

エージェント指向グリッドコンピューティング (MAO-Grid) の提案と DS-GA の適用によるロードバランシングの最適化

中山 功一[†] 松井 博和^{††}
下原 勝憲^{†,†††} 片井 修^{†††}

本論文では、ネットワークに接続された多数のパーソナル・コンピュータ (PC) が、互いの余剰計算機資源を相互利用する分散処理システムの設計モデルの 1 つとして、分割されたタスクを自律的なエージェントとするエージェント指向グリッドコンピューティング (MAO-Grid) を提案する。また、MAO-Grid においてタスク処理量を最大化するロードバランシングを獲得するエージェント学習手法として、マルチエージェントシステムにおいて有効な動的離隔型 GA (DS-GA) を用いる。MAO-Grid に DS-GA を適用した実験から、計算機資源やネットワーク資源が動的に変化する環境において、システム全体のタスク処理量を最大化にするロードバランシングを獲得した。

Multi-agent Oriented Grid Computing (MAO-Grid) and Optimization of Its Load Balancing by DS-GA

KOICHI NAKAYAMA,[†] HIROKAZU MATSUI,^{††}
KATSUNORI SHIMOHARA^{†,†††} and OSAMU KATAI^{†††}

In this paper, we propose the Multi-agent Oriented Grid Computing (MAO-Grid) approach to efficiently use the surplus resources of personal computers (PCs) connected to a network. Dynamically Separating GA (DS-GA), one of the effective learning algorithms for multi-agent systems, is used as the optimization algorithm for MAO-Grid. We conducted experiments applying the DS-GA to MAO-Grid. The results showed that load balancing was optimized according to the characteristics of computational resources in dynamic environments.

1. はじめに

ここ 10 年ほどで、パーソナル・コンピュータ (PC) は会社や家庭などに広く大量に普及した。また、PC のさらなる高速化、すなわち、計算能力の拡大の流れはとどまるところを知らない。ところが、多くの会社や家庭では、これら PC の計算機資源のすべてがつねに使用されているわけではない。このため、これらの計算機資源の有効利用を目指すグリッドコンピューティング (Grid Computing, 以下 Grid と記す)^{1),2)} やその応用^{3),4)} が研究されている。一般的に、Grid とは、ローカルなネットワークを用いる PC クラスタ⁵⁾ とは異なり、インターネットなどを用いて物理的

に離れた計算機資源を利用する分散処理システムを意味する。タスク処理の依頼者と計算機資源の供給者の区分が不変であるマスタ・ワーカ型 (またはマスタ・スレーブ型) と、可変である相互協力型に大別できる。SETI⁶⁾ などのすでに実用化されている大規模な Grid の多くは、タスクの依頼者がすべてのタスクを集中管理し、供給者の計算機資源を使用するマスタ・ワーカ型である。これに対し近年、一般のユーザが、ネットワークに接続された多数の PC の計算機資源を互いに利用する相互協力型の Grid が注目されている。

相互協力型 Grid の実現には、セキュリティやスケラビリティ、容易な運用性やタスク分割方法など、考慮すべき様々な問題がある。本論文では、これらの問題のうち、従来の集中管理型 Grid における解決法が、相互協力型 Grid にはそのままでは適用できない問題の 1 つとして、タスク処理のスケジューリング、すなわち、ロードバランシングの問題に注目する。

複数のコンピュータが協力して処理する Grid では、どのタスクをどのコンピュータに分配し処理するかと

[†] ATR ネットワーク情報学研究所

ATR Network Informatics Laboratories

^{††} 三重大学工学部

Faculty of Engineering, Mie University

^{†††} 京都大学大学院情報学研究科

Graduate School of Informatics, Kyoto University

いうロードバランシングの適切な決定が重要である。従来の PC クラスタでは、使用可能な計算機資源やネットワークに関する情報が既知で変化しない場合が多く、ロードバランシングの集中管理が可能である。一方、インターネットなどを用いた大規模な Grid では、これらの情報が未知で動的に変化する場合があるため、適切なロードバランシングの事前決定が困難である。さらに、相互協力型 Grid では、システム全体を把握したロードバランシングの集中管理が必ずしも容易ではない。そのため、相互協力型 Grid では、ロードバランシングの自律適応的な決定手法が有効である。また、一般のユーザを対象とするシステムでは、個々のユーザに多くの負担を強いることのないロードバランシングの決定手法も重要である。

本論文では、これらを考慮した Grid の設計モデルとして、自律したエージェントによるロードバランシングの適応的な制御、すなわちエージェント指向アプローチ^{7),8)}に注目し、分散処理される個々のタスクとその処理プログラムを1つのタスク処理エージェントとして設計するエージェント指向グリッドコンピューティング (Multi-agent Oriented Grid Computing, 以下 MAO-Grid と記す) を提案する。

MAO-Grid は、PC 間を移動できる多数のタスク処理エージェントにより分散処理を実現する Grid の設計モデルである。エージェントは、自らの保持するタスクとともに、そのタスクの処理が可能な PC へ自律的に移動し、移動先の余剰計算機資源を用いてタスクを処理する。すなわち、タスク依頼 PC がタスク処理 PC を指定するのではなく、タスク処理エージェントが自律的にネットワーク上で計算機資源に余裕のある PC を探索し、逐次的にタスクを処理する。タスク処理エージェントに高い自律性を与えることで、タスク依頼 PC とタスク処理 PC を区別する必要なく、だれでもいつでも利用できる Grid の設計を目指すものである。このような設計モデルにより、一度タスクを依頼した PC が電源を切っても、エージェントは移動先の計算機資源を用いたタスク処理を継続できる Grid が期待される。さらに、ネットワーク環境が未知で動的に変化する場合でも、ロードバランシングをユーザが事前に設定する必要のない、柔軟で頑健な分散処理の実現が期待される。

MAO-Grid において、エージェントのタスク処理戦略として表現されるロードバランシングの決定には、環境に応じた適応的な手法が重要である。しかし、従来の学習手法では、個々のエージェントによる自律的な学習では、必ずしもシステム全体として適切なロード

バランシングが得られず、エージェント集団による全体的な学習には、エージェントの集中管理が必要であった。本論文では、MAO-Grid 全体にとって適切なロードバランシングをエージェントの自律学習により適応的に獲得する手法として、動的隔離型遺伝的アルゴリズム (以下 DS-GA と記す)^{9),10)} を適用する。DS-GA は、MAO-Grid が内包するエージェントの学習手法である。エージェントは、チャンネルに動的に隔離され、相互作用をチャンネル内に制限される。MAO-Grid 全体におけるタスク処理量をシステム最適性、システム最適性の向上をシステム最適化と定義し、相互協力型 Grid においてシステム最適なロードバランシングをユーザが気にすることなく実現する Grid の設計モデルとして MAO-Grid を提案し、DS-GA の適用によるシステム最適化について述べる。

本論文の構成は以下のとおりである。2章で、MAO-Grid を提案し、学習手法として適用する DS-GA について述べる。3章で、ボトルネックが計算機資源である場合のロードバランシングのシステム最適化について述べる。4章で、ボトルネックがネットワーク資源である場合のロードバランシングのシステム最適化について述べる。5章で、実際の Grid 環境への適用に向けた応用実験について述べる。6章で、まとめと今後の課題を述べる。

2. エージェント指向グリッドコンピューティング (MAO-Grid) の提案

本章では、複数の PC 間を移動するタスク処理エージェントにより分散処理を実現する MAO-Grid を提案する。また、エージェント学習によりロードバランシングを制御する DS-GA の適用について、従来の単純な GA (以下、sGA と記す) との比較により述べる。

2.1 MAO-Grid の概要

MAO-Grid は、PC 間で共有する計算機資源からなる共有フィールドと、その中に存在する多数のタスク処理エージェントからなる。共有フィールドは、ネットワークに接続された多数の PC から任意に提供された余剰計算機資源からなる。提供される資源は均一である必要はない (図 1)。エージェントは、いずれかの PC の共有フィールド内に存在し、他の PC の共有フィールドへ移動できる。自らの行動を決定する戦略と、タスクの格納領域を持つ。タスクは、処理するプログラムと処理されるデータからなる。

MAO-Grid におけるタスク処理の手順の概要を以下に示す。一般的に、分散処理システムでは、処理すべきタスクに対して以下の手順が必要である。(1) 部分

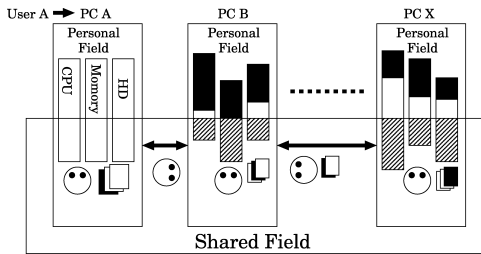


図 1 エージェント指向グリッドコンピューティングの概念
Fig. 1 Multi-agent Oriented Grid Computing (MAO-Grid).

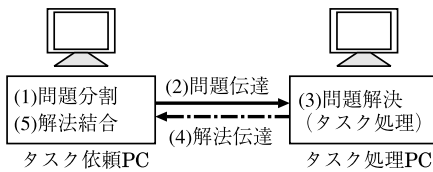


図 2 分散処理されるタスクの一般的な流れ
Fig. 2 General flow of task processing in Grid computing.

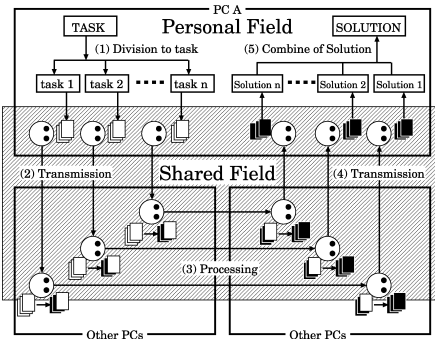


図 3 分散処理されるタスクの流れ
Fig. 3 Flow of task processing in MAO-Grid.

問題への分割, (2) 部分問題の伝達, (3) 部分問題の解決, (4) 部分解法の伝達, (5) 部分解法の結合 (図 2)。MAO-Grid では, (1) 部分問題への分割および (5) 部分解法の結合は, 既存の技術を用いてタスク依頼 PC により実行されるものとし, タスク依頼 PC により分割されたそれぞれの部分問題を MAO-Grid におけるタスクと呼ぶ。このため, MAO-Grid で扱えるタスクは, 一般的な Grid と同様に部分問題への分割が容易なものに限られる。

タスクは, タスク依頼 PC に所属するエージェントに渡される。共有フィールド内の計算機資源では, タスク処理のために必要な共通の処理系 (プログラム言語など) が利用可能であり, エージェントは自らの所属する PC の共有フィールドにおいてタスクを自律的に処理する。処理手順を図 3 に示す。共有フィールド

内のエージェントは, (1) タスク依頼 PC からタスクとその処理プログラムを受け取り, (2) 共有フィールド内を移動する。タスク処理が可能な PC に移動したエージェントは (3) 自らのタスク処理を実行する。処理が完了したエージェントは, (4) 処理を依頼した PC へ移動し, (5) 処理が完了したタスクを PC に渡す。

エージェントは, コンピュータウイルスとは異なり, 共有フィールド以外ではいずれの処理も実行しない。エージェントは, 自らの保持する行動戦略に基づいて移動する PC や処理するタスクなどを決定する。このため, タスク処理に関するロードバランシングは, エージェントの行動戦略により決定される。

タスクの分散処理において適応的な決定が有効である対象は, 必ずしもロードバランシングには限らない。ただし, 本論文では, エージェントが学習する対象はロードバランシングのみとし, エージェントが学習する行動戦略としてエージェントの状態に応じた移動先を示す移動テーブルを用いる。

2.2 MAO-Grid に適用する DS-GA の概要

エージェント学習に適用する DS-GA の基本的な枠組みは以下のとおりである。DS-GA は, MAO-Grid の計算機資源, およびネットワーク資源を複数のチャンネルに離隔する。エージェントは, いずれかのチャンネルに所属し, 他のチャンネルに所属するエージェントとは相互作用しない。より多くのタスク処理により高い評価値を得たエージェントは 2 つに分裂する。また, 低い評価値を得たエージェントは消滅する。チャンネル内エージェント数が規定値を超えた場合, そのチャンネル内エージェントの半数は, 使用されていない他のチャンネルに移住する。すべてのチャンネルが使用されている場合, ランダムに選択したチャンネル内エージェントをすべて消滅させ, そのチャンネルに移住する。これにより, そのチャンネル内エージェントは, 2 つのチャンネルへ動的に離隔される。また, チャンネル内エージェント数が 0 となった場合, そのチャンネルの使用を中止する。MAO-Grid におけるチャンネルの概念を図 4 に示す。

マルチエージェントシステムにおいて有効な最適化手法である DS-GA は, 局所的な情報に基づく学習により, 高いシステム最適性を得る性質がある。そのため, MAO-Grid においても, DS-GA を用いたエージェント学習により, 各エージェントがネットワーク環境全体を把握する必要なく, タスク処理に必要な複数の資源を効率的に利用するロードバランシングを学

本論文では, エージェントの所属 PC の変更をエージェントの移動, 所属チャンネルの変更をエージェントの移住と表現する。

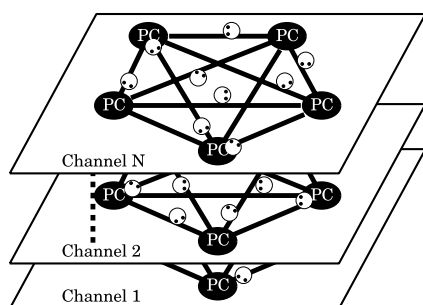


図 4 チャンネルの概念

Fig. 4 Conceptual representation of channels.

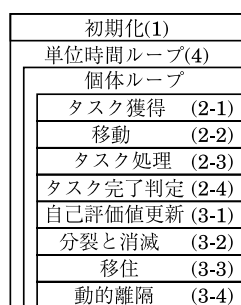


図 5 NS チャートで示す MAO-Grid のアルゴリズム

Fig. 5 Algorithm of MAO-Grid shown by NS chart.

習できると期待される。

2.3 DS-GA の MAO-Grid への適用

DS-GA を MAO-Grid に適用するため, MAO-Grid のエージェント a は, MAO-Grid における行動戦略としての移動テーブルとタスク格納領域に加え, DS-GA における自己評価値 $E_A(a, t)$ とチャンネル番号を持つ。DS-GA を適用した MAO-Grid のアルゴリズムの概要を NS チャートで図 5 に, 詳細を以下に示す。

- (1) 初期化: MAO-Grid に加わる PC は, 自らの余剰計算機資源を MAO-Grid の共有フィールドとして提供する。共有フィールドは N_C のチャンネルに離隔される。各チャンネル内に, N_{Lim} のエージェントを作成する。エージェントは, いずれかの PC に所属する。エージェント a の初期自己評価値は $E_A(a, 0)$ とし, 移動テーブルの値はランダムに選択される。移動テーブルは実験ごとに異なるため, 実験ごとに述べる。
- (2) MAO-Grid によるタスク処理アルゴリズム:
 - (2-1) タスク獲得: 処理すべきタスクを保持しないエージェントは, 処理を依頼すべきタスクが存在する PC に所属する場合, PC からタスクを受け取る。
 - (2-2) 移動: エージェントは, 移動テーブルに従い, タスクとともに次の PC へ移動する。所

属するチャンネルは変化しない。単位時間ごとに, 移動するエージェントは, ランダムな順序で選択される。移動に必要なネットワーク資源が不足するエージェントは, 移動できずにその PC にとどまる。

- (2-3) タスク処理: エージェントは, 共有フィールド内において, 自らが所属する PC, および所属するチャンネルに存在する計算機資源を用いてタスクを処理する。単位時間ごとに, 処理するエージェントはランダムな順序で選択される。処理に必要な計算機資源が不足するエージェントは, 処理できずにその PC にとどまる。
- (2-4) タスク完了判定: 処理の完了したタスクを保持するエージェントは, タスクを依頼した PC に所属する場合, PC にタスクを渡す。
- (3) DS-GA によるエージェント学習アルゴリズム:
 - (3-1) 自己評価値更新: タスク処理を完了したエージェントは, 評価値 E_A が Q 増加する。また, 単位時間ごとに, エージェントの評価値が $Cost$ 減少する。
 - (3-2) 分裂と消滅: 自己評価値が初期値 $E_A(a, 0)$ の 2 倍を超えたエージェントは, 2 つに分裂する。2 つのエージェントは, 自己評価値を半分ずつ引き継ぐ。移動テーブルは, 突然変異確率 P_{mut} で変異する。自己評価値が負になったエージェントは, 消滅する。
 - (3-3) 移住: エージェントは, 移住確率 P_{mig} に従い, ランダムに選択されたチャンネルに移住する。所属する PC は変化しない。
 - (3-4) 動的離隔: チャンネル内エージェント数が増加し, 限界数 N_{Lim} を超えた場合, そのチャンネル内エージェントの半数は, 使用されていないチャンネルに移住する。すべてのチャンネルが使用されている場合は, ランダムに選択されたチャンネル内エージェントをすべて消滅させ, そのチャンネルに移住する。
- (4) 単位時間ループ: 上記 (2), (3) を 1 単位時間とし, 繰り返す。

3. 計算機資源に応じたロードバランシングの最適化

本論文の実験では, 実際の Grid 環境を単純化してモデル化したシミュレーションにより, 提案する MAO-Grid の特徴と DS-GA によるロードバランシングに対する性質について述べ, それらの特徴から示される

有効性について議論する．

環境に応じた適切なロードバランシングを議論するには、計算機資源とネットワーク資源をとともに考慮する必要がある．本章では、計算機資源がボトルネックになる場合、すなわち、前章の図 2 における (3) 問題解決 (タスク処理) が、(2) 問題伝達および (4) 解法伝達より遅い場合について議論する．4 章では、ネットワーク資源がボトルネックになる場合、すなわち、(2) 問題伝達および (4) 解法伝達が、(3) 問題解決 (タスク処理) より遅い場合について議論する．5 章では、計算機資源とネットワーク資源のボトルネックが混在する場合、および多数の PC が存在する場合について議論する．

Grid では、計算機資源が均一なマシンからなるホモジニアス Grid と、多種多様なコンピュータからなるヘテロジニアス Grid に大別できる．計算機資源が均一なホモジニアス Grid では、タスクの計算負荷が均一となるようなロードバランシング (負荷分散) が重要である．一方、インターネットなどで接続された多種多様なコンピュータからなるヘテロジニアス Grid では、負荷分散に加え、異なる機能を持つ計算機資源に応じたロードバランシング (機能分散) が重要である．そこで、本章では、ヘテロジニアス Grid における負荷分散と機能分散という異なるロードバランシングに対する性質について述べる．

本章の実験では、計算機資源がボトルネックである場合について議論するため、ネットワーク資源に制約はないものとする．各パラメータは、初期チャネル数 $N_C = 10,000$ 、チャネルの限界数 $N_{Lim} = 50$ 、突然変異確率 $P_{mut} = 0.01$ 、移住確率 $P_{mig} = 0.01$ 、初期評価値 $E_A(a, 0) = 10$ 、 $Q = 5$ 、 $Cost = 1$ とし、異なるランダム系列を用いて 100 試行する．

3.1 逐次処理を含まないタスクによる検証

3.1.1 実験モデル

本節の実験モデルでは、学習手法の基本的な性質の違いに注目するため、1 ステップで処理が完成する 2 種類のタスク $\{X, Y\}$ を用いる．MAO-Grid の環境として、5 つの PC $\{A, B, C, D, E\}$ を用いる．ロードバランシングの検証を容易にするため、タスク依頼 PC を PC A のみとし、タスクを分散して処理する PC を他の 4 つの PC $\{B, C, D, E\}$ とする．PC A では、タスク X とタスク Y の処理が等しい比率で発生する．PC 間ネットワークの概念を図 6 に示す．

資源に関する負荷分散について検証するため、各 PC が提供する計算機資源の大きさはそれぞれ異なるものとする．また、資源に関する機能分散について検

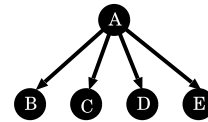


図 6 PC 間ネットワークの概念

Fig. 6 Conceptual representation of PC Network.

表 1 共有フィールドに存在する計算機資源とタスク処理に必要な計算機資源

Table 1 Computational resources in the shared field and computational resources required for task processing.

[余剰計算機資源 $\times 10,000$]			[必要計算機資源 Task X]		
PC	資源 α	資源 β	Step	資源 α	資源 β
PC A	0	0	Step 1	1	3
PC B	2	6			
PC C	6	2	[必要計算機資源 Task Y]		
PC D	4	12	Step	資源 α	資源 β
PC E	12	4	Step 1	3	1

表 2 エージェントの移動テーブル

Table 2 Transfer table of agents.

Task Step	Task X	Task Y
Step 1	T(X,1)	T(Y,1)

$T(*,*) \in \{B, C, D, E\}$

証するため、タスク処理に必要な計算機資源と各 PC が提供する計算機資源は 2 種類 $\{\alpha, \beta\}$ とし、比率はそれぞれ異なるものとする．さらに、各 PC の余剰計算機資源は互いに未知であるものとする．各 PC の提供する単位時間あたりに使用可能な計算機資源、およびタスク X・Y の処理に必要な計算機資源を表 1 に示す．また、エージェントの移動テーブルを表 2 に示す． $T(*,*)$ の値が移動先 PC を示す．

PC A でタスクを依頼されたエージェントは、依頼されたタスクの種類 $\{X, Y\}$ と、初期化時にランダムに選択された自らの移動テーブルの値に従い、いずれかの PC $\{B, C, D, E\}$ に移動する．

PC $\{B, C, D, E\}$ に移動したエージェントは、移動先 PC の同一チャネル内の共有フィールドに存在する計算機資源を用いてタスクを処理する．

本節の実験モデルでは、PC $\{D, E\}$ は、PC $\{B, C\}$ のそれぞれ 2 倍の資源を保持する．そのため、PC $\{D, E\}$ が、PC $\{B, C\}$ のそれぞれ 2 倍のタスクを処理することで負荷分散が実現される．また、タスク X の処理には資源 β を、タスク Y の処理には資源 α を多く必要とし、PC $\{B, D\}$ には資源 β が、PC $\{C, E\}$ には資源 α が多く存在する．そのため、タスク X の処理を PC $\{B, D\}$ で、タスク Y の処理を PC $\{C,$

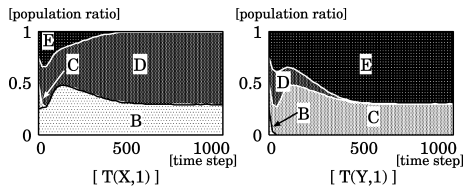


図 7 DS-GA における移動テーブルの値ごとの個体数比率の推移
Fig. 7 History of the population ratio among various transfer table in DS-GA.

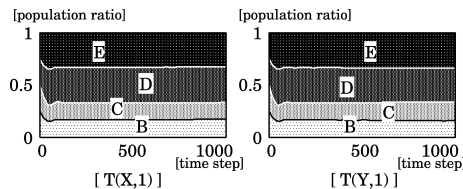


図 8 sGA における移動テーブルの値ごとの個体数比率の推移
Fig. 8 History of the population ratio among various transfer table in sGA.

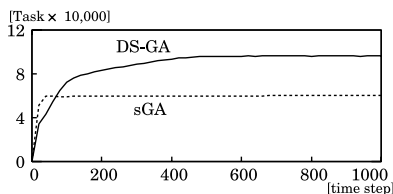


図 9 単位時間あたりのタスク処理数の推移
Fig. 9 History of the average number of processed tasks.

E} で処理することで機能分散が実現される。

3.1.2 実験結果

本節の実験では、100 回の試行すべてにおいて以下のような様相を示した。実験結果の例として、DS-GA および sGA を用いたエージェントの学習結果のうち、移動テーブルの値 $T(X, 1)$, $T(Y, 1)$ ごとの個体数比率の推移を図 7 (DS-GA) および 図 8 (sGA) に示す。それぞれの線の間の領域が、その移動テーブルにおける値の比率を示す。また、全個体によるタスク処理量の推移を図 9 に示す。実線は DS-GA におけるタスク処理量を示し、破線は sGA におけるタスク処理量を示す。

本節の実験モデルでは、DS-GA を用いたエージェント学習では、タスク X を保持するエージェントは PC {B, D} へ、タスク Y を保持するエージェントは PC {C, E} へ移動するテーブルを学習し、機能分散を実現した。また、PC {B, C} へ移動するエージェントのそれぞれ 2 倍のエージェントが PC {D, E} へ移動するテーブルを学習し、負荷分散を実現した。そのため、PC {B, C, D, E} の資源の効率的な利用が

実現できた。

一方、sGA を用いたエージェント学習では、タスクの種類にかかわらず、PC {D, E} が PC {B, C} のそれぞれ 2 倍のタスクを処理する負荷分散は実現したが、機能分散は実現できなかった。その結果、DS-GA により学習した MAO-Grid は、sGA により学習した MAO-Grid に比べ、単位時間あたり約 1.5 倍のタスクを処理できた。

3.1.3 考察

本節の実験において、エージェントは、いずれの学習手法においても負荷分散は実現できた。これは、より負荷の少ない PC へ移動するエージェントほどタスクを処理できる確率が高く、より多くの報酬を得るためである。一方、計算機資源の種類に応じた機能分散は、sGA を用いた単純な学習では実現できなかった。これは、エージェントが、自らのタスクに必要な計算機資源や、各 PC において余裕のある計算機資源の種類を知覚できないためである。すなわち、あるエージェントが資源をより効率良く使用するように移動先の PC を変え、他のエージェントが使用できる資源を増やしたとしても、移動先を変えたエージェントの報酬は増加しない。

これに対し、DS-GA では、機能分散により効率良く資源を利用するエージェントの多いチャネルほど動的離隔により速く増加するため、集団レベルでは“集団全体におけるタスク処理量”を知覚できる。このような、単純な知覚を持つ個体が集団となることで創発する高次の知覚である“群知覚”¹¹⁾に基づく学習により、最適な機能分散が実現できたと考えられる。

MAO-Grid では、単純な学習手法を用いても負荷分散に関するロードバランシングは最適化が可能であるが、DS-GA の適用により、機能分散に関するロードバランシングも最適化が可能であるといえる。

3.2 逐次処理を含むタスクによる検証

逐次的に複数の処理ステップが必要なタスクでは、処理ステップごとに必要な計算機資源が異なる場合がある。また、MAO-Grid では、自律的にタスクを処理するエージェントが、処理ステップごとにタスク依頼 PC に戻る必要はない。このため、それぞれの処理ステップごとに適切なタスク処理 PC を探索するロードバランシングの最適化が有効である。

本節では、複数の処理ステップが必要な逐次処理を含むタスクに対する性質を実験的に検証する。実験モデルとして、処理ステップごとに処理に必要な資源が異なるため、処理ステップごとに機能分散の最適化が必要なタスクを用いる。

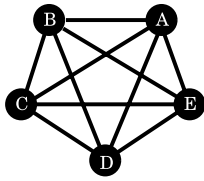


図 10 PC 間ネットワークの概念

Fig. 10 Conceptual representation of PC network.

表 3 共有フィールドに存在する計算機資源

Table 3 Computational resources in the shared field.

[余剰計算機資源×10,000]

PC 資源	資源 α	資源 β
PC A	8	1
PC B	1	16
PC C	5	9
PC D	17	5
PC E	3	3

表 4 タスク処理に必要な計算機資源

Table 4 Computational resources required for task processing.

[必要計算機資源 Task X]			[必要計算機資源 Task Y]		
Step	種類	資源 α	Step	種類	資源 α
Step 1		1	Step 1		16
Step 2		1	Step 2		8
Step 3		1	Step 3		4
Step 4		1	Step 4		2

3.2.1 実験モデル

本節の実験モデルでは，MAO-Grid の環境として，5 つの PC {A, B, C, D, E} を用いる（図 10）．このうち，タスクを依頼する PC は PC {A, B} とし，タスクを処理する PC は 5 つすべての PC {A, B, C, D, E} とする．また，2 種類の計算機資源 {α, β} と 2 種類のタスク {X, Y} を用いる．タスク X は PC A で，タスク Y は PC B で，それぞれ等しい比率で発生する．タスク {X, Y} は，それぞれ 4 つのステップ {1, 2, 3, 4} で処理され，ステップごとに必要な計算機資源が異なる．エージェントの学習結果がシステム最適であるかどうかの検証を容易にするため，エージェントの移動テーブルのシステム最適な集合が一意に定まる値として，単位時間あたりに使用可能な各 PC の計算機資源を表 3 に，タスク {X, Y} の処理ステップごとに必要な計算機資源を表 4 に示すものとする．

PC {A, B} でタスクを依頼されたエージェントは，自らの依頼されたタスクの種類とそのタスクの処理ステップに応じて，移動テーブル（表 5）の示すいずれかの PC {A, B, C, D, E} に移動する．移動したエージェントは，移動先 PC の同一チャネル内の共有

表 5 エージェントの移動テーブル

Table 5 Transfer table of agents.

Task Step	Task X	Task Y
Step 1	T(X,1)	T(Y,1)
Step 2	T(X,2)	T(Y,2)
Step 3	T(X,3)	T(Y,3)
Step 4	T(X,4)	T(Y,4)

$T(*,*) \in \{A, B, C, D, E\}$

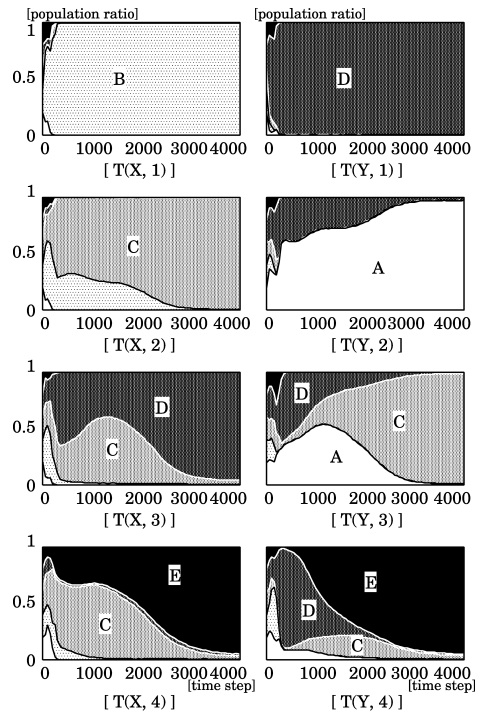


図 11 DS-GA における移動テーブルの値ごとの個体数比率の推移

Fig. 11 History of the population ratio among various transfer table.

フィールドに存在する計算機資源を用いてタスクを処理する．

本節の実験モデルでは，タスク X を PC {A → B → C → D → E → A} の順に，タスク Y を PC {B → D → A → C → E → B} の順に処理することですべての資源を効率的に利用できる．

3.2.2 実験結果

本節の実験では，100 回の試行すべてにおいて以下のような様相を示した．実験結果の例として，DS-GA を用いたエージェントの学習結果のうち，移動テーブルの値ごとの個体数比率の推移を図 11 (DS-GA) に示す．

DS-GA を用いたエージェント学習では，いずれの種類のタスクを保持するエージェントも，MAO-Grid 全体にとって最適なロードバランシングを実現する移動テーブルを学習した．このときのタスク X および

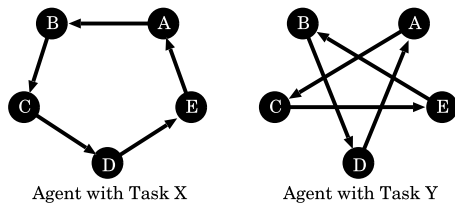


図 12 DS-GA におけるエージェントが獲得した移動経路
Fig. 12 Acquired routes by DS-GA.

Y を保持するエージェントの移動経路を図 12 に示す．PC {A, B, C, D, E} の資源の効率的な利用が実現できた．

3.2.3 考 察

本節の実験では、約 1000 単位時間において、一部の移動テーブルに関してシステム最適な経路を示す値とは異なる値を持つ個体が一時的に増加した．具体的には、 $T(X, 3) = C$ (最適解は D), $T(X, 4) = C$ (最適解は E), $T(Y, 3) = A$ (最適解は C), $T(Y, 4) = D$ (最適解は E) などである．これらの値が示す PC は、システム最適な経路が示す PC に比べ、そのタスクの処理ステップに関して処理能力が高い．より処理能力が高い PC へ移動するエージェントほど処理できる確率が高く、個体最適性は高い．このため、初期においてこれらの PC へ移動するエージェントが増加したと考えられる．

しかし、処理される確率が高いがシステム最適ではない上記の PC へエージェントが移動する場合、MAO-Grid 全体で処理できるタスクの量は、システム最適な経路に比べ小さい．そのため、群知覚を持つ DS-GA では、集団レベルで“チャンネル全体におけるタスク処理量”を知覚し、最終的にはそれを最大化する最適な行動を学習できたと考えられる．

タスクの処理が逐次的な環境においても、DS-GA は集団レベルで創発する群知覚に基づき、最適な機能分散が実現できた．このため、DS-GA を適用した MAO-Grid は、処理ステップごとの逐次的なタスク処理 PC の探索により、ロードバランシングの最適化が可能であるといえる．また、本節の結果は、物理的なネットワーク上において、効率的なタスク処理に必要な論理的なネットワークをボトムアップ的に獲得しており、MAO-Grid における新たなネットワーク生成法としての可能性が示唆される．

なお、sGA を用いたエージェント学習では、エージェントの移動テーブルの収束は見られず、4000～5000 単位時間におけるタスク処理量は、DS-GA を用いた場合の約 63%であった．ここで、負荷分散のみが

表 6 環境変化後の共有フィールドに存在する計算機資源
Table 6 Computational resources in the share field after environmental change.

[余剰計算機資源×10,000]		
PC 資源	資源 α	資源 β
PC A	8	1
PC B	1	16
PC C	3	3
PC D	17	5
PC E	5	9

可能な sGA により実現されるタスク処理量と、負荷分散だけでなく機能分散も可能な DS-GA により実現されるタスク処理量の違いは、手法を適用する PC 間における機能の不均一性に依存する．このため、すべての PC の機能が均一である環境では、DS-GA により実現されるタスク処理量は、sGA により実現されるタスク処理量と原理的に等しいといえる．逆に、使用可能な資源や処理に必要な資源に関する不均一性が大きい場合、負荷分散のみが可能な sGA を用いた場合のタスク処理量と、機能分散も可能な DS-GA を用いた場合の処理量の違いは大きい．

3.3 動的環境への適応

実際のネットワーク環境では、各々の計算機資源が動的に変化する可能性がある．本節では、DS-GA を適用した MAO-Grid の動的環境に対する性質を検証するため、前節の実験モデルにおいて、5000 単位時間から各 PC の計算機資源を変化させる．ここでは、最適解の検証が容易なように、PC C と PC E の計算機資源の値を入れ替える．変化後の計算機資源を表 6 に示す．

本節の実験では、100 回の試行すべてにおいて以下のような様相を示した．実験結果の例として、DS-GA を用いたエージェントの学習結果のうち、移動テーブルの値ごとの個体数比率の推移を図 13 に、エージェントの学習した移動経路を図 14 に示す．動的に変化した環境においても MAO-Grid 全体にとって最適な移動テーブルを学習したことから、DS-GA を適用した MAO-Grid は、計算機資源が未知で動的に変化する環境においても、適応的にシステム最適なロードバランシングを得る性質を持つといえる．

4. ネットワーク資源に応じたロードバランシングの最適化

インターネットのような大規模ネットワークでは、分散処理のボトルネックが計算機資源ではなく、ネットワーク資源である場合もある．また、MAO-Grid が利用可能なネットワーク資源は動的に変化する．本章で

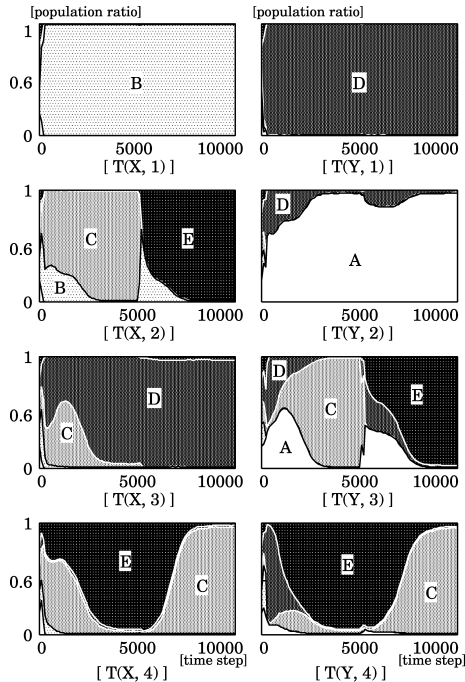


図 13 環境変化後の移動テーブルの値ごとの個体数比率
Fig. 13 History of the population ratio among various transfer table after environmental change.

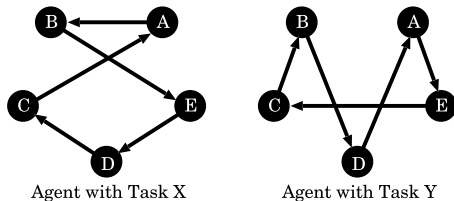


図 14 エージェントが獲得した環境変化後の移動経路
Fig. 14 Acquired routes after environmental change.

は、未知で動的なネットワーク資源に注目し、DS-GAを用いた学習によるシステム最適化について述べる。

本章の実験では、ネットワーク資源がボトルネックである場合について議論するため、計算機資源に制約はないものとする。各パラメータは、初期チャネル数 $N_C = 20,000$ 、チャネルの限界数 $N_{Lim} = 50$ 、突然変異確率 $P_{mut} = 0.05$ 、移住確率 $P_{mig} = 0.01$ 、すべてのエージェントの初期評価値 $E_A(a, 0) = 10$ 、 $Q = 10$ 、 $Cost = 0.5$ とし、異なるランダム系列を用いて 100 試行する。

4.1 2 ステップタスクによる検証

4.1.1 実験モデル

本章では、図 15 に示す単純な実験モデルを用いる。図中の数字はネットワーク資源の量を示す。

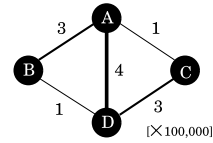


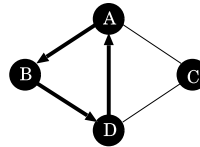
図 15 PC 間ネットワークの概念
Fig. 15 Conceptual representation of PC network.

表 7 タスクのサイズとエージェントの移動テーブル

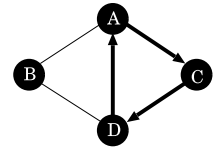
Table 7 Task size and Transfer table of agent.

(A) Size of Task			(B) Transference Table of Agent		
State \ Task	Task X	Task Y	State \ Task	Task X	Task Y
State 0	3	1	State 0	T(X,0)	T(Y,0)
State 1	1	3	State 1	T(X,1)	T(Y,1)
State 2	3	1	State 2	T(X,2)	T(Y,2)

$T(*,*) \in \{A, B, C, D\}$



(A) Optimal route of Task X



(B) Optimal route of Task Y

図 16 タスク X とタスク Y に関する最適な経路

Fig. 16 Optimal routes of task X and task Y.

本節の実験モデルでは、4 つの PC $\{A, B, C, D\}$ と 5 つのリンク、2 種類のタスク $\{X, Y\}$ を用いる。4 つの PC は、異なる種類の計算機資源を保持する。PC A は、タスクを依頼する PC であり、PC $\{B, C, D\}$ はタスクを処理する PC である。ただし、PC $\{B, C\}$ はステップ 1 の処理ができるが、ステップ 2 の処理はできず、PC D はステップ 2 の処理はできるが、ステップ 1 の処理はできないものとする。5 つのリンクは、異なるネットワーク資源を保持する。リンク $\{AB, AC, AD, BD, CD\}$ は、それぞれ図 15 に示すネットワーク資源を保持する。

PC A では、タスク X とタスク Y の処理が等しい比率で発生する。タスク X とタスク Y は、処理ステップ 1 により状態 0 から状態 1 へ、ステップ 2 により状態 1 から状態 2 へ変化する。タスクのサイズは、表 7 (A) のとおり、処理に応じて変化する。

本項の実験モデルでは、エージェントの移動テーブルは、表 7 (B) のとおり、自らの保持するタスク種類と状態に基づいて次に移動する PC を示す。

タスク X を保持するエージェントは、 $A \rightarrow B \rightarrow D \rightarrow A$ の経路で移動する場合 (図 16 (A)) に、タスク Y を保持するエージェントは、 $A \rightarrow C \rightarrow D \rightarrow A$ の経路で移動する場合 (図 16 (B)) に、ネットワーク資源を有効に利用できる。しかし、各エージェントは

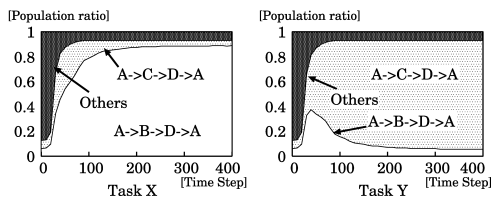


図 17 DS-GA における移動テーブルの値ごとの個体数比率の推移
Fig. 17 History of the population ratio among various transfer table in DS-GA.

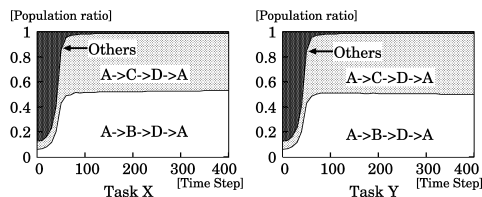


図 18 sGA における移動テーブルの値ごとの個体数比率の推移
Fig. 18 History of the population ratio among various transfer table in sGA.

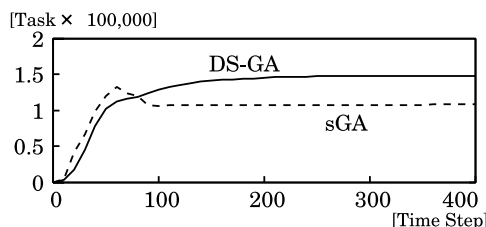


図 19 単位時間あたりのタスク処理量の推移
Fig. 19 History of the average number of processed tasks.

ネットワーク資源を直接知覚することはできない。

4.1.2 実験結果

本節の実験では、100 回の試行すべてにおいて以下のような様相を示した。実験結果の例として、DS-GA および sGA を用いたエージェントの学習結果のうち、学習した移動経路ごとの個体数比率の推移を図 17 (DS-GA) および図 18 (sGA) に示す。また、全個体によるタスク処理量の推移を図 19 に示す。

sGA と DS-GA の実験結果から、ほとんどのエージェントはいずれも 50 単位時間以内に $A \rightarrow B \rightarrow D \rightarrow A$ 、または $A \rightarrow C \rightarrow D \rightarrow A$ のいずれかの移動経路を学習した。ただし、DS-GA による学習においてのみ、エージェントはタスク X を保持する場合に PC B へ、タスク Y を保持する場合に PC C へ移動する経路を学習し、MAO-Grid 全体として多くのタスクを処理できた。

4.1.3 考察

本節の実験では、 $A \rightarrow B \rightarrow D \rightarrow A$ または $A \rightarrow C \rightarrow D \rightarrow A$ の経路で移動するエージェン

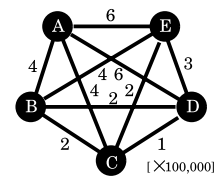


図 20 PC 間ネットワークの概念
Fig. 20 Conceptual representation of PC network.

表 8 処理可能な PC と処理によるタスクサイズの変更比率
Table 8 Executable PCs and the changing ratio of task sizes.

(A) Executable PC		(B) Magnification of Task Size	
Program	Executable PC	Program	Magnification [times]
Program B	PC B	Program B	$\times 0.5$
Program C	PC C	Program C	$\times 0.5$
Program D	PC D	Program D	$\times 3$
Program E	PC E	Program E	$\times 2$

トだけがタスク処理を完了できる。エージェントはいずれの経路を移動しても等しい評価を得るため、どちらの経路が適切であるかを直接知覚できない。そのため、エージェントがシステム最適な経路を移動するテーブルを保持していた場合でも、エージェント知覚のみに基づいて学習した場合、そのエージェントの比率は増加しない。

DS-GA では、チャンネル内のエージェントが他のチャンネル内のエージェントよりシステム最適である場合、そのチャンネルに所属するエージェントの比率は群知覚により増加する。その結果、システム最適な経路を移動するエージェントが全体を占めたと考えられる。

MAO-Grid では、単純な学習手法を用いた場合には、ネットワーク資源に関するロードバランシングのシステム最適化は必ずしも可能ではないが、DS-GA の適用によりネットワーク資源に関するロードバランシングのシステム最適化が可能であるといえる。

4.2 4 ステップタスクによる検証

4.2.1 実験モデル

本節では、より複雑なモデルとして、4 ステップで処理が完了するタスクを用いる。実験モデルを図 20 に示す。図中の数字はネットワーク資源の量を示す。

このモデルは 5 つの PC $\{A, B, C, D, E\}$ と 10 のリンクからなる。各リンクのネットワーク資源を図 20 に示す。PC A はタスクを依頼する PC である。PC B, PC C, PC D, および PC E は、それぞれプログラム B, C, D, および E により、タスクを処理する PC である (表 8 (A))。

また、本節では、プログラム B, C, D, E, およびデータからなる 1 種類のタスク X を用いる。タスク

表 9 数学的に求められたエージェントの最適移動テーブル
Table 9 Optimal transfer table of agent calculated mathematically.

(A) Transference Table		(B) Optimal Table 1		(C) Optimal Table 2	
current PC	Next PC	current PC	T(X,*)	current PC	T(X,*)
PC A	T(X,A)	PC A	B	PC A	C
PC B	T(X,B)	PC B	C	PC B	D
PC C	T(X,C)	PC C	D	PC C	E
PC D	T(X,D)	PC D	E	PC D	A
PC E	T(X,E)	PC E	A	PC E	B

$T(X,*) \in \{A, B, C, D, E\}$

X はプログラム B, C, D, および E により順不同で処理される。

タスク X のサイズは, 処理ごとに変化する (表 8 (B))。タスク X の初期サイズは 4 である。タスクのサイズと比べてエージェントのサイズは小さいものとする。タスクが 4 つの異なる PC {B, C, D, E} の異なる処理ステップ {B, C, D, E} によりそれぞれ 1 回ずつ処理され, PC A に戻ると処理は完了する。

エージェントの移動テーブルは, エージェントが所属する PC に応じて次に移動する PC を示す (表 9 (A))。

ランダムに選択されたエージェントから移動するが, 移動によるネットワークの通信量が各リンクの限界量を超える場合, そのエージェントは移動できないものとする。エージェントの学習結果がシステム最適であるかどうかの検証を容易にするため, エージェントの移動テーブルのシステム最適な集合が一意に定まる値として, ネットワーク構造とネットワーク資源は図 20 に示すものとする。最適な経路の組合せは, 半数のエージェントが最適テーブル 1, 残りの半数が最適テーブル 2 (表 9 (B), (C)) を獲得した組合せである (付録参照)。

4.2.2 実験結果と考察

本節の実験では, 100 回の試行すべてにおいて以下のような様相を示した。実験結果の例として, DS-GA における移動テーブルごとの個体数比率の推移を図 21 に示す。この結果におけるエージェントの経路を図 22 に示す。

エージェントはシステム最適性にとって最適な経路を学習した。この結果は, 前節で示した DS-GA の性質が, より複雑な環境においても現れたといえる。このため, DS-GA を適用した MAO-Grid は, 処理ステップごとの逐次的なタスク処理 PC の探索により, 自らのタスクにとって適切な経路選択によるロードバランシングの最適化が可能であるといえる。

なお, sGA を用いたエージェント学習では, エージェントの移動テーブルの収束は見られなかった。ま

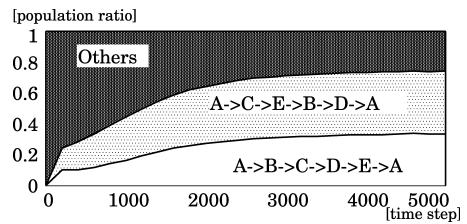


図 21 移動テーブル 1, 2 を持つ個体数比率の推移

Fig. 21 History of the population ratio among transfer table 1 and 2.

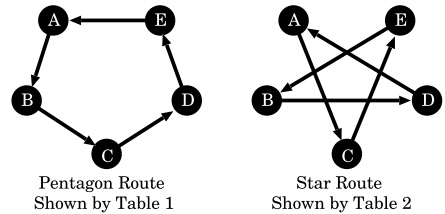


図 22 エージェントが獲得した最適な移動経路

Fig. 22 Acquired optimal routes.

た, 4000~5000 単位時間におけるタスク処理数に関して, DS-GA を用いた場合に比べて優位な差は見られなかった。これは, 本実験においては, システムにとって最適ではない準最適解の多くが, 最適解と近いタスク処理量を実現できるためであると考えられる。

本節の結果も, 前章の結果と同様に, 物理的なネットワーク上において論理的なネットワークをボトムアップ的に獲得している。特に, 次に移動するリンクに注目するとネットワーク資源に余裕があるが, その次のリンクには余裕のない場合においても, 移動経路全体に関して適切な移動先を獲得していることから, 複雑なネットワークにおけるトラフィックの新たな制御法としての可能性が示唆される。

4.3 動的環境への適用

実際のネットワーク環境では, ネットワーク資源の状態は動的に変化する。本節の実験では, 5000 単位時間においてネットワーク資源を図 23 に示すように変更する。変化部分を下線で示す。このときのシステム最適な経路の組合せは, 半数がテーブル 1 を, 残りの半数が表 10 に示すテーブル 3 を獲得した場合である (付録参照)。

本節の実験では, 100 回の試行すべてにおいて以下のような様相を示した。実験結果の例として, DS-GA を用いたエージェントの学習結果のうち, テーブル 1, テーブル 2 およびテーブル 3 を持つエージェントの個体数比率の推移を図 24 に示す。動的環境においても, エージェントはシステム最適な経路を DS-GA により

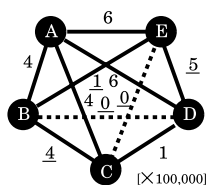


図 23 ネットワーク資源の変化

Fig. 23 Environmental change of network resources.

表 10 数学的に求められたネットワーク変化後のシステム最適なエージェントの移動テーブル

Table 10 Optimal transfer table of agent after environmental change.

Optimal Table 1		Optimal Table 3	
current PC	T(X,*)	current PC	T(X,*)
PC A	B	PC A	C
PC B	C	PC B	E
PC C	D	PC C	B
PC D	E	PC D	A
PC E	A	PC E	D

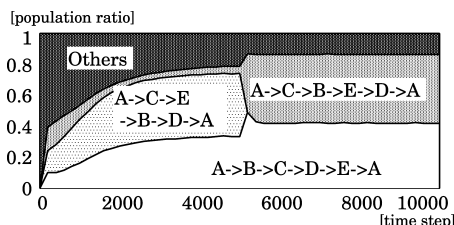


図 24 移動テーブル 1 と移動テーブル 3 を持つエージェント数比率の推移

Fig. 24 History of the population ratio among transfer table 1 and 3.

学習した、DS-GA を適用した MAO-Grid は、ネットワーク資源が未知で動的に変化する環境においても、適応的にシステム最適なロードバランシングを得る性質を持つといえる。

5. 実際の環境への適用に向けて

実際のネットワーク環境では、計算機資源とネットワーク資源の両方が不均一な場合がある。また、数多くの PC が接続されており、それらがつねに使用可能とは限らない。本章では、実際の Grid 環境への適用に向けて、これらの環境における MAO-Grid の有効性について議論する。

5.1 計算機資源とネットワーク資源の両方に対する機能分散

5.1.1 実験モデル

本節の実験では、計算機資源とネットワーク資源の両方に対し機能分散が必要な場合において検証する。計算機資源は 1 種類、タスクはそれぞれ 1 ステップで処理が完成する 2 種類 $\{X, Y\}$ とする。

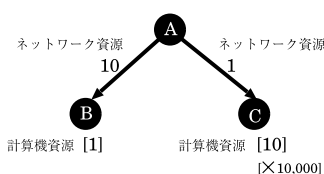


図 25 PC 間ネットワークの概念

Fig. 25 Network resources between PCs and computational resources in the shared field.

表 11 タスクのサイズとタスク処理に必要な計算機資源およびエージェントの移動テーブル

Table 11 Computational resources required for task processing, task size and transfer table of agent.

(A) タスクのサイズ		
Task	Task X	Task Y
State	Task X	Task Y
State 0	10	1

(B) 必要計算機資源		
Task	Task X	Task Y
State	Task X	Task Y
State 0	1	10

(C) エージェントの移動テーブル		
Task	Task X	Task Y
State	Task X	Task Y
State 0	T(X,0)	T(Y,0)

$T(*,*) \in \{A, B\}$

本節の実験モデルにおける余剰計算機資源とネットワーク資源を図 25 に、タスクのサイズとタスク処理に必要な計算機資源、エージェントの移動テーブルを表 11 に示す。

各パラメータは、初期チャネル数 $N_C = 10,000$ 、チャネルの限界数 $N_{Lim} = 50$ 、突然変異確率 $P_{mut} = 0.01$ 、移住確率 $P_{mig} = 0.001$ 、初期評価値 $E_A(a, 0) = 10$ 、 $Q = 5$ 、 $Cost = 1$ とし、異なるランダム系列を用いて 100 試行する。

5.1.2 実験結果と考察

本節の実験では、100 回の試行すべてにおいて以下のような様相を示した。実験結果の例として、DS-GA および sGA を用いたエージェントの学習結果のうち、移動テーブルの値 $T(X, 0)$ 、 $T(Y, 0)$ ごとの個体数比率の推移を図 26 に、タスク処理量の推移を図 27 に示す。

エージェントはシステム最適なタスクの処理 PC を学習した。この結果は、前章までで示した DS-GA の性質が計算機資源とネットワーク資源の両方が不均一な環境においても現れており、DS-GA を適用した MAO-Grid は、このような環境においても適応的にシステム最適なロードバランシングを得る性質を持つといえる。

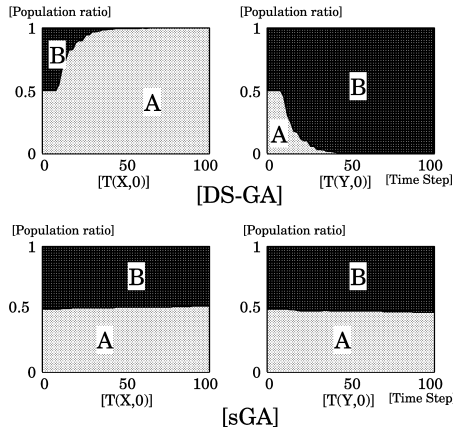


図 26 移動テーブルの値ごとの個体数比率の推移

Fig. 26 History of the population ratio among various transfer table.

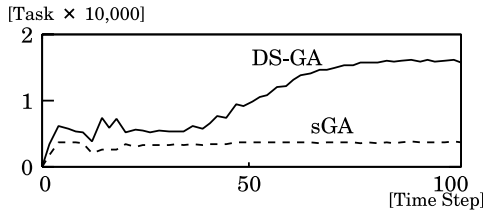


図 27 単位時間あたりのタスク処理数の推移

Fig. 27 History of the average number of processed tasks.

5.2 多数の計算機が存在する環境への適用

5.2.1 実験モデル

本節の実験では、多数の PC が存在し、各 PC の余剰計算機資源に対し機能分散が必要な場合において検証する。ただし、3 章の実験と同様に、ネットワーク資源に制約はないものとする。

本節の実験における PC 数は 2,000、計算機資源は 2 種類 $\{\alpha, \beta\}$ 、タスクはそれぞれ 1 ステップで処理が完成する 2 種類 $\{X, Y\}$ とする。PC は、余剰計算機資源 α の多い α 群と、 β の多い PC を β 群の 2 種類とし、それぞれランダムに選択する。

2,000 の PC のうち、初期化時に使用可能な 200 の PC をランダムに選択する。また、200 単位時間ごとに、使用可能であった PC から使用不可能となる PC と、使用不可能であった PC から新たに使用可能となる PC を、それぞれ 40 ずつランダムに選択する。

余剰計算機資源とタスク処理に必要な計算機資源、エージェントの移動テーブルを表 12 に示す。 $T(*, *)$ の値が移動先 PC を示し、各 PC の余剰計算機資源や使用可能性は未知であるものとする。

各パラメータは、初期チャネル数 $N_C = 100$ 、チャネルの限界数 $N_{Lim} = 10,000$ 、突然変異確率

表 12 タスクのサイズとタスク処理に必要な計算機資源およびエージェントの移動テーブル

Table 12 Computational resources required for task processing, task size and transfer table of agent.

(A) 必要計算機資源		
Task \ 資源	資源 α	資源 β
Task X	5	1
Task Y	1	5

(B) 余剰計算機資源 ($\times 1,000$)		
PC \ 資源	資源 α	資源 β
α 群 PC	10	2
β 群 PC	2	10

(C) エージェントの移動テーブル

Task \ State	Task X	Task Y
State 0	$T(X, 0)$	$T(Y, 0)$

$T(*, *) \in \{0, 1, \dots, 1999\}$

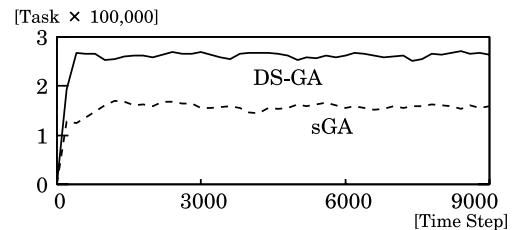


図 28 単位時間あたりのタスク処理数の推移

Fig. 28 History of the average number of processed tasks.

$P_{mut} = 0.01$ 、移住確率 $P_{mig} = 0.001$ 、初期評価値 $E_A(a, 0) = 10$ 、 $Q = 5$ 、 $Cost = 0.5$ とし、異なるランダム系列を用いて 100 試行する。

5.2.2 実験結果と考察

本節の実験では、100 回の試行すべてにおいて以下のような様相を示した。実験結果の例として、DS-GA および sGA を用いたエージェントの学習結果のうち、タスク処理量の推移を図 28 に、移動テーブルの値 $T(X, 0)$ 、 $T(Y, 0)$ に関して、 α 群と β 群のそれぞれの PC へ移動する個体数比率の推移を図 29 に、使用可能な PC へ移動する個体数比率の推移を図 30 に示す。

図 29 より、DS-GA を適用した MAO-Grid では、タスクと計算機資源に応じた機能分散を実現したことが分かる。また、図 30 より、いずれの学習手法においても 2,000 の PC から使用可能な PC を学習し、負荷分散を実現したことが分かる。

一般に GA では、解の探索空間が大きいかほど学習効率が悪くなる。このため、すべてのタスクを集中管理し、各タスクの処理 PC を探索する方法では、PC 数が増加するほど探索効率が悪い。一方、MAO-Grid で

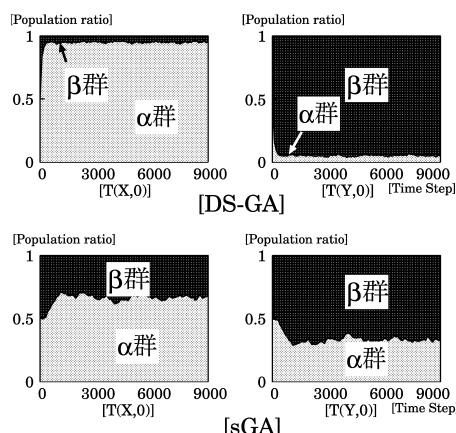


図 29 α 群 PC と β 群 PC へ移動する個体数比率の推移
Fig. 29 History of the population ratio of group α and β .

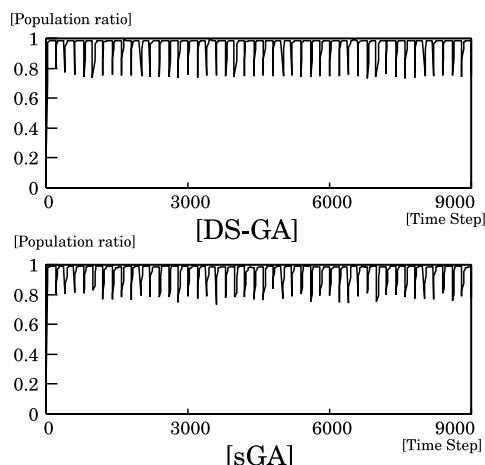


図 30 使用可能な PC へ移動する個体数比率の推移
Fig. 30 History of the population ratio of available PCs.

探索されるロードバランシング，すなわち，エージェントの移動テーブルとその個体数比率は，DS-GA と sGA のいずれの手法においてもエージェントの増加率の違いにより学習される．エージェントの増加率の違いは，PC 間における処理効率の違いに依存し，PC 数に依存しない．すなわち，PC 数がいくつであっても，より処理効率の高い PC でタスクを処理するエージェントがより速く増加する．このため，PC 数に対し十分なエージェント数であれば，原理的に負荷分散に関するロードバランシングの探索効率には，PC 数に依存せず等しい．また，この性質は，ネットワーク資源に関する負荷分散についても同様である．

一方，DS-GA の持つ群知覚による機能分散に関しては，チャンネル間における処理効率の差に基づくため，チャンネル内 PC 数が増加するほど探索効率が低下す

る．このため，PC 数が数万以上となる場合において不均一な資源に対する機能分散を実現するためには，PC を数百～数千といったある程度の数でグループ分けし，各グループ内でチャンネルに動的に離隔する必要があると考えられる．このとき，グループとグループの間の機能が不均一である場合には，機能分散は実現されない．しかし，ある程度多数の PC からなるグループ間，たとえば，関東の PC グループと関西の PC グループの間や，日本の PC グループとアメリカの PC グループの間では，機能の不均一性が小さいと考えられる．このため，DS-GA を適用した MAO-Grid は，グループ間では効率的に負荷分散が実現でき，グループ内では効率的に負荷分散と機能分散を実現できるため，新たな Grid の設計モデルとして有効であるといえる．

6. おわりに

本論文では，多数のタスク処理エージェントにより分散処理を実現する Grid の設計モデルとして，エージェント指向グリッドコンピューティング (MAO-Grid) を提案した．また，MAO-Grid の提案と DS-GA の適用により，集中管理を必要としない自律的なタスク処理エージェントによるロードバランシングの最適化を示した．さらに，MAO-Grid に DS-GA を適用した実験から，計算機資源とネットワーク資源が未知で動的に変化する環境において，エージェントの適切な経路，およびタスク処理 PC の選択により，MAO-Grid 全体のタスク処理量を最大化するロードバランシングを獲得した．これらの結果から，群知覚を持つ DS-GA を適用した MAO-Grid という設計モデルの有効性を示した．また，実験において，効率的なタスク処理に必要な論理的なネットワークをボトムアップ的に獲得しており，新たなネットワーク生成法やトラフィックの制御法としての可能性が示唆された．

本論文の実験モデルでは，すべてのタスクの処理に対し均一な評価を与えている．DS-GA では，チャンネル内の全エージェントが得る報酬を最大化する学習手法であるため，より重要なタスクにはより大きい報酬を与えることで，重要なタスクを優先的に処理できる．また，MAO-Grid の利用者は，自らの依頼するタスクに対しより高い報酬を与えることで，自らのタスクを優先的に処理できる．このため，報酬を与える権利自体を MAO-Grid の使用料に応じて決定することで，利用者の使用料に応じたタスクの優先処理を容易に実現できる．これらの実装方法は，今後の課題である．

なお，本論文で明らかにした Grid の設計モデルは

概念的なものである．実際に Grid を設計する場合には，セキュリティやタスク分割方法など，様々な問題がある．このため，具体的な実装方法は今後の課題であるが，本論文の結果は，ユーザが意識することなく適切なロードバランシングを実現する新しい Grid の設計モデルを示したといえる．また，この設計モデルは，分散処理 Grid だけでなくデータ Grid への応用も可能であると考えられ，これらへの応用も今後の課題である．

謝辞 本研究は独立行政法人情報通信研究機構の研究委託「人間情報コミュニケーションの研究開発」により実施したものである．

参 考 文 献

- 1) <http://www.globus.org>
- 2) Frey, J., Tannenbaum, T., Livny, M., Foster, I. and Tuecke, S.: Condor-G: a computation management agent for multi-institutional grids, *Proc. 10th IEEE International Symposium on High Performance Distributed Computing (HPDC10)*, pp.55–63 (2001).
- 3) 谷村勇輔, 廣安知之, 三木光範: Grid 計算環境における遺伝的アルゴリズムの提案, 情報処理学会論文誌, Vol.43, No.4, pp. 986–994 (2002).
- 4) Casanova, H. and Dongarra, J.: Netsolve: A Network Server for Solving Computational Science Problems, *The International Journal of Supercomputer Applications and High Performance Computing*, Vol.11, No.3, pp.212–223 (1997).
- 5) Sterling, T., Salmon, J., Becker, D.J. and Savarese, D.F.: *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*, The MIT Press (1999) .
- 6) <http://setiathome.ssl.berkeley.edu/>
- 7) Fukuda, M., Tanaka, Y., Suzuki, N., Bic, L.F. and Kobayashi, S.: A Mobile-Agent-Based PC Grid, *Autonomic Computing Workshop*, pp.142–150 (2003).
- 8) Manola, F. and Thompson, C.: Characterizing the agent grid (1999).
<http://www.objs.com/agility/tech-reports/990623-characterizing-the-agent-grid.html>
- 9) 中山功一, 松井博和, 野村由司彦: 動的離隔型 GA の提案, 情報処理学会論文誌：数理モデル化と応用, 進化的計算特集号, Vol.43, No.SIG 10 (TOM7), pp.95–109 (2002).
- 10) Nakayama, K., Shimohara, K. and Katai, O.: Dynamically Separating GA: A New Method of Achieving the System-level Optimality in MAS, *The journal of the three dimensional images*, Vol.16, No.4, pp.177–183 (2002).
- 11) Nakayama, K., Matsui, H., Shimohara, K. and Katai, O.: Multi-Agent Oriented Distributed Computing Optimized by DS-GA, *Proc. Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems*, pp.THE-1-2(CD-ROM) (2004).

付 録

4 章の実験モデルにおけるシステム最適解を数学的に求める．各エージェントが持つ移動テーブルは，状態数が 5，行動数が 5 であるため，これらの組合せは $5^5 (=3,125)$ 通りである．しかし，エージェントはつねにタスクを依頼する PC A から移動を始め，他の 4 つの PC {B, C, D, E} に 1 回ずつ移動し，PC A に戻る必要がある．このため，上記を満たしタスク処理を完了する ${}_4P_4 (=24)$ 通りのエージェントに関して考える．

24 種類の経路 (ABCDEA, ABCEDA, ..., AED-CBA) を *ROUTES*, 24 種類の経路の 1 つである $R \in ROUTES$ を通るエージェントを $Agents_R$, $Agents_R$ の数を N_R とそれぞれするとき，最適解は， $\sum_{R \in ROUTES} N_R$ を最大にする $N_R (R \in ROUTES)$ の組合せである．以下では，最大値を得る場合に，それぞれのエージェント数が満たすべき制約について考える．

24 の経路のすべてに対してエージェント数 $N_R (R \in ROUTES)$ が 0 以上という制約を持つので，以下に示す 24 の制約式が得られる．

$$N_R \geq 0 (\forall R \in ROUTES) \quad (1)$$

2 つの PC を結ぶリンク $L (AB, AC, \dots, DE)$ は，それぞれネットワーク資源量 $Limit_L$ という制約を持つ．5 つの PC に対し $10 (={}_5C_2)$ 個のリンクが存在するので，計 10 個の制約を持つ．例として，リンク AB の制約について考える．24 種類の経路のうち，リンク AB を通る経路は， xyz を任意の C, D, E としたとき， $Agents_{ABxyzA}$, $Agents_{ABxyzBA}$ の 12 種類である．テーブル表 8 (B) に基づいて，リンク AB を通るときのタスクのサイズについて計算すると， $Agents_{ABxyzA}$ は 4， $Agents_{ABxyzBA}$ は $6 (=4 \times 0.5 \times 3 \times 2 \times 0.5)$ である．これから，リンク AB に関する以下の 1 つの制約式が得られる．

$$4 \times N_{ABCDEA} + \dots + 6 \times N_{ACDEBA} + \dots \leq Limit_{AB} \quad (2)$$

同様に，他の 9 のリンクにおいても，それぞれ 1 つ，計 10 個の制約式が得られる．

すなわち，上記の問題は，24 種類の経路を通るエージェント数をそれぞれ軸とする 24 次元空間において，34 個の超平面で区切られる凸面体と交差する超平面 $\sum_{R \in ROUTES} N_R = K$ の中で最大の K を求める問題と同じになり，以下の解を得ることができる．

- 4.2 節の静的環境モデル： $N_{ABCDEA} = 500000$, $N_{ACEBDA} = 500000$ and $N_R = 0$ ($R \in$ the others).
- 4.3 節の動的環境モデル： $N_{ABCDEA} = 500000$, $N_{ACBEDA} = 500000$ and $N_R = 0$ ($R \in$ the others).

(平成 17 年 1 月 15 日受付)

(平成 17 年 5 月 16 日再受付)

(平成 17 年 6 月 8 日採録)



中山 功一（正会員）

2000 年三重大学工学部機械工学科卒業．2002 年三重大学大学院工学研究科博士前期課程（機械工学）修了．2005 年京都大学大学院情報学研究科博士後期課程（システム科学）修了．2002 年 ATR 人間情報科学研究所学外実習生を経て，2004 年 ATR ネットワーク情報学研究所研究員，現在に至る．京都大学博士（情報学）．



松井 博和

1992 年名古屋工業大学工学部電気情報工学科卒業．1994 年名古屋大学大学院工学研究科博士前期課程（情報工学）修了．1997 年同博士後期課程（電子機械工学）単位取得退学，神鋼電機株式会社入社．1998 年三重大学工学部助手（機械工学科），現在に至る．博士（工学，名古屋大学）．



下原 勝憲

1976 年九州大学工学部情報工学科卒業．1978 年九州大学大学院工学研究科（情報工学）修了，電信電話公社横須賀電気通信研究所入所．NTT コミュニケーション科学基礎研究所社会情報研究部長等を経て，2001 年 ATR 人間情報科学研究所所長，2004 年 ATR ネットワーク情報学研究所所長，現在に至る．1998 年から京都大学大学院情報学研究科客員教授．博士（工学，九州大学）．



片井 修

1969 年京都大学工学部機械工学科卒業．1971 年京都大学大学院工学研究科機械工学第二専攻修士課程修了．1974 年同博士課程単位取得退学，京都大学工学部助手（精密工学科）．1983 年同大学助教授．1994 年同大学院工学研究科教授，1998 年同情報学研究科教授，現在に至る．計測自動制御学会論文賞・著述賞等受賞．工学博士（京都大学）．