モデルの検討課題

本提案モデルでは、ヘテロな計算環境を考慮し、計算資源の性能に適應して生成子個体数を動的に変化させる仕組みを取り入れている。また探索性能を考慮し、近傍交叉の仕組みを取り入れている。そのため、提案モデルの有効性を検証する前に、近傍交叉の効果および生成子個体数の増加にともなう解探索性能への影響を調査する必要があると考えられる。

次章ではテスト関数を用いた数値実験により、近傍交叉の効果および生成子個体数の増加にともなう解探索能力への影響の調査を行う。

探索母集団とは別に優れた個体を保存する母集団のことをアーカイブ母集団と呼ぶ。

4. 数値実験によるアルゴリズムの性能評価

本章では、テスト関数を用いた数値実験により、近傍交叉の効果および生成子個体数の増加にともなう探索性能への影響の調査を行う。

4.1 対象問題および性能評価方法

本実験では、連続問題として Kursawe の数値実験に使用された KUR¹⁸⁾ および離散問題として Zitzler らの数値実験により使用された多目的ナップザック問題の 2 目的 750 荷物問題 (KP750 - 2)^{4),7)} を取り扱う。KUR

$$\begin{cases} \min & f_1 = \sum_{i=1}^n (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})) \\ \min & f_2 = \sum_{i=1}^n (|x_i|^{0.8} + 5 \sin(x_i)^3) \\ \text{s.t.} & \\ & x_i \in [-5, 5], i = 1, \dots, n, n = 100 \end{cases} \quad (1)$$

KUR は $f_1(x)$ において連続する 2 変数間の相互作用を持ち、 $f_2(x)$ において多峰性を有する問題である。本実験では、この問題を 100 個の設計変数を持つ問題として扱い、探索をより困難とさせる。

KP750 - 2

$$\begin{cases} \max & f_i(x) = \sum_{j=1}^{750} x_j \cdot p_{(i,j)} \\ \text{s.t.} & \\ & g_i(x) = \sum_{j=1}^{750} x_j \cdot w_{(i,j)} \leq W_i \\ & 1 \leq i \leq k, k = 2 \end{cases} \quad (2)$$

多目的ナップザック問題は、非常にシンプルで実装しやすい反面、問題自体は探索が非常に困難である。上式における $p_{(i,j)}$ および $w_{(i,j)}$ は、それぞれ i 番目の評価値を計算する際の j 番目の荷物に付随する利益値と重み値を表している。また、 W_i は i 番目の評価値計算を行う際の重み値の総和に対する制約値（上限値）である。

得られたパレート最適解集合を評価する手法は数多く存在するが、本研究では以下に示す評価手法を使用する。

- (1) 被覆率 (I_{cover})¹⁹⁾
- (2) パレート最適解集合の幅広さの評価 (Spread)²⁰⁾
- (3) Hypervolume⁷⁾
- (4) 優越個体割合 (Ratio of Non-dominated Individuals: RNI)²¹⁾

被覆率 (I_{cover}) は得られたパレート最適解集合を、多様性に関して絶対的に評価する方法である。目的関数空間におけるパレート最適解領域において、解集合が均一に分布しているかを評価することが可能である。

I_{cover} は各目的関数空間のパレート最適解領域を K 分割したときの、目的関数 i においてパレート最適解が存在している小領域の数 k_i の割合により求められる。 N 目的関数の対象問題における I_{cover} を求める式を次に示す。

$$I_{cover} = \frac{1}{N} \sum_{i=1}^N \frac{k_i}{K} \quad (3)$$

上記の式より、 I_{cover} の値が 1.0 に近いほど、解が全領域に求まっていると評価され、多様性の優れたパレート最適解集合が得られていると判断することができる。本実験では分割数 K を母集団サイズとする。

パレート最適解集合の幅広さの評価 (Spread) は、以下の式で計算され、値が大きいくほど幅広いパレート最適解集合が得られていることを確認することができる。

$$\text{Spread} = \sum_{i=1}^N \{\max f_i(x) - \min f_i(x)\} \quad (4)$$

Hypervolume は得られたパレート最適解集合が支配している領域の大きさを計算したものであり、パレート最適フロントへの収束度、多様性、幅広さを総合的に評価することができる。

優越個体割合 (RNI) は、Lee らによって用いられた手法²¹⁾ を 2 つの非劣解集合の比較へと拡張したものである。RNI ではまず、2 つの手法で得られた解集合 X と Y の和集合をとり S^U とする。次に、 S^U の中から、非劣解のみを選び出し、選ばれた非劣解集合を S^P とする。そして、 S^P の各手法の割合を $I_{RNI(X,Y)}$ として導き出すというものである。このため、この割合は最大値の 100% に近いほど、もう一方の手法を優越している、すなわち、より真の解に近い解が得られているものと判断することができる。

なお、本実験に使用する多目的 GA の共通パラメータを表 1 に示す。

4.2 近傍交叉の効果

近傍交叉において、近傍シャッフルは非常に重要な操作である。近傍シャッフルを行わなければ、毎世代同じペアによる交叉が行われる可能性が増し、局所解に陥った場合は抜け出せなくなる恐れがある。したがって、適度な大きさの幅において近傍シャッフルを行うことが重要である。この近傍シャッフルを行う幅（近傍シャッフル幅）の大きさはパラメータであり、近傍シャッフル幅率 (R_{nsw}) により決定される。 R_{nsw} は 0 ~ 1.0 までの実数であり、母集団サイズの割合による近傍シャッフル幅の大きさを示す。たとえば、 $R_{nsw} 0.1$

表 1 多目的 GA におけるパラメータ

Table 1 Parameters in Multi-Objective GA.

Problem	KUR	KP750-2
Population Size	100	250
Number of Dimensions	100	
Chromosome Length	2000	750
Crossover Probability		1.0
Crossover	two points crossover	
Mutation Probability	1/Chromosome Length	
Max Generation	250	2000

は母集団サイズの 1 割の幅の大きさで近傍シャッフルを行うことを意味する。 R_{nsw} の大きさにより個体どうしの近接度合いは変化し、小さくなるほど近接度合いは増すが、同じペアで繰り返し交叉が行われる可能性も高くなる。本節では、近傍交叉の効果を調査するため、代表的な多目的 GA の手法である NSGA-II³⁾ に近傍交叉を組み込み、数値実験により近傍シャッフル幅の変化による解探索能力への影響について検討を行う。なお、オリジナル NSGA-II では、アーカイブ母集団から非優越性と混雑度³⁾に従ってトーナメント選択により探索母集団を生成させているが、近傍交叉ではできるだけ異なる個体からなる母集団が必要となるため、アーカイブ母集団をそのまま探索母集団にコピーを行うものとする。

4.2.1 近傍シャッフル幅の変化による解探索能力への影響

KUR における R_{nsw} の変化による I_{cover} , Spread, Hypervolume, およびオリジナル NSGA-II と比較した RNI を図 7 に示す。なお、実験データは 30 回試行の平均値であり、比較のためオリジナル NSGA-II の結果も示している。

図 7 の結果から、特に R_{nsw} が 0.05 から 0.2 のとき、高い精度が得られていることが分かり、近傍交叉の効果を確認することができる。 R_{nsw} 0.0 のとき、つまり近傍シャッフルを行わないときは、同じペアにより交叉が行われる頻度が増すため、探索に影響が出たと考えられる。オリジナル NSGA-II では、アーカイブ母集団から復元抽出の選択を行い探索個体群を生成させているため、探索個体のパレート最適解に対する収束は早くなるものの、母集団は多様性を失いやすくなり、局所解にも陥りやすい。そのため、KUR のように多峰性のある問題において良好な解を得ることは難しい。しかし、近傍交叉を組み込むことで多様性を維持した探索が可能となり、探索能力を向上させることができる。KUR においてオリジナル NSGA-II と比較した R_{nsw} 0.1 における 30 回試行で得られたすべてのパレート最適解集合のプロット図を図 8 に示す。

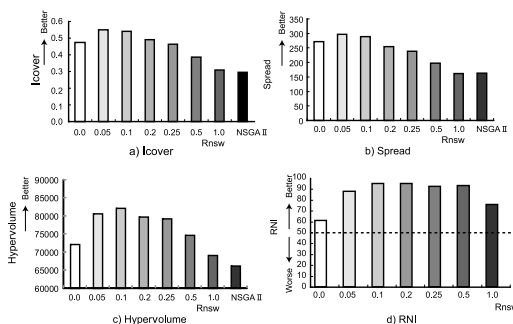


図 7 KUR における R_{nsw} の変化させたときの実験結果 (Icover, Spread, Hypervolume, RNI)
Fig. 7 Results of Icover, Spread, Hypervolume and RNI when R_{nsw} is changed in KUR problem.

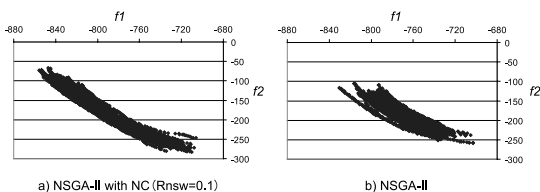


図 8 KUR における得られたパレート最適解
Fig. 8 Obtained pareto-optimal solutions in KUR problem.

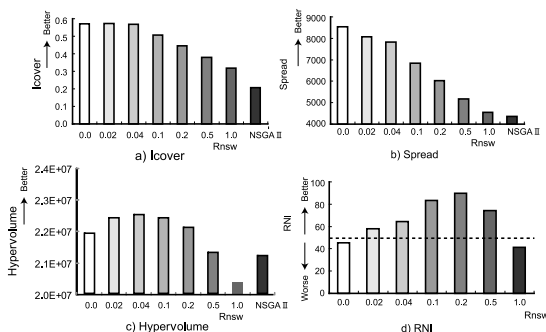


図 9 KP750-2 における R_{nsw} の変化させたときの実験結果 (Icover, Spread, Hypervolume, RNI)
Fig. 9 Results of Icover, Spread, Hypervolume and RNI when R_{nsw} is changed in KP750-2 problem.

近傍交叉の効果は視覚的にも確認でき、多様性に優れた幅広いパレート最適解集合が得られていることが分かる。

同様に KP750-2 の結果を図 9 に、オリジナル NSGA-II と比較した R_{nsw} 0.1 における 30 回試行で得られたすべてのパレート最適解集合のプロット図を図 10 示す。

KP750-2 は、非常に幅広いパレート最適フロントを持つ問題であり、パレート最適フロントを形成するパレート最適解の設計変数値も多様性に富んでいる。

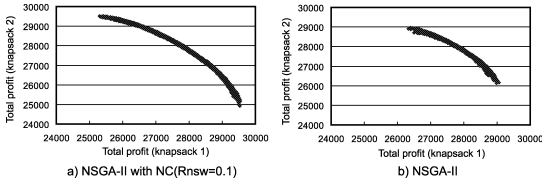


図 10 KP750-2 における得られたパレート最適解

Fig. 10 Obtained pareto-optimal solutions in KP750-2 problem.

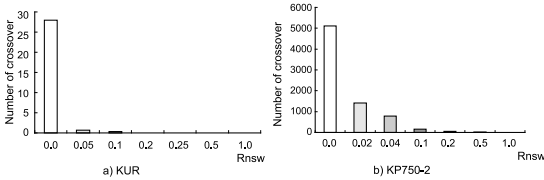


図 11 前世代と同じペアにより交叉が行われた回数

Fig. 11 The number of times that crossover was performed in the same pair as previous generation.

そのため、KUR と同様、幅広いパレート最適解集合を探索するには、母集団の多様性が非常に重要となってくる。結果として、図 9 および図 10 から多様性の優れた幅広いパレート最適解集合が得られていることが分かり、近傍交叉の有効性が確認できる。

4.2.2 近傍交叉に関する考察

5.2.1 項で様々な近傍シャッフル幅率を検討した結果、KUR, KP750-2 どちらの対象問題に対しても、 $R_{nsw}0.0$ では RNI が劣っていることが分かる。 $R_{nsw}0.0$ のとき I_{cover} および Spread は向上するが、シャッフルを行わないため前世代と同じペアの個体間での交叉が行われる頻度も増し、探索に影響が出たと考えられる。このことを確認するため、図 11 に、各対象問題における前世代と同じペアの個体により交叉が行われた回数を示す。図 11 から、近傍シャッフル幅が小さいほど前世代と同じペアの個体間による交叉が多く行われていることが確認できる。このことから、適切な近傍シャッフル幅による近傍シャッフルは、近傍交叉において重要であることが分かった。

4.3 生成子個体数増加の効果

本節では、生成子個体数の増加に対する解探索性能への影響を調査するため、近傍交叉を組み込んだ NSGA-II に生成子個体数増加のメカニズムを組み込み、テスト関数により数値実験を行う。生成子個体数のパラメータとして、2, 4, 6, 8, 10, 20 の 6 つのパラメータを用い、同じ評価計算回数を行ったときの比較および同じ世代数を経た場合の比較を行う。同じ評価計算回数を行ったときの比較では、生成子個体数

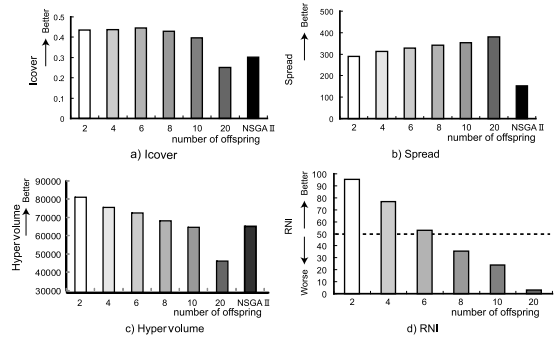


図 12 KUR における同じ評価計算回数を行ったときの実験結果 (Icover, Spread, Hypervolume, RNI)

Fig. 12 Results of Icover, Spread, Hypervolume, and RNI when the number of evaluations is fixed in KUR problem.

のパラメータを増加させるにともない、1 世代あたりの評価計算回数は増加し、全体の世代数は減少することになる。一方、同じ世代数を経た場合の比較では、生成子個体数のパラメータを増加させるにともない、全体の評価計算回数は増加することになる。

4.3.1 生成子個体数増加による解探索能力への影響

同じ評価計算回数を行ったときの KUR における実験結果を図 12 に示す。実験結果は 30 試行の平均値であり、図 12 a) は I_{cover} を、b) は Spread を、c) は Hypervolume を、そして d) はオリジナル NSGA-II と比較したときの RNI を示している。また、比較のためオリジナル NSGA-II の結果も示している。

図 12 a) から全体的に生成子個体数増加にともない多様性が失われていることが確認できる。また図 12 c) から生成子個体数増加にともない性能が劣る傾向にあり、図 12 d) の RNI から生成子個体数 8 以上はオリジナル NSGA-II と比較して劣っていることが分かる。これらの原因として、生成子個体数を増加させるにともない、1 世代あたりの評価計算回数も増加し、十分な世代数を経ることができなくなることが考えられる。一方、図 12 b) から、生成子個体数の増加にともない Spread が向上していることが確認でき、幅広いパレート最適解集合が得られることが分かる。

同様に KP750-2 における実験結果についても図 13 に示す。図 13 から KP750-2 に関しても、生成子個体数の増加にともない I_{cover} , Hypervolume および RNI が劣る傾向があり、Spread が向上していることが分かる。これらのことから、生成子個体数の増加にともない、1 世代あたりの評価計算回数が多くなるため収束が遅くなるが、幅広いパレート最適解集合を探索可能であることが分かった。

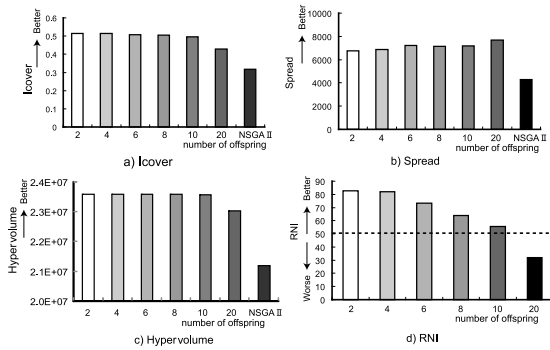


図 13 KP750-2 における同じ評価計算回数を行ったときの実験結果 (Icover, Spread, Hypervolume, RNI)
 Fig.13 Results of Icover, Spread, Hypervolume, and RNI when the number of evaluations is fixed in KP750-2 problem.

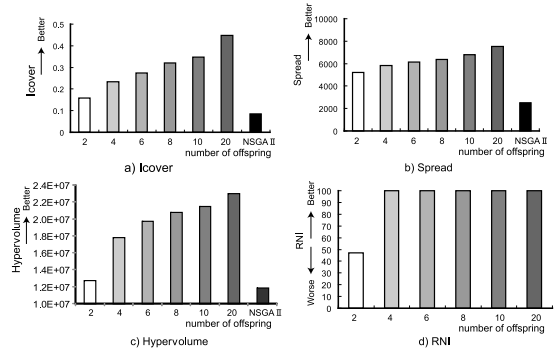


図 15 KP750-2 における同じ世代数を実行したときの実験結果 (Icover, Spread, Hypervolume, RNI)
 Fig.15 Results of Icover, Spread, Hypervolume, and RNI when the number of generations is fixed in KP750-2 problem.

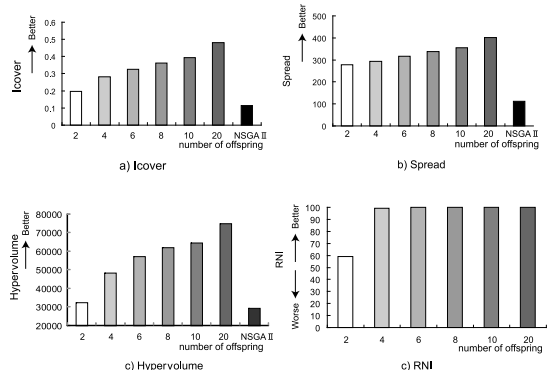


図 14 KUR における同じ世代数を実行したときの実験結果 (Icover, Spread, Hypervolume, RNI)
 Fig.14 Results of Icover, Spread, Hypervolume, and RNI when the number of generations is fixed in KUR problem.

次に同じ世代数を経た場合の生成子個体の増加による解探索性能の比較検討を行う。提案並列モデルでは計算資源の性能に適應して生成子個体数が変化するため、評価計算回数を固定して比較するだけでなく、世代数を固定して比較検討する必要がある。世代数を固定することにより、生成子個体の増加が解の探索能力に与える影響を確認することができ、計算資源の性能に適應して生成子個体数を変化させる利点を実証できる。世代数のパラメータとしては、KUR では 50 とし、KP750-2 では 200 とする。

図 14 に KUR における、世代数 50 と固定した場合の生成子個体の増加による実験結果を示す。また図 15 に KP750-2 における、世代数を 200 に固定した場合の実験結果を示す。

図 14 および図 15 から、両対象問題において生成

子個体数の増加にともない、すべての評価手法において優れた性能を示し、多様性に優れた幅広いパレート最適解集合が得られていることが確認できる。RNI に関しては、生成子個体数 4 以上のとき、100% となり NSGA-II と比較してすべての解において強く優越する結果となった。得られたパレート最適解集合を確認するため、KUR における生成子個体数 2, 10, 20 の 30 試行で得られた全パレート最適解集合のプロット図を図 16 に、KP750-2 における生成子個体数 2, 6, 10 の 30 試行で得られた全パレート最適解集合のプロット図を図 17 に示す。図 17 から生成子個体の増加にともない、収束および多様性に優れた幅広いパレート最適解集合が得られていることを視覚的にも確認することができる。

4.3.2 生成子個体数増加に関する考察

4.3.1 項の数値実験により、生成子個体数の増加にともなう解探索能力への影響について調査を行った。生成子個体の増加にともない、1 世代あたりの評価計算回数も増加するため、同じ評価計算回数で比較する場合、十分な世代数を経ることができなくなり、結果として精度が劣る結果となった。一方、世代数を一定にして比較する場合は、生成子個体数を増やすにつれ、幅広い多様性の優れたパレート最適解集合が得られることを確認した。このことから、同じ個体間で交叉回数を増加させ、数多くの生成子個体を生成するにともない、優れた解の発見につながると考えられる。

以上のことから、性能に適應して生成子個体数を変化させることにより、探索性能について精度の高いパレート最適解集合が得られることが期待できる。

4.4 まとめ

本章では、提案並列モデルの有効性を多目的 GA の

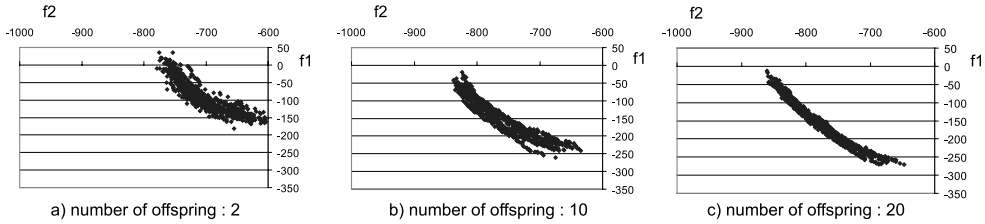


図 16 KUR における生成子個体数が 2, 10, 20 のときに得られたパレート最適解
 Fig. 16 Obtained pareto-optimal solutions with the number of offspring set to 2, 10 or 20 in KUR problem.

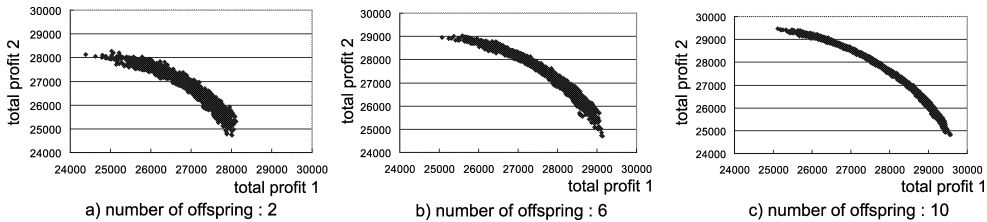


図 17 KP750 - 2 における生成子個体数が 2, 6, 10 のときに得られたパレート最適解
 Fig. 17 Obtained pareto-optimal solutions with the number of offspring set to 2, 6 or 10 in Knapsack problem.

表 2 計算資源
 Table 2 Calculation resources.

	Number of CPU	CPU	Memory
Master Process	1	Athlon64 3200+	1 GB
PC Cluster A	10	Pentium4 2.8 GHz	1 GB
PC Cluster B	15	Xeon 2.4 GHz	1 GB
PC Cluster C	15	PentiumIII 1 GHz	512 MB
PC Cluster D	10	PentiumIII 600 MHz	256 MB

探索性能の観点から示すため、テスト関数を用いた数値実験により代表的な多目的 GA の手法と比較検討を行った。近傍交叉の有効性を確認するため、近傍シャッフル幅の変化による解探索能力への影響を調査した結果、適切な近傍シャッフル幅による近傍シャッフルを行うことにより、多様性の優れた幅広いパレート最適解集合の探索が可能となることを確認した。また、計算資源の性能に適合した生成子個体数の変化の有効性を確認するため、生成子個体数増加による解探索能力への影響を調査した結果、数多くの子個体を生成するにともない精度の優れた解の発見が可能となることを確認した。これらにより、提案並列モデルを使用することにより優れた解探索性能を有することが期待できる。次章では、ヘテロな計算環境を利用した数値実験により、提案並列モデルの有効性を示す。

5. ヘテロ計算環境における数値実験

本章では、ヘテロな計算環境を使用して数値実験を

行い、提案並列モデルの有効性を検討する。代表的な多目的 GA である NSGA-II に対して、提案モデルを使用して並列化を行う場合と、標準の個体の評価のみをマスタスレーブモデルを利用して並列化を行う場合との、探索能力、並列度、およびオーバヘッドの影響について比較検討を行う。以降、比較対象となる標準の並列モデルをオリジナルマスタスレーブモデルと呼ぶ。

5.1 実験環境および実験手順

本実験では 1 台のマスタプロセスと 4 つの異なる性能を持つ PC クラスタを用い、計 50 CPU のスレーブプロセスを利用して提案モデルの有効性を検証する。使用した計算資源を表 2 に示す。グリッドミドルウェアとして Globus Toolkit (version 4.0.1)⁽²²⁾ を使用し、マスタプロセスから各 PC クラスタへのジョブの投入には Grid RPC の 1 つである Ninf-G (version 2.4)⁽²³⁾ を利用する。また、各 PC クラスタ内でのスケジューリングには Open PBS (version 1.2.0 p6)⁽²⁴⁾

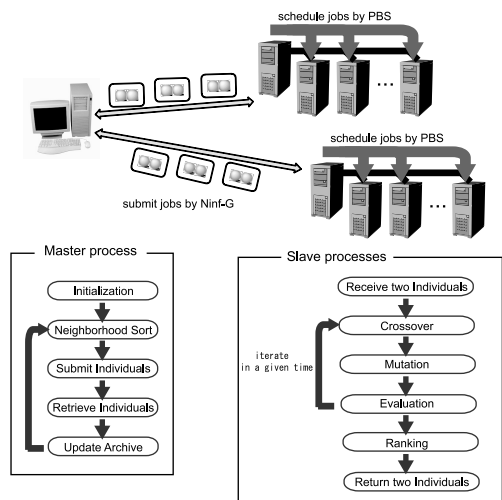


図 18 提案モデルにおけるジョブの実行の流れ

Fig. 18 Execution flow on the proposed model.

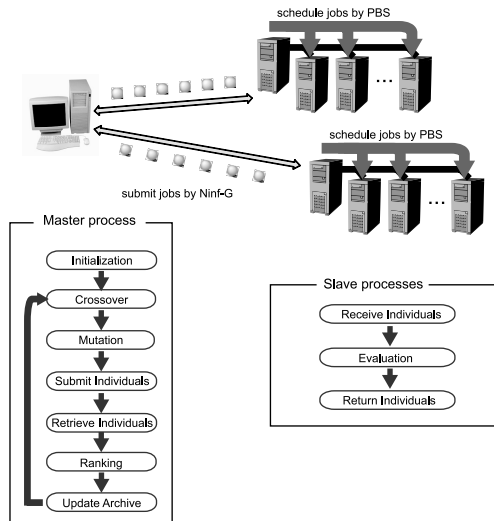


図 19 オリジナルマスタスレーブモデルにおけるジョブの実行の流れ

Fig. 19 Execution flow on the original master-slave model.

を用いた．本実験では各 PC クラスタのマスタノードに PBS のスケジューリングを行うマスタが存在し，各 PC クラスタに投入するジョブの数は分かっているものとする．

5.1.1 提案モデルにおける実験手順

提案モデルのジョブの実行の流れを図 18 に示す．マスタプロセスはまず初期母集団を生成し，目的関数空間において近接している順にソートを行う近傍ソート（近傍シャッフル）を実行する．次にソートされた母集団において，隣り合う 2 個体ずつを各 PC クラスタのマスタノードへ送信する．これは，Ninf-G を利用した非同期通信によるジョブ投入である．この際送信するデータは 2 個体の遺伝子情報である．そしてこれらの情報を受信した各 PC クラスタのマスタノードは，Open PBS によりスケジューリングを行い，スレーブプロセスとなる PC クラスタのスレーブノードにジョブを配布する．各スレーブプロセスは 3.5 節のステップ (1) からステップ (8) までを行い，マスタプロセスに 2 個体を送信する．この際マスタプロセスに送信するデータは 2 個体の遺伝子情報および目的関数値になる．マスタプロセスはすべてのスレーブプロセスのジョブの完了を待ち合わせ，アーカイブ母集団を更新する．なお，本実験では各スレーブプロセスは実行開始から計算資源上に常駐するものとし，マスタプロセスのすべての実行が終了するまで解放されないものとする．

5.1.2 オリジナルマスタスレーブモデルにおける実験手順

比較対象とするオリジナルマスタスレーブモデルに

におけるジョブの実行の流れを図 19 に示す．実験環境は提案モデルと同様に Globus Toolkit , Ninf-G , Open PBS を用いる．マスタプロセスは遺伝的操作を行い，母集団サイズ/スレーブプロセス数の数の個体を各スレーブプロセスに非同期通信により送信する．そして各スレーブプロセスは受信した個体に対して評価を行い，評価値をマスタプロセスに返信する．なお，提案モデルにおける条件と同様に，マスタプロセスはすべてのスレーブプロセスのジョブの完了を待ち合わせ，アーカイブ母集団を更新するものとする．また各スレーブプロセスはマスタプロセスのすべての実行が終了するまで解放されないものとする．

5.1.3 対象問題および実験条件

対象問題は KUR とするが，実最適化問題を想定して標準の評価計算に対してある一定の処理を付加させることにより計算負荷を増大させる．本実験では，評価時間が数秒になるように目的関数の評価計算を 10,000 回繰り返し行うものとした．KUR において計算負荷を増大させたときの各 PC クラスタの計算資源による 1 個体の評価時間は，PC クラスタ A で 5.82 秒，B で 8.62 秒，C で 10.17 秒，D で 17.06 秒であった．このとき提案モデルにおいて，最も性能の劣る PC Cluster D の計算資源が最低 2 個体を生成できるように，1 分間繰り返し交叉，突然変異，評価を行うように設定する．

得られた解集合に対する評価手法および共通パラメータは 4 章における数値実験で使用したものと同

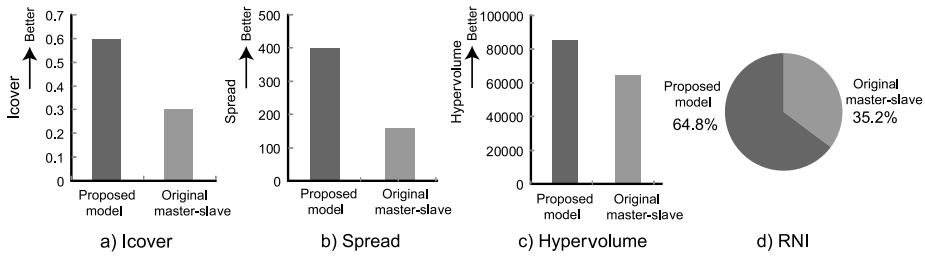


図 20 KUR における提案モデルとオリジナルマスタースレーブモデルの実験結果 (Icover, Spread, Hypervolume, RNI)

Fig. 20 Results of Icover, Spread, Hypervolume, and RNI on proposed parallel model and original master-slave model in KUR problem.

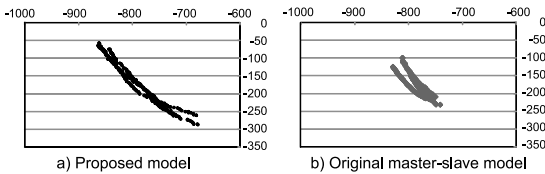


図 21 KUR における提案モデルとオリジナルマスタースレーブモデルによって得られたパレート最適解

Fig. 21 Pareto-optimal solutions obtained by proposed parallel model and original master-slave model in KUR problem.

じであり、提案モデルで使用する近傍シャッフル幅率 $R_{n,sw}$ は 0.1 とする。そして終了条件を実行開始から 120 分とし、120 分経過時に得られたパレート最適解集合の比較検討を行う。

5.2 実験結果

図 20 に実験結果を示す。実験結果は 3 回試行における平均値である。図 20 から、提案並列モデルはオリジナルマスタースレーブモデルと比較してすべての評価手法において優れた結果を得ていることが確認できる。特に Icover, Spread においては約 2 倍の性能を得ることが確認できる。近傍交叉の効果および性能の優れた計算資源による生成子個体数の増加が、解の探索能力に影響を与えたと考えられる。図 21 に 3 回試行で得られたすべてのパレート最適解集合をプロットした図を示す。図 21 から提案モデルにより幅広い多様性の優れたパレート解集合が得られていることが視覚的にも確認できる。

次に並列度の比較を行うため、提案モデルおよびオリジナルマスタースレーブモデルにおける、実行時間 120 分内にスレーブプロセスにより行われた総評価計算回数を表 3 に示す。表 3 から、オリジナルマスタースレーブモデルによる並列化では 23,336 回評価計算が行われたのに対して、提案モデルによる並列化では 40,361 回と約 1.7 倍に並列度向上を可能とした。その結果、解探索性能を向上させることができたと考え

表 3 提案モデルおよびオリジナルマスタースレーブモデルにおける実行時間 120 分内に行われた総評価計算回数

Table 3 Comparative number of evaluation times in proposed model and original master-slave model within 120 min.

	Number of evaluations
Proposed model	40,361
Original master-slave	23,336

表 4 各 PC クラスタ上の計算資源の平均 CPU 使用率

Table 4 Average CPU usage rate of processes on each PC cluster.

	A	B	C	D
Proposed model	88%	91%	95%	91%
Original master-slave	18%	44%	53%	89%

ことができる。

5.3 考察

提案モデルの特徴として、すべての計算資源は性能に適応して動的に負荷を変化させ、一定時間後ほぼ同時に処理を終了するため、すべての計算資源のアイドル時間を最小限にすることができる。このことを確認するため、実行中の各計算資源の負荷状況の調査を行う。表 4 に、提案モデルおよびオリジナルマスタースレーブモデルの各 PC クラスタにおける実行中の平均 CPU 使用率を示す。表 4 から、オリジナルマスタースレーブモデルでは性能の劣る PC クラスタ D の計算資源の平均 CPU 使用率は 89% と負荷が つねに生じているが、性能が優れた PC クラスタほど平均 CPU 使用率が低下しており、アイドル時間が大きくなっていることが分かる。すなわち、性能の劣る計算資源が世代交代を遅延させる原因となり、並列度の低下につながったと考えられる。一方、提案モデルではすべての計算資源において均一に高い負荷が生じており、最大限に使用されていることが確認できる。その結果として並列度の向上および探索性能の向上に貢献することができたといえる。

表 5 提案モデルおよびオリジナルマスタスレーブモデルにおけるオーバーヘッド

Table 5 Overhead time in the proposed model and the original master-slave model.

	Time
Proposed model	4m5s
Original master-slave	4m38s

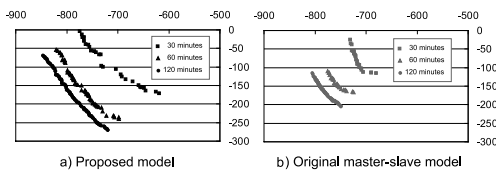


図 22 提案モデルとオリジナルマスタスレーブモデルにおける探索履歴

Fig. 22 Search history of the proposed model and the original master-slave model.

また提案モデルの特徴として、スレーブプロセスでの処理時間を増大させることにより、通信時間やスケジューリングに要する時間などのオーバーヘッドの影響を削減することが可能となる。このことを確認するため、オーバーヘッドの調査を行う。表 5 に提案モデルおよびオリジナルマスタスレーブモデルの実行中に要した総オーバーヘッド時間を示す。表 5 に示す総オーバーヘッドは、全実行時間から最も処理に時間を要したスレーブプロセスが計算を行った時間を引いたものであり、通信時間およびスケジューリングに要した時間を示している。なお、マスタプロセスにおけるアーカイブ母集団を更新する処理は評価計算時間と比較して十分小さいものであるため無視できるものとする。表 5 から、提案モデルはオリジナルマスタスレーブモデルと比較してわずかではあるがオーバーヘッド時間が短いことが分かり、オーバーヘッドの影響を軽減可能であることが確認された。

最後に、提案モデルの探索の特徴に関する考察を行う。提案モデルは、近傍交叉および優れた計算資源における生成子個体数の増加により、優れた探索能力を有している。図 22 に提案モデルとオリジナルマスタスレーブモデルにおける探索過程を示す。図 22 は各並列モデルにおける実行時間が 30 分、60 分、120 分経過した時点で得られた非劣解集合をそれぞれプロットしたものである。オリジナルマスタスレーブモデルと比較して、提案モデルでは、探索の序盤から幅広い多様性の優れた非劣解集合が得られている。つまり、提案モデルでは多様性を維持しながら幅広く探索が可能であり、収束、多様性、幅広さすべての目標を満たすパレート最適解集合を得ることが可能であるとい

える。

これらのことから、提案モデルでは高い並列度を有し、すべての計算資源を最大限に使用可能することで並列度を向上させ、オーバーヘッドの影響も軽減可能であるため、ヘテロ計算環境に適した並列モデルであることが分かった。

6. 結 論

本論文ではヘテロ計算環境に対応した多目的 GA の並列モデルの提案を行い、その有効性の検証を行った。提案モデルでは、高い並列度を有するマスタスレーブモデルを拡張し、評価だけでなく交叉および突然変異も並列の対象とした。そしてヘテロな計算環境を考慮し、計算資源の性能に適応した生成子個体数の変化により、動的な負荷の決定を実現可能とした。また、解探索性能を考慮し、近傍交叉を採用した。つまり、マスタプロセスは母集団を目的関数空間において近接している 2 個体を交叉ペアとして選択し、スレーブプロセスに送信する。本論文の前半では、近傍交叉および生成子個体数増加による解探索性能への影響を調査し、アルゴリズムの有効性を検証した。そして後半では、ヘテロな計算環境を使用した数値実験により提案並列モデルの有効性を検証した。数値実験の結果、提案並列モデルは従来のマスタスレーブモデルと比較して高い並列度および優れた解探索性能を示した。また提案並列モデルではすべての計算資源を最大限に使用可能であること、そしてオーバーヘッドの影響を軽減可能であることを確認した。

以上より、提案並列モデルでは高い並列度および優れた探索性能を有し、すべての計算資源を最大限に使用可能することにより並列度を向上させ、オーバーヘッドの影響も軽減可能であるため、ヘテロ計算環境に適した並列モデルであるといえる。

今後の課題として、交叉ペアを投入する計算資源の決定方法があげられる。現在では、どの交叉ペアをどのスレーブプロセスに送信するかはランダムに決定しており、探索過程で探索を重点的に行う領域を発見できた場合に性能の優れた計算資源を活用することを検討する必要がある。また本論文の数値実験では比較的探索が困難かつ幅広くパレート最適解が分布しているような対象問題を取り扱ったが、今後パレート最適フロントが既知である対象問題についても検討を行い妥当性を示すこと、そして目的関数の数が多い問題についても検討する必要がある。

参 考 文 献

- 1) Goldberg, D.E.: *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesley (1989).
- 2) Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms, *Proc. 1st International Conference on Genetic Algorithms*, pp.93–100, Mahwah, NJ, USA, Lawrence Erlbaum Associates, Inc. (1985).
- 3) Pratab, A., Deb, K., Agrawal, S. and Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, *Proc. Parallel Problem Solving from Nature VI Conference*, Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J. and Schwefel, H.-P. (Eds.), Paris, France, Lecture Notes in Computer Science, No.1917, pp.849–858, Springer (2000).
- 4) Laumanns, M., Zitzler, E. and Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Gloriestrasse 35, CH-8092 Zurich, Switzerland (2001).
- 5) Murata, T. and Ishibuchi, H.: Moga: Multi-objective genetic algorithms, *Proc. 2nd IEEE International Conference on Evolutionary Computing*, pp.289–294 (1995).
- 6) Yoshida, K., Kobayashi, S. and Asada, M.: Generating a set of pareto optimal decision trees by genetic algorithms, *Journal of the Japanese Society for Artificial Intelligence*, Vol.11, No.5, pp.725–732 (1996).
- 7) Deb, K., Zitzler, E. and Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, Vol.8, No.2, pp.173–195 (2000).
- 8) Fonseca, C.M. and Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, *Genetic Algorithms: Proc. 5th International Conference*, pp.416–423, Morgan Kaufmann (1993).
- 9) Nafpliotis, N., Horn, J. and Goldberg, D.E.: A Niche Pareto Genetic Algorithm for Multi-objective Optimization, *Proc. 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Vol.1, Piscataway, New Jersey, pp.82–87, IEEE Service Center (1994).
- 10) Deb, K., Zope, P. and Jain, A.: Distributed computing of pareto-optimal solutions with evolutionary algorithms, *EMO*, pp.534–549 (2003).
- 11) Streichert, F., Ulmer, H. and Zell A.: Parallelization of multi-objective evolutionary algorithms using clustering algorithms, *EMO 2005*, Coello, C.A., Aguirre, A.H. and Zitzler, E. (Eds.), Vol.3410 of LNCS, Guanajuato, Mexico, 9–11 March 2005, pp.92–107 (2005).
- 12) van Veldhuizen, D.A., Zydallis, J.B. and Lamont, G.B.: Considerations in engineering parallel multiobjective evolutionary algorithms, *IEEE Trans. Evolutionary Computation*, Vol.7, No.2, pp.144–173 (2003).
- 13) de Toro Negro, F., Ortega, J., Ros, E., Mota, S., Paechter, B. and Mart, J.M.: Psfga: Parallel processing and evolutionary computation for multiobjective optimisation, *Parallel Comput.*, Vol.30, No.5-6, pp.721–739 (2004).
- 14) Lamont, G.B., Coello, C.A. and van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, Norwell, MA, USA (2002).
- 15) Branke, J., Schmeck, H., Deb, K. and Reddy, M.: Parallelizing multi-objective evolutionary algorithms: Cone separation, *Congress on Evolutionary Computation*, Vol.2, pp.1952–1957, IEEE (June 2004).
- 16) Hiroyasu, T., Watanabe, S. and Miki, M.: Neighborhood cultivation genetic algorithm for multi-objective optimization problems, *Proc. 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL-2002)*, pp.198–202 (2002).
- 17) Hiroyasu, T., Kim, M. and Miki, M.: Spea2+: Improving the performance of the strength pareto evolutionary algorithm2, *Parallel Problem Solving from Nature — PPSN VIII*, pp.742–751 (2004).
- 18) Kursawe, F.: A Variant of Evolution Strategies for Vector Optimization, *Parallel Problem Solving from Nature, 1st Workshop, PPSN I*, Schwefel, H.P. and Männer, R. (Eds.), Vol.496 of Lecture Notes in Computer Science, Berlin, Germany, pp.193–197, Springer-Verlag (1991).
- 19) 比屋根一雄：並列遺伝的アルゴリズムによる多目的最適化問題のバ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).
- 20) Zitzler, E. and Thiele, L.: An evolutionary algorithm for multiobjective optimization: The strength pareto approach, Technical Report 43, Gloriestrasse 35, CH-8092 Zurich, Switzerland (1998).
- 21) Lee, T.H., Tan, K.C. and Khor, E.F.: Incrementing Multi-objective Evolutionary Al-

gorithms: Performance Studies and Comparisons, *1st International Conference on Evolutionary Multi-Criterion Optimization*, pp.111-125 (2001).

- 22) The Globus Alliance. <http://www.globus.org/>.
 23) Tanaka, Y., Nakada, H., Sekiguchi, S., Suzumura, T. and Matsuoka, S.: Ninf-G: A Reference Implementation of RPC-based Programming Middleware for Grid Computing, *Journal of Grid Computing*, Vol.1, No.1, pp.41-51, Kluwer Academic Publishers (2003).
 24) Cluster Resources, TORQUE Resource Manager. <http://www.clusterresources.com/pages/products/torque-resource-manager.php>

(平成 19 年 1 月 29 日受付)

(平成 19 年 3 月 15 日再受付)

(平成 19 年 4 月 17 日採録)



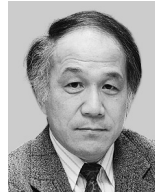
吉井 健吾 (学生会員)

1982 年生。2007 年同志社大学大学院工学研究科博士前期課程修了。最適設計，多目的遺伝的アルゴリズム，並列処理等の研究に従事。同年，ソニー株式会社入社。



廣安 知之 (正会員)

1997 年早稲田大学理工学研究科後期博士課程修了。現在，同志社大学工学部助教授。創発的計算，進化的計算，最適設計，並列処理等の研究に従事。IEEE，電気情報通信学会，計測自動制御学会，日本機械学会，超並列計算研究会，日本計算工学会各会員。E-mail: tomo@is.doshisha.ac.jp.



三木 光範 (正会員)

1950 年生。1978 年大阪市立大学大学院工学研究科博士課程修了，工学博士。大阪市立工業研究所研究員，金沢工業大学助教授を経て 1987 年大阪府立大学工学部航空宇宙工学科助教授，1994 年同志社大学工学部教授。進化的計算手法とその並列化，および知的なシステムの設計に関する研究に従事。著書は『工学問題を解決する適応化・知能化・最適化法』(技法堂出版)等多数。IEEE，米国航空宇宙学会，人工知能学会，日本機械学会，計算工学会，日本航空宇宙学会等会員。通産省産業技術審議会委員等歴任。超並列計算研究会代表。