Transactions of JSCES, Paper No.20040001

離散最適化のための大域的交叉メカニズムを持つ分散遺伝的アルゴリズム*

Global Crossover based Distributed Genetic Algorithm for Discrete Optimization Problems

三木 光範[1], 廣安 知之[1], 勝崎 俊樹[2], 水田 伯典[2]

Mitsunori MIKI, Tomoyuki HIROYASU , Toshiki KATSUZAKI and Takanori MIZUTA

[1] 同志社大学工学部 (〒610-0321 京都府京田辺市多々羅都谷 1-3)

[2] 同志社大学大学院 (〒610-0321 京都府京田辺市多々羅都谷 1-3)

This paper proposes a new method of genetic algorithms (GAs) for discrete optimization problems. For discrete optimization problems, the performance of Distributed GAs (DGAs) are not so good. We propose a new method of increasing the performance of DGAs for discrete optimization problems. The features of the proposed method, Global Crossover based DGA (GCDGA), are multiple crossover operations applied to the elite individuals and DGA without migration. We apply GCDGA to job-shop scheduling problems (JSPs). The experiments on JSPs showed that GCDGA has a better performance than the conventional GAs, and GCDGA provides an efficient distributed scheme in GAs for discrete optimization problems.

Key Words: Genetic Algorithms, Distributed Genetic Algorithms, Discrete Optimization Problems, Job-shop Schedule Problems, Global Crossover

1. はじめに

遺伝的アルゴリズム（Genetic Algorithms: GAs）は生物の遺伝と進化を模倣したアルゴリズムであり，ヒューリスティック法とランダム探索法を有効に組み合わせた手法である[(1)]．GA はその実装が容易であることから，確率的探索，学習，最適化など幅広い分野で応用されている[(2)〜(4)]．

しかしながら，GA における解探索は膨大な反復計算を必要とするため，計算コストが非常に高くなるという問題がある．この問題に対する解決法の一つとして，GA の並列処理があげられる．その手法の 1 つに集団を分割する島モデルの分散 GA（Distributed Genetic Algorithms: DGAs）がある．分散 GA では，集団を複数のサブ集団（島）に分割し，各島ごとに独立に遺伝的操作を行い，一定期間ごとに異なるサブ集団間で個体情報を交換する移住と呼ばれる操作を行う[(5)]．分散 GA は粗粒度の並列モデルに分類され，PC クラスタとの親和性が高い[(6)]．

連続最適化問題において，分散 GA は単一集団 GA（Single Population Genetic Algorithms: SPGAs）と比較して高品質の解が得られると報告されている[(7)]．一方，離散最適化問題においては，単一集団 GA を用いた研究は多いが[(8)〜(10)]，分散 GA を用いた研究は少なく，その性能は明らかとなっていない．そこで，本研究では離散最適化問題の中からジョブショップスケジューリング問題（Job-shop Scheduling Problem: JSP）を取り上げ，分散 GA の性能を検証した．その結果，連続最適化問題とは異なり，離散最適化問題に対しては分散 GA の性能は良好ではないことがわかった．そこで，本研究では，離散最適化問題に対して有効な，分散 GA における新たなサブ集団間での情報交換スキームの提案を行い，提案手法の性能検証および考察を行う．

2. ジョブショップスケジューリング問題に対する分散 GA の性能

ジョブショップスケジューリング問題（JSP）は，各仕事を処理する機械の順序（技術的順序），および，各機械上での各仕事（作業）が処理時間は与えられている状況下で，すべての仕事を処理し終えるまでの総所要時間（Makespan）を最小にするような作業の順序を

* 原稿受付 2003 年 9 月 12 日, 改訂年月日 2003 年 12 月 18 日, 発行年月日 2004 年 1 月 29 日.

Table 1　Performance of TSSB for JSP [11]

Problem	n	m	Opt.(LB UB)	Best
ft10[14]	10	10	930	930
orb1[16]	10	10	1059	1064
ta21[17]	20	20	(1539 1647)	1659
ta31[17]	30	15	(1764 1766)	1771
ta41[17]	30	20	(1859 2023)	2045

決定する問題である．JSP は代表的な離散最適化問題として知られ，GA 以外の最適化手法による研究も行われている．TSSB（A tabu search method guided by shifting bottleneck）[11] のようにタブーサーチなどを用いた解探索を行うことで，Table 1 に示すように大規模な問題に対しても良好な結果が得られることが分かっている．ここで n は仕事数を，m は機械数を示す．また，Best は最良解であり，Opt. における LB，UB は下界値と上界値である．しかし，本研究の目的は並列処理に適した分散 GA の性能向上であるため，分散 GA を改良することで JSP に対してより良好な解探索性能を得ることを考える．

JSP に対する分散 GA の性能を検証するため，単一集団 GA と分散 GA の数値実験における性能比較を行う．実験では，交叉法に Inter-Machine JOX [8] [*1] を，突然変異には Job-Based Shift Change [8] [*2]を用い，アクティブスケジュールを得るため GT 法 [12] [*3]による強制操作 [13] を適用した．世代交代モデルには CCM [9] [*4]を適用し，生成子個体数は 20 とした．GA のパラメータは，集団サイズ 800，交叉率 1.0，突然変異率 0.1 とした．分散 GA における移住パラメータについては，分散 GA の性能に大きく影響を与えるため，移住率を 0.1, 0.2, 0.5 の 3 通り，移住間隔を 2, 5, 10, 20 世代の 4 通りを用い，最良の結果を示したものを実験結果とした．実験は，最適解を得るか，評価計算回数が 100 万回に達した時点で打ち切った．対象問題を Fisher・Thompson より提案された 10 仕事 10 機械の JSP である ft10 問題（最適解の Makespan : 930）[14] として実験を行い，100 回試行した結果の Makespan の平均値（avg）と最適解取得率（success）を Table 2 に示す．表中の DGA n とあるのは，分散 GA におけるサブ集団数が n であることを示し，I の列は最良の結果を得た試行の移住間隔を，R の列は移住率を示している．

[*1]全機械での仕事に基づく順序の継承を考慮した交叉法

[*2]ランダムに 1 つの仕事を選び，全ての機械上でその仕事を左または右に Shift Change する突然変異の方法

[*3]Inter-Machine JOX によって得られる子個体を確実に実行可能状態にするために行う修正操作

[*4]JSP のように交叉を用いても容易に良好な個体を生成することが困難である問題に有効とされる世代交代モデル

Table 2　Performance of SPGA and DGA for JSP(ft10)

Method	avg	success	I	R
SPGA	930.69	0.87	-	-
DGA 4	930.57	0.89	2	0.2
DGA 10	930.67	0.88	5	0.5
DGA 20	930.87	0.86	10	0.5
DGA 40	931.14	0.79	2	0.5
DGA 80	931.88	0.66	2	0.5

サブ集団数が 4 のものを除いては，移住率 0.5 の試行が最良の結果を示した．また移住間隔についても，サブ集団数 4, 40 および 80 のものにおいて 2 世代が最良の結果を得ていることから，短い方が性能が高くなる傾向にある．このことから，移住による情報交換を頻繁に行えば分散 GA を用いて比較的良好な性能が得られることがわかる．

しかしながら，分散 GA の性能は単一集団 GA の性能と比較して大きな差はなく，サブ集団数 20 以上の場合には単一集団 GA よりも性能が低くなっている．このことは，移住を行うことによって集団全体の多様性を維持し，探索を効果的に行うことができても，適切な情報交換を行い性能を向上させることが十分できていないことを示しているといえる．

連続最適化問題の場合にはサブ集団数を多くすることで性能が向上すると報告されている [7]．これは各サブ集団において成長する部分解（ビルディングブロック）が移住によって組み合わせられ，効果的に最適解を得ることができたためである．しかし，JSP では移住が効果的に機能しなかった．この点が離散最適化問題と連続最適化問題の相違であると考えられる．

3. 新たなサブ集団間情報交換スキーム

3.1　分散 GA の問題点　前節の実験結果より，離散最適化問題における分散 GA の問題点は，移住を行っても適切な情報交換がなされにくい点にあると考えられる．離散最適化問題は連続最適化問題とは異なり，対象問題に特化した染色体のコーディング法や交叉法を用いることが多いため，部分解が他の個体に受け継がれにくい．また，対象問題によっては，大きな部分解が存在しないことも考えられる．

移住による情報交換は交叉を実行することで実現する．すなわち，移住先のサブ集団内で移住個体が交叉に組み込まれることで，そのサブ集団内の個体との情報交換がなされる．しかし，移住によりサブ集団間での個体の移動が行われた際に，あるサブ集団から良好な個体が移住したとしても，移住先のサブ集団でそれ

が有効利用される（ここでは適切な交叉が行われることを指す）保証はない．また，解探索速度が低下する探索中盤以降においては，大域的探索よりも局所的探索が必要となることが多いため，性質が大きく異なるような移住個体は有効利用されることが少なくなる．そのため，各サブ集団における有力な情報を持った個体が他のサブ集団に移住しても，その個体に近い性質の個体との交叉が行われなければ，より良い個体が生成される可能性は低い．

これらのことから，離散最適化問題においては移住による情報交換だけでは，分散 GA の性能が十分に引き出されず，各サブ集団間の情報交換においては，適切な交叉対象個体のペアを作成することが重要であると考えられる．探索時間を十分に長くすれば，適切な交叉のペアが生成されにくいという問題は解決される可能性がある．しかし，限られた時間内で良好な解を得るためには，適切な交叉のペアを自動的に生成するような操作が不可欠である．

3.2　大域的交叉型分散遺伝的アルゴリズムの提案

本節では前節で示した分散 GA における問題点を解消し，適切な交叉のペアを生成する新たな手法，大域的交叉型分散 GA（Global Crossover based Distributed Genetic Algorithm: GCDGA）の提案を行う．

GCDGA は，各サブ集団における有力な情報を持つエリート個体による情報交換を中心とした探索を行う．具体的には，各サブ集団からエリート個体を含む数個の個体を抽出し，それらの個体による交叉を連続して行う．この操作を，大域的交叉（Global Crossover: GC）と呼ぶ．GC を適用することでサブ集団間の情報交換が行われるため，GCDGA では移住を行わない．Fig.1 に GCDGA の操作を示す．

GCDGA では移住に代わって GC を行う．すなわち，GC 適用までは分散 GA と同じく各サブ集団内で独立して GA を行い，GC 間隔ごとに GC を実行する．GC は (1)GC 個体群の決定，(2) 交叉対象個体の選択，(3) 交叉対象個体による多段交叉，(4)GC 終了判定の 4 つのフェーズからなる．GC 操作の詳細を次節で説明する．

GC は移住に置き換えられる操作であり，移住のように適用間隔（GC 間隔）ごとに実行される．GC 間隔に達した時点で，(1)〜(4) の 4 つのフェーズを実行する．

各操作の詳細を以下で説明する．

1. **GC 個体群の決定**

   GC において交叉を適用する個体候補である GC 個体群を，各サブ集団から選択する．GC 個体群は，各サブ集団から半数の個体をランダムに選択した個体からなる．ただし，その中には必ず各サブ集団のエリート個体が含まれるものとする．この個体群選択により，GC を実行することによる影響範囲は，最大でも各サブ集団内の半数の個体となる．

2. **交叉対象個体の選択**

   GC 個体群から交叉対象となる個体を選択する．これは，次の多段交叉を行うための親個体候補の選択である．交叉対象個体には，各サブ集団の GC 個体群から 2 個体ずつ選択する．この 2 個体の中には高い確率でサブ集団内のエリート個体が選択されるようにし，多段交叉による情報交換の効率を高める．

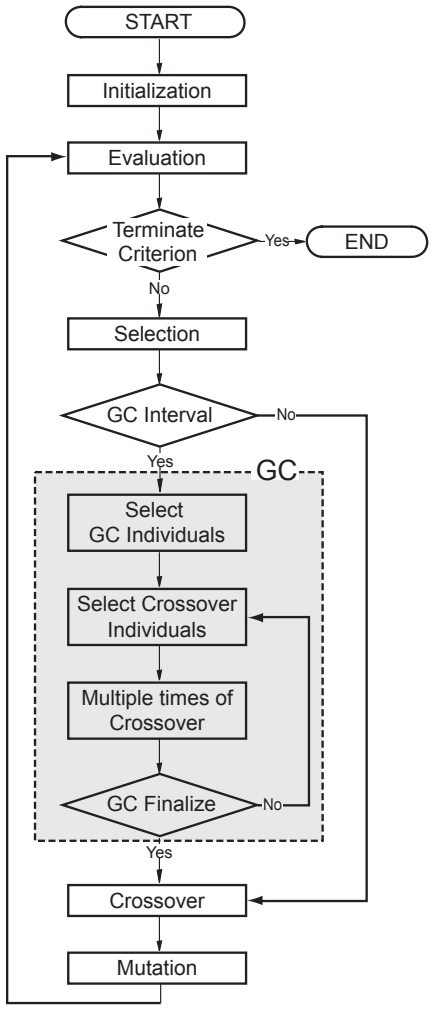

Fig.1　Flow of GCDGA

3. **交叉対象個体による多段交叉**

   ここで，選択された個体に対して交叉を行い，各サブ集団間で個体の情報交換をする．この交叉において親個体として選択される 2 個体は，必ず異なるサブ集団からの個体とする．親個体のペアは，全ての交叉対象個体を用いて非復元抽出により作成する．

   親個体のペアを決定したら，それぞれのペアに対して交叉を n 回行い子個体を $2n$ 個生成する．つまり，全体で $2n \times$ (ペア数) の子個体を生成する．次に親個体と子個体での生存選択を行う．生存選択では，交叉によって生成された子個体のうち最良のものと親個体を系列ごとに比較を行い，最良の個体を 1 個体ずつ選択し，次の交叉対象個体とする．ここでいう系列とは，交叉に用い

る親個体2個体それぞれにおいて，片方の親個体の特徴をより濃く受け継ぐ子個体にその親個体自身を加えたものである．

この交叉終了後，得られた子個体はすぐにGC個体群に戻さずに，子個体同士で再度非復元抽出によりペアを作り，同様の交叉を指定回数連続して行う．この連続した交叉を多段交叉と呼ぶ．

4. **GC終了判定**
多段交叉終了後，あらかじめ設定されたGCの終了条件を満たしていない場合には，交叉対象個体を再度選択し，多段交叉を繰り返す．GCの終了条件を満たしたら，GC個体群を分割し，サブ集団に戻す．その後，再び各サブ集団での探索に戻る．

3.3 GCDGAの特徴 GCDGAの特徴をまとめると次のようになる．

- 特定個体への連続した複数回の交叉
GCDGAはGC個体群から交叉対象個体を選択し，それらの個体を用いて交叉を連続して行うため，交叉対象個体の高速な進化が期待できる．
- エリート個体重視
GC個体群には必ず各サブ集団のエリート個体が含まれ，交叉対象個体群にエリート個体が含まれやすいため，エリート個体による探索が重点的に行われる．これにより，各サブ集団からの有力な個体情報が交換され，多段交叉での効果的な探索が期待できる．
- 移住を行わない
GCDGAは移住を行わないため，GC中の交叉対象個体以外は各サブ集団において独自に進化することになる．そのため，探索の終盤まで多様性を維持することが可能になると期待できる．
- 実行環境への柔軟な応用性
GCDGAは交叉法やGAのモデルに依存することなく適用可能である．基本的に，分散GAに適用できるものであれば採用できるため，既存の環境に対する応用が容易である．

一方，GCDGAを適用するためには次の点を考慮する必要があると考えられる．

- GCDGA独自のパラメータ設定が必要
GCDGAには独自のパラメータが存在する．これらの設定がGCDGAの性能に大きく影響を与えるため，性能を引き出すためには予備実験が必要となる．
- 交叉法に対する依存性
GCDGAにはどのような交叉法も適用可能である．しかし，移住を行わずに交叉対象個体による交叉を連続して行うことで情報交換を行っているため，親の形質を大きく破壊するような交叉法を適用すると，情報交換が適切に行われないためGCが有効に機能しない．
- サブ集団数の設定
GCDGAは各サブ集団から個体を集め，連続して交叉を行うことで移住よりも高い効果を得ようとする手法であるため，サブ集団数を多くして多様な個体を集めなければ高い性能を得にくい．

これらのことから，GCDGAを適用する際には用いる交叉法とサブ集団数の設定を十分に考慮しなければならないといえる．なお，最適なサブ集団数については5.2節で検討する．

4. 提案手法の性能

前節で提案したGCDGAの性能を単一集団GAおよび分散GAと比較することで検証する．この際，問題としては代表的なJSPとして，2章で用いたft10問題，Lawrenceより提案されたla19問題[15]，および，Applegate･Cookより提案されたorb1問題[16]を用いた．

4.1 ft10問題に対する性能 前節で提案したGCDGAの性能を単一集団GAおよび分散GAと比較することで検証する．数値実験で用いたGCDGAの設定は次のようである．GCは10世代目から5世代ごとに適用し，エリート個体選択率は0.5，GC中の交叉では子個体を1回あたり20個生成，多段交叉中の交叉回数は2回とした．また，GCの終了はGC中の評価計算回数が8万に達した時点とした．

なお，GCの開始世代と適用世代間隔は分散GAの移住同様，問題依存が大きいと考えられるため予備実験により決定した．また，多段交叉における親個体の選択については，各サブ集団のエリート個体と他の個体との交叉を目指すため，エリート個体選択率は0.5とした．GC中の交叉での1回あたりの子個体生成数は，どの程度子個体を生めば良好な解を得られるかを考察した結果，20個体に決定した．選択できる多段交叉中の交叉回数およびGC中の評価計算回数は，どの程度の評価計算を行えばGCによる十分な情報交換が行えるのかについて検討した結果，適切と思われる値を用いた．すなわち，GCに関係する個体数はこの場合400であり，200ペア作成されることになる．各ペアが20個体ずつの子個体を生成し，その交叉を2世代分実施し，この全体のプロセスを親個体のペアを変化させて10回繰り返す．これがGC中の評価計算回数が8万回である根拠となる．ここで用いた値は予備実験で得られた値であるが，多くの対象問題に対して普遍性の高い値であると考えている．

なお，GCDGAにおいても実験の打ち切り評価計算回数は100万回としているため，単一集団GAおよび分散GAと同じ評価計算回数で実験を終了する．その他のパラメータおよび環境は2節で用いたものと同じとした．対象問題をft10問題とした実験結果をTable

Table 3　Performance of GCDGA and conventional GAs

	GCDGA		Conventional	
Method	avg	success	avg	success
SPGA	-		930.69	0.87
DGA 4	930.83	0.86	930.57	0.89
DGA 10	930.85	0.86	930.67	0.88
DGA 20	930.53	0.90	930.87	0.86
DGA 40	930.43	0.93	931.14	0.79
DGA 80	930.27	0.95	931.88	0.66

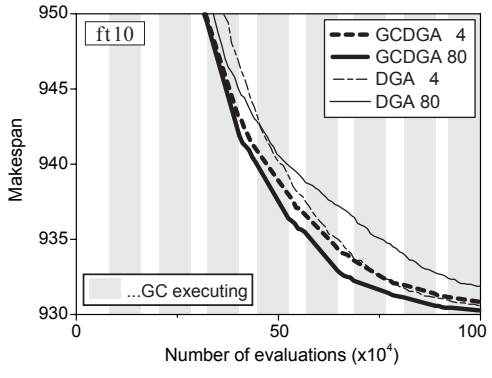

Fig.2　History of Makespan on GCDGA and DGA

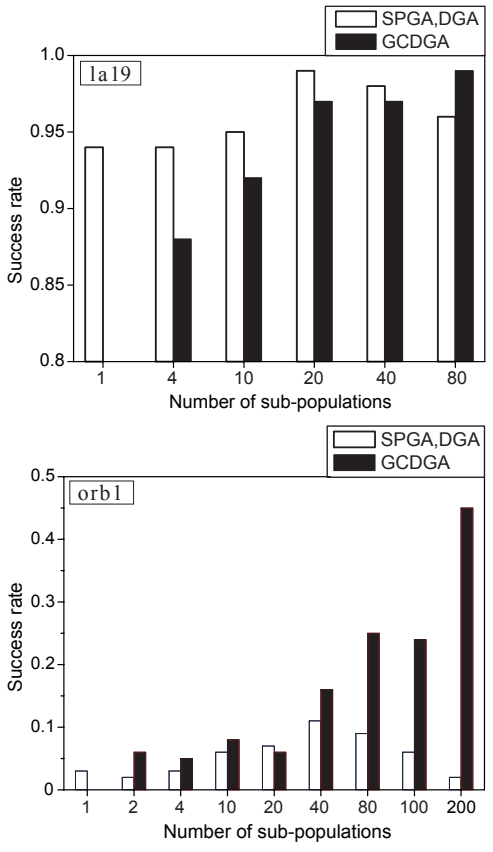

Fig.3　Success Ratio on GCDGA and DGA

3 に示す．なお，表において，GCDGA の列は GCDGA の性能を，Conventional の列は 2 節で示した通常の GA の性能を示している．

GCDGA の性能は，サブ集団数が 20 以上のとき分散 GA および単一集団 GA の性能を上回っている．また，サブ集団数を多くすることで性能が向上しており，サブ集団数を 80 とした場合に最高の性能を示している．一方，サブ集団数が 10 以下の場合には通常の分散 GA よりも性能が悪い．この原因は 3.3 節で述べたように，サブ集団数が少ないと交叉対象個体が少なくなるため，情報交換の効率が低下するためである．

この結果から，サブ集団数を多くした上で GCDGA を適用すれば，個体を各サブ集団に分散することによる効果が有効に活用され，解の成長が効果的に行われると考えられる．このことを検証するため，上記の実験における解成長の履歴について検討した．Fig.2 に Makespan の 100 試行平均の履歴を示す．なお，図における GCDGA n とあるのは，GCDGA におけるサブ集団数が n であることを示す．

図中の背景が網がけの部分は GC 実行中の箇所を示す．GCDGA は探索の中盤においてすでに分散 GA よりも高い性能を示している．一方，GCDGA においてサブ集団数 4 と 80 の探索結果を比較すると，3 度目の GC を適用するまではほぼ同じ性能を示しているが，それ以降の GC による効果が異なり，サブ集団数を多くした方が GC による解の改善効果が大きい．

GCDGA は分散 GA よりも高い性能を示すが，サブ集団数が少ない場合には探索の後半における解の改善能力が低くなってしまい，最終的に分散 GA よりも性能が低くなっている．一方で，サブ集団数を多くすれば GC 適用による探索後半の解の改善が大きいことから，より高い性能が得られることがわかる．

これらのことから，GCDGA はサブ集団数が多い場合，探索前半においては移住を行わなくても GC によって同等以上の性能が得られ，探索中盤以降においては GC の実行によって適切な個体による交叉が行われるため，分散 GA よりも高い性能が得られることがわかる．

4.2　他の問題に対する性能　GCDGA の性能をさらに検討するため，la19 問題（10 仕事 10 機械問題，最適解の Makespan: 842）と orb1 問題（10 仕事 10 機械問題，最適解の Makespan: 1059）に対し，同じパラメータを用いて実験を行った．それぞれの問題における最適解取得率（Success rate）を Fig.3 に示す．

la19 問題に対する結果を見ると，GCDGA の性能は分散 GA よりもやや低い．サブ集団数を 80 とすることでサブ集団 20 の分散 GA と同等の性能を得られているが，それ以上の性能は得られなかった．一方，orb1 問題に対する性能は GCDGA による探索の性能が非常に高くなっている．サブ集団数を増やすごとに性能が向上しており，200 の時には最高の性能を示している．

Table 4　Performance of GCDGA and conventional GAs

Prob.	Opt.	GCDGA	SPGA[8]
ft10	930	930.0	931.9
la11	1222	1222.0	1222.0
la12	1039	1039.0	1039.0
la13	1150	1150.0	1150.0
la14	1292	1292.0	1292.0
la15	1207	1207.0	1207.0
la16	945	945.7	946.0
la17	784	784.0	784.0
la18	848	848.0	848.0
la19	842	842.1	845.9
la20	902	906.1	906.5

このことから，orb1 問題は GC による情報交換の効果を得やすい問題であるといえる．

これらから，GCDGA の性能は対象問題によって異なり，非常に効果的な探索を行うことができる問題と，分散 GA の性能と同等である問題があることがわかる．

4.3　従来手法との比較　提案手法の性能を既存の手法と比較することで検証する．ここでは，本論文で採用した交叉法，Inter-Machine JOX による性能を示した文献 [8] との性能比較を行う．比較の条件を同じとするため，パラメータは，個体数 600，交叉率 1.0，突然変異率 0.1，評価計算回数 300 万回とした．ft10 問題, Lawrence より提案された JSP である la11〜20 問題 [15] に対する GCDGA を用いた数値実験結果と従来手法（SPGA）の実験データ [8] を Table 4 に示す．結果は 100 試行の Makespan の平均値を示している．

従来手法において，la11〜15, 17, 18 問題においてはすべての試行で最適解が得られている．GCDGA でもこれは同様であり，同等の性能を得ている．また，他の問題に対しては，従来手法ではすべての試行で最適解を得るには至っていないが，GCDGA においては，ft10 問題においてすべての試行で最適解を得ており，他の問題でも，従来手法と比較して良好な性能を得ている．

また，最適解を得ることが難しいことで知られる大規模な JSP である Thailand より提案された ta21 問題 [17]，Yamada･Nakano より提案された yn1〜2 問題 [18]，Storer らより提案された swv20 問題 [19] に対する Inter-Machine JOX を用いた単一集団 GA との比較を上述のパラメータのもとで行う．

ta21 問題，yn1〜2 問題，swv20 問題に対する GCDGA と従来手法（SPGA）を用いた数値実験結果を Table 5 に示す．なお，問題の規模は jobs，machines として示す．結果は 20 試行の Makespan の平均値を示している．

Table 5　Performance of GCDGA and conventional GAs for large-scale JSP

Prob.	jobs	machines	GCDGA	SPGA
ta21	20	20	1737.8	1737.6
yn1	20	20	917.2	920.95
yn2	20	20	947.0	948.65
swv20	50	10	2823	2823

Table 5 より，大規模な JSP として採用した 4 問題に対し，ta21 問題のように GCDGA と比較して従来の手法の方がわずかに良好な結果を示す場合もあるが，yn1〜2 問題，swv20 問題の結果より，提案手法は従来手法と比較して良好，もしくはほぼ同等の性能を持つと考えられる．

一方，従来の手法では世代交代モデルとして MGG を用いているのに対し，GCDGA は分割集団を用いた手法であるため並列化効率に優れている．そのため，PC クラスタなど並列処理環境での利用価値は高いと考えられる．

5. 考察

GCDGA による性能向上のメカニズムを考察するため，JSP における解探索の進捗を表す新たな指標の導入を行い，GCDGA における探索が分散 GA よりも効果的に行われていることを示す．

5.1　最適クリティカルペア　JSP では，最適解が複数個存在するため，部分解の推移のような指標を用いて探索の進捗を観察することができない．そこで，本節では JSP における部分解の数を示す新たな指標を導入する．

ft10 問題に対して数値実験を行い，最適解について詳細に検討したところ，70 種類以上の最適解があることがわかった．また，この得られた最適解を解析したところ，クリティカルパスについてはほとんど一致していることがわかった．また，orb1 問題に対しても，同様の傾向が見られた．よって探索中の個体におけるクリティカルパスが最適解のものと共通しているかどうかを調べることで，探索中の個体の部分解の数を知ることが可能となる．

クリティカルパスは単独で存在しているわけではなく，クリティカルパスの前後の作業との関連が強い．そこで，解に含まれているクリティカルパスの位置を調べるのではなく，クリティカルパスの連続が解の中に含まれているかどうかを検討することとした．これによって，どの程度の部分解が探索中のスケジュールに含まれているかを把握することができる．

ここでは，同一機械上でのクリティカルパス上作業が

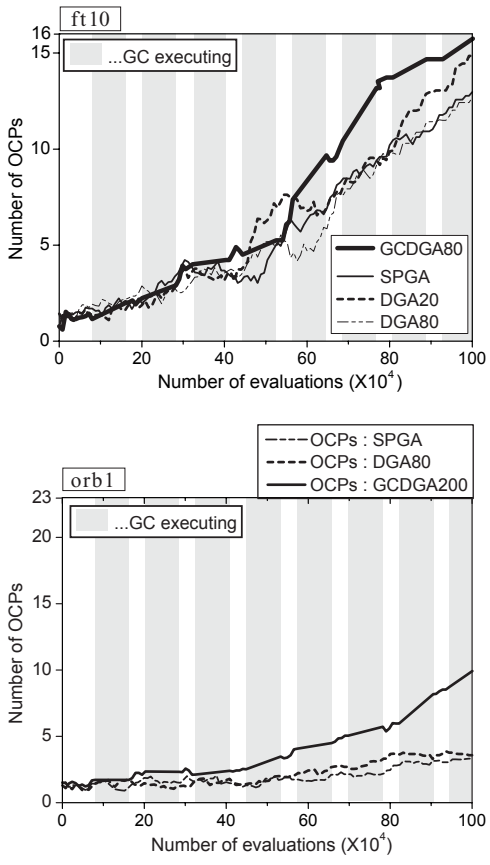

Fig. 4 History of OCPs on SPGA, DGA and GCDGA

2つ連続しているもの，および異なる機械間のクリティカルパス上作業が時間軸上で2つ連続しているものを最適解の構成要素，クリティカルパスペア（Critical path Pair: CP）と呼び，最適解中の CP を OCP（Optimum CP）と呼ぶ．OCP が探索中のスケジュールにいくつ含まれているかを調べることで，最適解への解探索がどのように進んでいるかを知る指標となる．本論文では，あるスケジュール中の CP が OCP と一致している数を OCP 数と呼ぶ．最適解を得たときに OCP 数は最大となる．次節で OCP 数の推移について示し，GCDGA による探索の効果を検討する．

5.2 OCP数の推移 OCP数の推移より，GCDGA の性能向上について検討する．対象問題として，ft10 問題と orb1 問題を扱った．なお，最適解における OCP 数は ft10 問題が 16，orb1 問題が 23 である．

GCDGA の探索性能を検証するため，単一集団 GA および分散 GA との比較を行った．Fig.4 に 4 節で行った実験における最良解中の OCP 数の推移を示す．Fig.4 上は ft10 問題，下は orb1 問題に対する実験結果である．なお，図中の背景が網がけの部分は GC 実行中であることを示す．

まず，ft10 問題の結果について考察する．GCDGA における OCP 数は，探索の中盤までは単一集団 GA や分散 GA と大きな差はない．しかしながら，5 回目の GC によって OCP 数が大きく上昇し，比較的 OCP 数の多いサブ集団数 20 の分散 GA と比較しても大きな差がある．以降の GC によってさらに OCP 数が上昇し，これは探索の終盤まで続いている．

Fig.2 より，適合度においては探索の序盤から GCDGA は分散 GA よりも性能が高いといえた．一方，OCP 数による部分解の構築という点から見ると，GCDGA は探索の中盤以降に分散 GA よりも大きく増加している．移住に変わって GC を行うことで，探索序盤では比較的良好な個体の成長を早める一方，探索の中盤以降は各島における部分解が GC によって組み合わされていき，最適解を構築してゆく．このことから，GCDGA による探索により高い性能が得られていると考えられる．

次に orb1 問題の結果について検討する．GCDGA では探索序盤から OCP 数が分散 GA よりも多い．また，探索後半における OCP 数は，GCDGA においては増加を続けているものの，単一集団 GA や分散 GA においては停滞している．OCP 数が停滞することは，部分解の成長がなくなっていることから局所解への収束を意味していると考えられる．よって，orb1 問題では GC によって各島の有力個体を使って交叉させ探索することが有効に機能しているといえる．

次に，GCDGA による探索がサブ集団数によってどのように影響するかを検討する．ft10 問題および orb1 問題におけるサブ集団数ごとの OCP 数の推移を Fig.5 に示す．Fig.5 上は ft10 問題に対してサブ集団数 10 および 80 として実験を行ったもの，下は orb1 問題を対象としてサブ集団数を 20, 80 および 200 として実験を行ったものである．図における Best は探索中の最良解の OCP 数を，Elites はすべてのサブ集団におけるエリート個体の集合に含まれていた OCP 数を示す．

2 つの問題に対する実験結果において，最良解における OCP 数の推移についてはほぼ共通している．探索の序盤ではサブ集団数による影響がほとんどなく，探索の中盤以降から GC 適用による効果に差が生じるという点である．2 つの問題ともに，サブ集団数を多くした方が多い OCP を得られている．また，各サブ集団中の全エリート個体を集めるとサブ集団数が多い方がより多い OCP を含んでいる点についても共通している．この原因の一つは，サブ集団数が増えることでエリート個体数が増えたためであるが，サブ集団数を増やし集団全体の多様性を高めたことによって全エリート個体集合により多くの OCP が存在するようになったともいえる．

GC によって，各サブ集団中のエリート個体を中心にして探索が進められるため，各エリート個体中に多くの部分解が存在していることが望ましい．ft10 問題は比較的簡単な問題であるため，探索を進めることで全エリート個体集合の OCP 数が増加している．しかし，orb1 問題では探索を続けることで全エリート個体集合の OCP 数が減少している．これは，大きな局所解が存在しているために，探索初期には存在していた小さな部分解が破壊され，集団中に存在しなくなるため

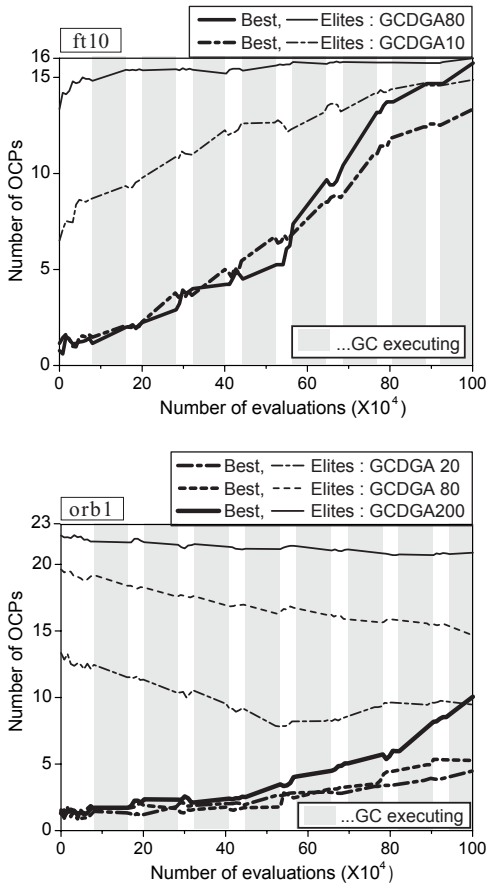

Fig.5　History of OCPs on GCDGA

だと考えられる．

分散 GA においてはサブ集団数を増やし多様性を向上させることでこの問題を解決し，探索の後半まで集団全体の OCP 数を維持することができる．さらに GCDGA では移住を行わないため，GC による集団全体からの個体を用いた交叉を行うことで，部分解を組み合わせて良好な個体を生成できる．このことにより，GCDGA による探索においてサブ集団数を増やすことが探索性能向上につながっていると考えられる．

以上の考察と実験結果から，GCDGA では分散 GA に用いるサブ集団数よりも多くした方が性能が向上するといえる．集団サイズにもよるが，サブ集団内の個体数を 5〜10 として，サブ集団数を増やした方が性能が向上すると考えられる．

6. おわりに

本研究では，離散最適化問題に対する分散 GA における新たな手法，大域的交叉型分散 GA（GCDGA）の提案とジョブショップスケジューリング問題での性能評価および考察を行った．数値実験の結果，GCDGA の性能について次の点が明らかとなった．

- 従来の分散 GA よりも高い性能を示す．
- サブ集団数を増やすことで性能が向上する．
- 多くの部分解を GC に採用することで探索性能が向上する．

今後の課題は，他の離散最適化問題に対する GCDGA の性能を検証することである．なお，本研究は文科省からの補助を受けた同志社大学学術フロンティア研究プロジェクトにおける研究の一環として行った．ここに記して謝意を表す．

参考文献

(1) Holland, J. H.: *Adaptation In natural and Artificial Systems*, University of Michigan Press (1975).

(2) Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wasley Publishing Company (1989).

(3) 北野弘明: 遺伝的アルゴリズム, 産業図書 (1993).

(4) 坂和正敏, 田中雅博: 遺伝的アルゴリズム, 朝倉書店 (1995).

(5) Tanese, R.: Distributed Genetic Algorithms, *Proc.3rd International Conference on Genetic Algorithms*, pp. 434–439 (1989).

(6) Cantú-Paz, E.: A survey of parallel genetic algorithms, *Calculateurs Paralleles*, Vol. 10, No. 2 (1998).

(7) 三木光範, 廣安知之, 畠中一幸, 吉田純一: 並列分散 GA による計算時間の短縮と解の高品質化, 日本計算工学会論文集 (2000).

(8) 小野功, 小林重信: Inter-machine JOX に基づく JSP の進化的解法, 人工知能学会誌, Vol. 13, No. 5, pp. 780–790 (1998).

(9) Ono, I., Nagata, Y. and Kobayashi, S.: A Genetic Algorithm Taking Account of Characteristics Preservation for Job Shop Scheduling Problems, *Proc. of the International Conference on Intelligent Autonomous Systems 5*, pp. 711–718 (1998).

(10) Sakuma, J. and Kobayashi, S.: Extrapolation-Directed Crossover for Job-shop Scheduling Problems: Complementary Combination with JOX, *GECCO 2000*, pp. 973–980 (2000).

(11) Merelli, E. and Pezzella, F.: A tabu search method guided by a shifting bottleneck for a job shop scheduling, *European Journal of Operational Research*, Vol. 120, pp. 297–310 (2000).

(12) Giffler, B. and Thompson, G.: Algorithms for solving production scheduling problems, *Operations Research*, Vol. 8, pp. 487–503 (1960).

(13) Kobayashi, S., Ono, I. and Yamamura, M.: An Efficient Genetic Algorithm for Job Shop Scheduling Problems, *Proc. of 6th International Conference on Genetic Algorithms*, pp. 506–511 (1995).

(14) Fisher, H. and Thompson, G.: Probabilistic learning combinations of local job-shop scheduling rules, *Industrial Scheduling* (eds. by Muth, J.F. and Thompson, G.L.), pp. 225–251 (1963).

(15) Lawrence, S.: Resource constrained project scheduling, *an experimental investigation of heuristic scheduling techniques* (1984).

(16) Applegate, D. and Cook, W.: A computational study of the job-shop scheduling instance, *ORSA Journal on Computing*, Vol. 3, pp. 149–156 (1991).

(17) Thailand, E.: Benchmarks for basic scheduling problems, *European Journal of Operation Research*, Vol. 64, pp.278–285 (1993).

(18) Yamada, T. and Nakano, R.: A genetic algorithm applicable to large-scale job-shop instances, Manner, R., Manderick, B. (eds.), *Parallel instance solving from nature 2* , North-Holland, Amsterdam, pp. 281–290.

(19) Storer, R.H., Wu, S.D. and Vaccari, R.: New search spaces for sequencing instances with application to job shop scheduling, *Management Science* ,Vol. 38, pp. 1495–1509 (1992).