論文要旨

エージェント間で直接情報交換ができない分散環境において，エージェントが自分に割り当てられたタスクを達成するために共有資源をどの程度の期間確保するかを決定する問題を考える．このとき長期間確保すればタスクを多く達成できる可能性があるが，他のエージェントのタスク達成を妨害し，結果的にシステム性能の低下を招く．本論文では上記の競合を回避する協調的な制御規則を獲得するために，このような資源共有問題を優先順位のある多基準意思決定問題として取り扱う．すなわち，エージェント間に生じる競合の回避と性能の追求という複数の行動規範に弱いトレードオフを考慮したα-domination 戦略を導入することにより，分散強化学習を用いて協調的な制御規則を獲得することを提案する．提案手法を分散データベースシステムに適用し，協調的な政策の獲得により高いスループット性能が得られることを示す．

論　文

α-domination 戦略に基づく分散強化学習と資源共有問題への応用*

青木　圭†・池田　心‡・木村　元§・小林　重信†

Distributed Reinforcement Learning based on α-domination Strategy and its Application to Shared Resource Problems*

Kei Aoki†, Kokolo Ikeda‡, Hajime Kimura§ and Sigenobu Kobayashi†

In the distributed systems in which information cannot be exchanged directly among agents, we deal with problems of deciding how each agent holds the shared resource. To achieve a lot of tasks greedily, agents tend to attempt to hold the resources for a long term. However the system performance decreases consequentially because it competes with the processing of other agents' tasks. To acquire cooperative policies that avoid above competition, we formulate the shared resource problems to multi-criteria decision making problems with the priority level by using the domain knowledge. We propose autonomous distributed control using distributed reinforcement learning that narrows the choice of action space by using the α-domination strategy based on value functions for the performance and the cooperation. The proposed method is applied to the distributed database systems, and simulation results shows that our method acquires cooperative policies and improves the throughput performance of the system.

1. はじめに

ネットワークで結ばれた分散システム上には，計算機やデータなどの資源が複数存在し，それぞれ不特定多数のエージェントによって共有されている [2]．このような環境の下では，一般に，各エージェントがそれぞれのタスクを処理するために利己的に性能を追求すると競合が生じて，結果としてシステムの性能も各エージェントの性能も低下してしまう状況に陥る．

本論文ではその様な問題の典型的なクラスとして，互いを観測できない複数のエージェントがいくつかの必要な共有資源を排他的に確保してジョブを処理する問題を考える．必要な資源が他のエージェントに既に確保されているとき，それが解放されるのを待つ時間を適切に制御し，処理量を最大化する問題を資源共有問題と呼ぶ．システム性能はエージェントの処理量の総和で与えられ，システムの目的はこの総処理量の最大化である．

本問題の難しさは，各エージェントは他のエージェントの情報を得ることはできないので，それぞれの処理量を最大化するように意思決定を行ったとしても，システムの目的が達成されるとは限らないことにある．例えば，近視眼的な視点では各エージェントは資源に対する待ち時間を長く設定した方が処理量を最大化する上で有利である．しかし，すべてのエージェントが同じ視点で処理量の最大化を追求すると，互いに既に確保した資源を待ちあうデッドロック状態を引き起こしやすい．デッドロック状態に陥ると，いずれかのエージェントが処理をあきらめて資源を解放しない限り，どちらも処理を完了することができないだけでなく，それまでに確保している資源を必要とする他のエージェントの処理をも妨げてしまう．結果的に各エージェントの処理量は低下し，システム性能が損なわれる．このように資源共有問題は各エージェントの利己的な性能追求が競合を生じ，結果的にシ

* 原稿受付　1995 年 8 月 1 日
† 東京工業大学　大学院　総合理工学研究科　Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology; 4259, Nagatsuda-cho, midori-ku, Yokohana city, Kanagawa 226-8502, JAPAN
‡ 京都大学　学術情報メディアセンター　Academic Center for Computing and Media Studies, Kyoto University; Hon-machi, yosida, sakyo-ku, Kyoto, 606-8501, JAPAN
§ 九州大学　大学院　工学研究院　Faculty of Engineering, Kyushu University; 6-10-1, Hakozaki, Higashi-ku, Fukuoka, 812-8581, JAPAN
Key Words: distributed reinforcement learning, multi agent system, α-domination strategy, multiple-criteria decision making, shared resource problems, distributed database systems.

ステム性能に悪影響を及ぼすという難しい課題である．

また，実システムを扱うためには不確実性などの特徴を考慮して制御規則を設定する必要がある．これに対してマルチエージェントシステム (Multi Agent Systems, MAS) という枠組みの下で分散強化学習 (Distributed Reinforcement Learning, DRL) を用いた制御規則の獲得が注目されている [13] が，上記の競合を解決する方法は知られていない．そこで本研究では競合回避のために，待ち時間をなるべく短く設定すればデッドロックが生じにくくなるというヒューリスティクスを用いて，利他的な行動規範である譲歩を導入する．各エージェントの利己的な性能追求と利他的な譲歩という 2 つの行動規範に基づいてシステム性能を向上させる制御規則を獲得するために優先順位のある多基準意思決定問題として資源共有問題を定式化する．各行動規範に対してそれぞれ報酬を設定して価値関数を定義し，強化学習の 1 手法である SARSA 学習を用いて更新する．

2 つの行動規範に基づく意思決定は以下のように行う．まず，各エージェントは任意の状態において 2 つの行動規範の価値関数で構成される空間上に実行できる全行動を並べる．そして，多目的最適化の分野で池田らによって提案された弱いトレードオフによって導かれる Pareto 集合を効率的に求める α-domination 戦略 [6] を用いることにより，有望な行動集合を選別することができる．そのようにして得られた有望な行動集合から任意の探索戦略に従って性能追求の観点で一意に行動を選択することにより，競合回避と性能追求のバランスを実現することが期待できる．このとき，探索行動空間として α-Pareto 集合を利用することにより，全行動空間が大きい場合でも探索を有望な行動集合に絞って行うことができる．本論文では競合回避のための行動規範である譲歩の導入による多基準化と α-domination 戦略による選択と探索の効率化により資源共有問題のシステム性能の向上を目指す．

本論文では資源共有問題の一例である分散データベースシステムのトランザクション処理を実験対象に取り上げる．これをベンチマークとしてモデル化した問題に提案手法を適用して，いくつかの環境でシミュレーション実験を行って獲得した性能を示し，α-domination 戦略に基づく分散強化学習の有効性と有用性を確認する．

以下では 2 章で対象問題として資源共有問題を取りあげてモデル化し，問題の所在を明らかにする．3 章では競合回避のための行動規範である譲歩を導入し，多基準の価値関数を基に α-domination 戦略で有望な行動集合を絞り込む分散強化学習アルゴリズムを提案する．4 章では提案手法を分散データベースシステムのトランザクション処理のベンチマーク問題に適用してその有効性と有用性を確認する．5 章はまとめである．

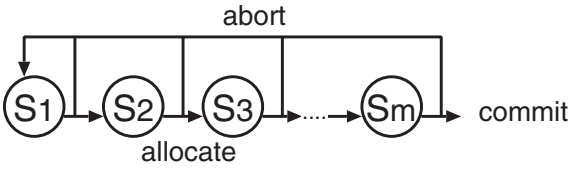


Fig. 1 The flow chart of holding resource

2. 問題設定

本論文では MAS を環境内のエージェントが相互作用することにより何らかの利益を獲得することを目的とするシステムと定義する．システム全体を集中制御できない分散環境において MAS を扱うために，各エージェントが自律的に意思決定を行う自律分散制御の考え方が必要である．そのような MAS において制御規則の獲得を難しくする問題のひとつにシステムとエージェントの利益が相反するという問題がある．このときエージェントの利己的な挙動はシステムの利益と反するため，協調的な制御が同時に要求される．本章ではこのような特徴を有する対象問題クラスとして資源共有問題を取りあげる．以下では資源共有問題をモデル化し，その問題の所在を明らかにする．

2.1 資源共有問題のモデル化

互いを観測できない複数のエージェントがいくつかの必要な共有資源を排他的に確保してジョブを処理する問題を考える．必要な資源が既に確保されているとき，それが解放されるのを待つ時間を適切に制御し，処理量を最大化する問題を本論文では資源共有問題と呼ぶ．

環境 $E=\{N,S\}$ はエージェント集合 N と資源集合 S からなる．各エージェント $n\in N$ はジョブ j_n を保持する．$s_i^n\in S$ は j_n の実行に必要な資源であり，m はその数である．

$$j_n=\{s_1^n,s_2^n,\cdots,s_m^n\} \tag{1}$$

ジョブに必要な資源 s_i^n には順序があるので，Fig. 1 に示すようにエージェント n は s_i^n に対して $i=1,2,\cdots,m$ の順に Fig. 2 の要領で確保を試みる．

- s_i^n が他のエージェントに確保されていなければ s_i^n を排他的に確保し，つぎに s_{i+1}^n の確保を試みる．
- s_i^n がすでに確保されている場合，解放されるまで利用できない．そこで期間 $timeout_n(s_i^n)$ の間待機し，解放されなければジョブを一度あきらめてこれまで

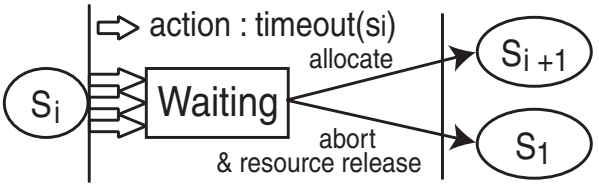


Fig. 2 The process of holding resource

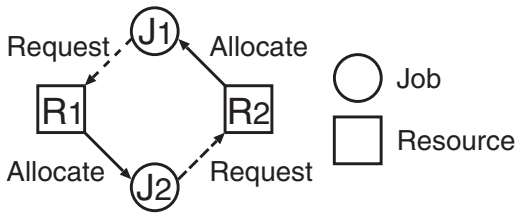


Fig. 3 The dead lock condition

確保した資源を全て解放し，s_1 から再度開始する．必要な資源を全て確保したらジョブ j_n を処理して処理数 $commit_n$ を増やし，新たなジョブ j'_n を開始する．

制御変数は各エージェント n が各必要資源 s_i^n を待つ待機時間間隔であるタイムアウト政策 $\pi_n \in \mathit{\Pi}$ である．

$$\pi_n = \{timeout_n(s_1^n), \cdots, timeout_n(s_m^n)\} \tag{2}$$

本問題では分散環境を考慮し，互いのエージェントの情報を全く得ることができないと仮定する．相互作用の結果として得る自己の処理量を通じてのみ，システムとの関連を知覚できる．これはインターネットやイントラネット環境を通じて共有資源を扱うことが多くなりつつある近年の状況に合致する．したがって，各エージェントはシステム全体を知覚できないため，システムの目的に直接貢献できない．そこで，各エージェントの性能としてそれぞれが処理完了した単位時間あたりのジョブの処理量が与えられ，タイムアウト政策を適切に決定することにより，それぞれ性能向上を目指す．このような各エージェントの性能追求 $\max_{\pi_n} commit_n$ を基に，システムの目的である全エージェントが処理完了した単位時間あたりのジョブの総処理量の最大化 $\max_{\mathit{\Pi}} \sum_n commit_n$ が求められる．

2.2 問題の所在

各エージェントにとってジョブを確実に処理完了するためにはなるべく長く待つタイムアウト政策を選択し，必要資源を確保している他のエージェントの処理が完了するか，処理をあきらめて資源が解放されるまで待つことが有効である．しかし，既に確保されている資源に対して長く待つという政策は，Fig. 3 に示すような互いに確保している資源を要求しあうことにより膠着状態に陥るデッドロック状態を引き起こしやすい．この状態に陥ると，いずれかのエージェントが処理をあきらめて資源を解放しない限り，どちらも処理を完了することができないだけでなく，それまでに確保している資源を必要とする他のエージェントの処理をも妨げてしまう．結果的に各エージェントの処理量は低下し，システムの性能が損なわれることになる．このように資源共有問題は本来システム性能に結びつくはずの各エージェントの利己的な性能追求がデッドロックという競合を生み，結果的にシステムの性能低下を招くという困難な問題点を持つ．

3. 接近法

上述したように資源共有問題では各エージェントが利己的に性能を追求すると競合が発生し，システムの性能を低下させる．したがって性能追求と共に競合を回避する方法が必要であるが，環境中の他のエージェントと情報交換できないために，競合するエージェント同士が明示的に相互作用して競合を検出・回避するような協調制御を行うことは困難である．したがって協調制御を導くためには，分散環境を前提に何らかの情報を与えて競合を回避する必要がある．そこでヒューリスティクスを利用して競合回避のための行動規範として譲歩を導入し，多基準の観点から適切に意思決定を行う方法を提案する．

3.1 競合回避のための譲歩

資源共有問題において競合は一時的なデッドロックが原因で生じる．デッドロックに陥るといずれかのエージェントがジョブを放棄して資源を解放しない限り，どのジョブも処理を完了できない．したがって，競合を回避するためにはデッドロック状態をなるべく早く解消する必要があるが，エージェント間で情報を交換できないため，デッドロックを検出することは容易ではない．

そこでデッドロックの検出を行わずに競合を回避する方法を考える．エージェント n は必要な資源 s_i^n が他のエージェントによって既に確保されている場合，それが解放されるのを時間 $timeout_n(s_i^n)$ だけ待つが，これが短いほどデッドロックになりにくいというヒューリスティクスを利用して以下のように譲歩する行動規範を定義する．

$$\min_{\pi_n} \sum_i timeout_n(s_i^n) \tag{3}$$

しかし，各エージェントができるだけ競合が起きないように譲歩しすぎてタイムアウト時間が短い政策を選択すれば，競合回避は促進されるものの，当然ながら必要資源の確保が難しくなり，各エージェントの性能が損なわれ，結局システムの性能も得られない．そのため，この行動規範は副次的なものとして扱う必要がある．

各エージェントにとって性能追求の行動規範のみに従うと競合を生じ，譲歩の行動規範のみに従うと性能が低下することを考慮して，双方のバランスをとる優先順位付きの多基準意思決定問題として資源共有問題を扱う．すなわち，主目的を性能追求 $\max_{\pi_n} commit_n$，副目的を譲歩 $\min_{\pi_n} \sum_i timeout_n(s_i)$ として，システム性能の最大化 $\max_{\mathit{\Pi}} \sum_n commit_n$ を目指す．

3.2 分散強化学習を用いた自律分散制御

競合のために各エージェントの単位時間あたりの処理量には不確実なノイズが含まれることは避けられない．この場合，政策の長期的な評価を行える強化学習を用いて制御政策を獲得することは有望である [14] ．

本論文では性能追求と譲歩の行動規範に関する報酬を

それぞれ設定し，各エージェントにベクトルで与える．

$$\mathbf{r_n} = \begin{cases} commit_n, & \text{for } Q_c^n \\ \sum_i timeout_n(s_i), & \text{for } Q_t^n \end{cases} \quad (4)$$

価値関数をそれぞれ定義 ((5) 式) し，SARSA 学習 [14] を用いてそれぞれ更新 ((6) 式) する．

$$\mathbf{Q}^n(x,a) = \{Q_c^n(x,a), Q_t^n(x,a)\} \quad (5)$$
$$\mathbf{Q}^n(x,a) = (1-\lambda)\mathbf{Q}^n(x,a) + \lambda(\mathbf{r_n} + \gamma\mathbf{Q}^n(x',a')) \quad (6)$$

ここで Q_c は性能追求のためのジョブの処理量に関する価値関数，Q_t は譲歩のための政策の待ち時間に関する価値関数で，λ は学習率，γ は割引率である．x は観測した状態，a は選択した行動，x' は遷移後の観測状態で，a' は x' で選択する行動である．

3.3 α-domination 戦略に基づく行動選択

本論文では 2 つの行動規範に基づく意思決定において，まず行動空間全体を対象に多基準の観点から有望な行動集合 (Feasible Action Set, FAS) を選別する．そして得られた FAS を対象に性能向上の観点から任意の探索戦略を用いて行動を一意に決定する方法を提案する．

Fig. 4（左）に示すように 2 つの行動規範に関する価値関数で構成される空間に，任意の状態において選択できる全行動をプロットする．このとき，多目的最適化の Pareto 集合の観点から，ある程度有望な行動集合を選別することができる．しかし，”a”と”b”のついた行動に着目すると，行動 b に対してわずかに性能に関する Q_c が高いけれども，譲歩に関する Q_t がかなり低い行動 a も Pareto 集合に含まれてしまうことがわかる．行動 b に対して性能向上は少ししか期待できないにもかかわらず，競合を引き起こすリスクが大きい行動 a は望ましくないが，従来の性能追求のみの行動選択ではこのような行動が優先的に選択され，競合が引き起こされやすい．

そこで α-domination 戦略 [6] を FAS の選択に用いることにする．α-domination 戦略は多目的な要素に対して，dominance 判定に弱いトレードオフを導入する手法で，dominance 抵抗解を Pareto 集合から効率よく排除できる有効性が示されている．弱いトレードオフ関係は次式で表現される．

$$\alpha_{tc} \leq \frac{\Delta Q_c}{\Delta Q_t} \leq \frac{1}{\alpha_{ct}} \quad (7)$$

ここで ΔQ_c と ΔQ_t は Q_c と Q_t の変化量，α_{tc}，α_{ct} はトレードオフパラメータである．本論文では単純に $\alpha = \alpha_{tc} = \alpha_{ct}$ とした．

各価値関数間のトレードオフ比が厳密に設定できないことはしばしば言われることだが、その上下限を設定することは比較的容易である．例えば，ドルとユーロの為替レートを固定して考えることはできないけれども，100 ドル＋100 ユーロの価値と 1 ドル＋105 ユーロの価値は常識的に差があることが明らかであり，共に dominate されないからといって同等に扱うことは無駄である．このとき，1 ユーロは少なくとも 0.1 ドルよりは価値があるといった知識を利用して，1 ユーロ＞0.1 ドル，1 ドル＞0.1 ユーロといった最低限の価値関係が記述できるとき，これを弱いトレードオフとして定義することにより，有用な候補だけを選択肢に残すことができる [6] ．

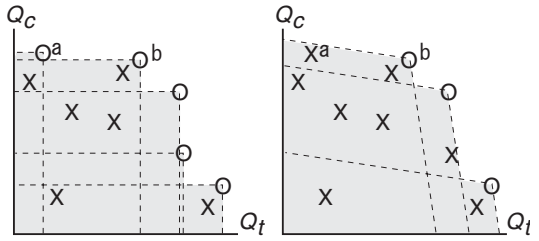


Fig. 4 The Pareto action set (left) and the α-Pareto action set (right) in any state. FAS consists of actions that are described as ∘ which dominates ×. The gray part represents the dominate area.

この弱いトレードオフ関係を用いて Fig. 4(右) に示すように「Q_c の観点で行動 b が行動 a に少々劣っていても、Q_t の観点で行動 b が行動 a におおいに勝る」場合は行動 a を Pareto 集合から排除することができる．この場合を行動 b が行動 a を α-dominate するといい，こうして得られる集合を α-Pareto 集合という [6] ．本論文では α-Pareto 集合を FAS とし，優先順位に基づき Q_c を最大化する行動を ϵ-greedy 戦略などの探索戦略を用いて FAS から一意に決定する．

また α-domination 戦略は探索効率を向上させる効果も期待できる．ϵ-greedy 戦略などのランダム性を用いた探索戦略は単純さなどから良く用いられるが，連続値の行動を離散化するなどして行動空間が大きい場合などでは，行動空間全域をランダムに探索するのは効果的ではない．また，他のエージェントとの競合を考慮すると，ランダムなどを利用した探索的な選択によって他のエージェントの性能に大きい影響を与える行動を実行することは望ましくない．このとき 2 つの行動規範の価値関数からあるトレードオフの下で選別された FAS を探索空間として利用することにより，探索による競合の発生と無駄な探索を低減し，効率よく学習することが期待できる．

以上が競合回避のために譲歩という行動規範を定義した優先順位付多基準意思決定問題を対象とする α-domination 戦略に基づく分散強化学習アルゴリズムである．Fig. 5 にその概要を示す．

3.4 関連研究

環境を共有する意思決定エージェントで構成する MAS はゲーム理論の分野で古くから研究が行われ，特に Nash 均衡という観点から多くの有益な分析や知見が得られている [10] ．しかしゲーム理論の観点では環境の完全観測性が前提にあり，資源共有問題のように情報が限定された環境を扱うことはできない．また，Schneider[12] らや

(1) 価値関数 $\mathbf{Q}$ を初期化
(2) 状態 x を観測
(3) 意思決定
- $\mathbf{Q}(x,a),\forall a$ に関して α-Pareto となる行動集合を FAS に登録（Fig. 4 右の $\circ$ の行動）
- Q_c に関して FAS から探索戦略（ϵ-greedy 選択など）により一意に行動 a を決定

(4) 行動 a を実行して報酬 $\mathbf{r}$ と次状態 x' を観測
(5) 意思決定 (3) を行い，行動 a' を決定
(6) (6) 式で各価値関数を更新
(7) $x=x'$，$a=a'$ として，(4) に戻る

Fig. 5　The distributed reinforcement learning algorithm based on α-domination strategy

Guestrin[4] らは限定された通信を行える MAS において協調制御を獲得できる有効な手法を提案しているが，完全な分散環境である資源共有問題などには適用できない．

これ対して協調行動を学習する政策勾配法 [11] や分散強化学習 [5] を Nash 均衡の観点で捉える研究がある．これらの研究は分散環境に適用できるが利己的なエージェント間に生じる競合とシステム性能低下の問題を想定するものではない．我々の知る限りでは MAS におけるこの困難な問題を直接扱うことができる効果的な分散強化学習手法は知られていない．

また MAS ではないが多目的性の取り扱いについては辞書的順序を利用した手法 [3] や目標領域に平均報酬ベクトルを指向する強化学習 [7] がある。これらの手法は意思決定そのものを多目的の観点から行うものであるが，本論文は優先順位のある行動規範に基づく多基準性の下で，行動選択の候補を効率的に絞り込むものである．

4. 分散データベースシステムへの応用

本章では資源共有問題のベンチマークとして分散データベースシステム (Distributed Database System, DDB) のトランザクション処理をモデル化した問題に提案手法を適用し，有効性と有用性を示す．

4.1 トランザクション処理とデッドロック

トランザクション処理とは，並列処理において個別に行われる動作の集合からなる計算処理をいい，統一概念として原子性，一貫性，分離性，持続性という ACID 特性が定義されている．各トランザクションはジョブが与えられると処理を開始し，処理の終了をコミット，途中で中止することをアボートという．アボートした処理は全て元通りに戻さなければならないなどデータの一貫性を維持するために参照や書換えが行われるデータをロックする方法が広く使用されている [1]．あるトランザクションがデータをロックすると，他のトランザクションはそのデータにアクセスできなくなりロックが外される

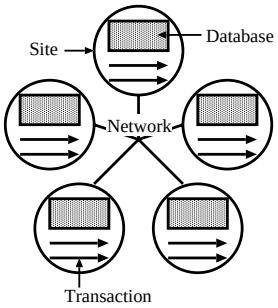


Table 1　The parameters

パラメータ	値
データベースサイト数	5
コミットまでに必要な資源数	8
並列動作トランザクション数	10〜200
CPU 処理時間	15 ms
I/O 処理時間	35 ms
他サイトへのリクエスト率	40 %
自サイトへのメッセージ遅延	1 ms
他サイトへのメッセージ遅延	100 ms

Fig. 6　The model of DDB

まで待機状態に入る．

このとき資源を一定時間待ったトランザクションをデッドロックとみなし，強制的にアボートさせる方法をタイムアウトといい，単純かつ実用的なため広く用いられている．しかし，動的で不確実な環境において良い性能を引き出すタイムアウトの設定は非常に難しい [15]．

システムの目的は各エージェントが単位時間あたりにコミットした処理量を表すスループットの総計を最大化することであり，本論文ではこのスループットの総計をシステムの性能として評価する．

4.2 実験設定

トランザクション自体の処理は詳細を観察することはできない．そこで本論文では Fig. 6 に示す DDB の各サイトをエージェントとし，そこから生成されるトランザクションは同じタイムアウト政策に従うものとする．

トランザクションはコミットまでに複数の様々なデータの確保を必要とするため，(2) 式をそのまま政策とすることはできない．そこでトランザクション処理の特性を利用して確保データ数に関する関数としてタイムアウト政策を定義する [9]．a, b は政策パラメータである．

$$timeout(s_i) = a * b^{i-4}, \qquad i = 1,2,\cdots,m. \tag{8}$$

各サイトは政策パラメータ (a,b) を決定し，一定期間環境にトランザクションを生成してスループットを受け取る．

実験では Table 1 に示すパラメータで DDB を設定し，各サイトの保持データ数が 100 の均一環境と，ひとつのサイトが 100 で他の 4 サイトが 300 の不均一環境に対して適用した．これは限界性能を計算するために単純な設定を用いている．また，並列動作トランザクション数 (Multi Programming Level, MPL) を固定した定常環境で 10 から 200 まで実験した．MPL が大きいほど環境が混雑しデッドロックが起こりやすくなる．本課題の詳細は参考文献 [8] に記述されている．

実験期間は全体で 1.5×10^9ms，1.0×10^5ms 間隔で意思決定し，学習終盤の 1.0×10^6ms 間の性能で評価を行う．予備実験に基づき強化学習パラメータは学習率 0.1，割引率 0.9，α-domination 戦略の α は均一環境では MPL が 10 から 100 で 0.005，110 から 200 で 0.001 を用いた．不均一環境では全 MPL で 0.0001 を用いた．政策パラメー

タは a は 500 間隔で (500,7500) の範囲を 15 段階，b は 0.05 間隔で (1.05,1.5) の範囲を 10 段階に離散化した．したがって各サイトは行動数 150 から意思決定し，システム全体ではサイト数 N に対して 150^N の組合せとなる．Fig. 7 から Fig. 10 は 10 試行の平均値をプロットした．

4.3 比較手法

従来運用に用いられる固定値のタイムアウト政策 (Fix Best) とスループット最大化を行う Q-learning 政策と比較を行う．Fix Best 政策は確保データ数に関係なく固定値のタイムアウト時間をとる最もよく使われる政策であり，Fig. 7 では各 MPL に関して固定タイムアウト時間を (500,20000) の範囲で離散的に全探索して得られた最良値を求め，それをプロットした．

Q-learning 政策は各エージェントが性能追求の行動規範である処理数最大化 $\max_{\pi_n} commit_n$ のみで学習を行い，競合は考慮しない．各設定は提案手法と同じであるが，譲歩の行動規範を用いずに α-domination 戦略ではなく全行動空間から ϵ-greedy 戦略で意思決定すること，価値関数の更新を Q-learning[16] で行う点が異なる．

また，限界性能を示すために，全サイトを一括した集中管理の下で政策パラメータを離散的に全探索して得られた組合せ最良解も示す．このために全探索のための予備実験では，均一環境では全てのサイトに同じ政策を共有させ，不均一環境では保持データ数が 100 のサイトと 300 のサイト毎にそれぞれ政策を共有させた．このように同じパラメータのエージェントが同一の共有政策を用いることにより，システム全体で決定する政策パラメータ空間をかなり小さくし，その中で網羅的に全探索して組合せ最良解を求めた．これは分散環境では実行不可能な方法であるが獲得性能の理論的な限界 (Theoretical Limit) を示すことができる．

α-domination 戦略の有効性を示すための実験では，提案手法で用いた譲歩の行動規範に関する報酬を性能追求の行動規範に関する報酬と α をトレードオフ比のパラメータと見なして重み付け線形和することにより単目的化する Q-learning (Linearly Weighted Sum Q, LWS-Q) との比較を行う．トレードオフパラメータ α に関しては性能追求と譲歩の優先関係から (7) 式の左式の関係が重要であることを考慮している．

$$r_n = (1-\alpha)commit_n + \alpha \sum_i timeout_n(s_i) \qquad (9)$$

4.4 実験結果と考察

均一環境において各 MPL 毎に得られたスループット性能を Fig. 7 に示す．従来運用でよく用いられている固定タイムアウト政策は最良の値を選択しても十分な性能が得られていないことがわかる．すなわち，タイムアウト時間を保持資源数に関して設定する (8) 式で示される政策表現の妥当性が，従来の固定タイムアウト政策より性能を向上できることから確認された．提案手法はほぼ全ての MPL において処理量最大化のみを追求する Q-learning を上回り，限界性能である組合せ最良解と同等の性能を獲得した．以上より，譲歩の行動規範の導入と α-domination 戦略によって探索行動空間を絞り込む分散強化学習の有効性が確認された．

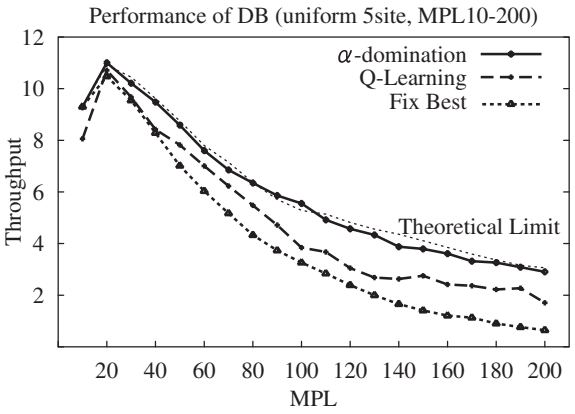


Fig. 7 The throughput performance at MPL10-200 in the uniform DDB

均一環境において MPL50 の時の学習曲線を Fig. 8 に示す．学習序盤における性能差は，α-domination 戦略により早い段階から多くの価値の低い行動が dominate されるためと考えられる．報酬は他のサイトの政策により大きく変動するため，学習序盤の探索的な行動によって競合が起きやすいが，α-Pareto 集合である FAS に有望な行動が登録されることでこの競合が早期に解消できていると考えられる．以上より，アルゴリズム以外は同設定の実験で学習速度の違いが観察されることから，提案手法の探索の効率化が有望であることが確認された．

不均一環境において各 MPL 毎に得られたスループット性能を Fig. 9 に示す．極端に混雑するサイトと他の 4 サイトとの性能はトレードオフ関係にある．限界性能を

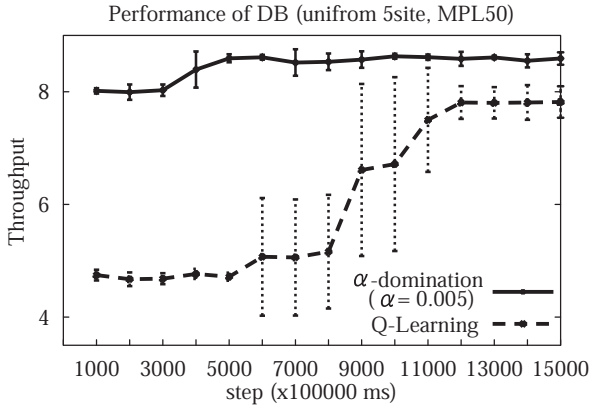


Fig. 8 The learning curve at MPL50 in the uniform DDB. Each point is the average throughput performance for 10^7 ms (10 trials). The error bar represents the variance.

表す組み合わせ最良解は混雑するサイトをある程度犠牲にすることで全体の性能を向上させる政策群であり，混雑するサイトのスループットは他のサイトの約 1/6 程度となる．提案手法ではそのような政策群の他に，ある程度揃った政策群が学習される試行が確認された．このときスループットは他のサイトの約 1/3 程度となる．これはスループットの総計を観測できないことと実験パラメータなどの設定が同一なためであると考えられる．これらのことから理論的な限界性能には平均で至っていないが，処理量最大化のみを追求する Q-learning よりも性能は良い．

なお，不均一環境では競合の問題の他に，システム全体の性能の最大化とサイト間の平等性とのバランスといった質の異なる問題がある．これは一般に運営者のポリシーに依存するものであるため，どちらを優先させるかを決めることは困難なので本論文では扱わなかった．

4.5 α-domination 戦略の効果

α-domination 戦略において α は性能追求と譲歩の間の弱いトレードオフ関係を表すパラメータで，獲得性能に影響する．本問題において MPL100 程度の場合は経験的にスループット性能を 1 向上させるために 200ms から 300ms 程度までの待ち時間 $\sum_s timeout(s)$ の増加は許容できるが，それ以上は費用対効果が悪いことがわかっている [8] ．また待ち時間を増加させる場合に処理量もかなり増加しないと許容しないといったように譲歩しすぎれば，性能追求がないがしろになってしまうと考えられる．したがって，α が大きすぎると主目的が達成されずに性能が急速に悪化し，3.0×10^{-3} から 5.0×10^{-3} 程度で適切なトレードオフ関係が得られて性能が良くなり，小さくなるにつれて性能が悪化すると予想される．

均一環境において MPL100 の時，α を変化させたときのスループット性能を Fig. 10 に示す．予想通り適切なトレードオフ比である 3.0×10^{-3} から 5.0×10^{-3} で良い性能を示した．α を大きく設定することは譲歩しすぎることになるので，そのような設定は本来用いないが，実

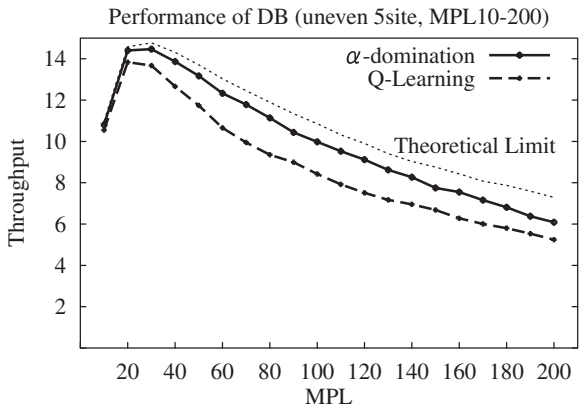


Fig. 9 The throughput performance at MPL10-200 in the uneven DDB

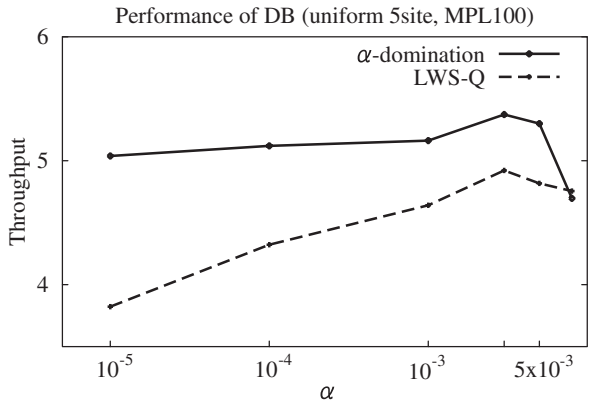


Fig. 10 The relation between α and throughput performance (the uniform DDB, MPL100)

験では全サイトが短いタイムアウト政策を選択するようになり，急激な性能悪化が見られた．α を小さくしていくにつれて競合のリスクは許容され，α-Pareto 集合に含まれる行動数は増加していくはずだが，相互作用を介して競合しづらい行動が結果的に評価されて FAS に選択され易い傾向があり，性能の低下は徐々に起こることが確認された．

α を適切に調節することは突き詰めれば適切なトレードオフ比を求めることになる．そこで，単純に比較することはできないけれども α をトレードオフ比とした LWS-Q と提案手法の結果を並べ，その違いを考察する．

Fig. 10 では Fig. 8 の学習曲線で述べたことと同様に，LWS-Q は大きな行動空間での探索の困難さなどから全体的に性能が低いことが確認された．特に α が小さいときは性能追求が重視されて競合が発生しやすく，性能が低下していく様子が観察されたという点で提案手法の結果と異なることがわかった．

以上のことから提案手法は LWS-Q と同様にトレードオフを用いるものの，どの程度譲歩するかという観点で適度に α を選択すれば LWS-Q よりも良い性能をロバストに獲得できることが確認された．

5. おわりに

資源共有問題を対象に，競合回避のための譲歩という行動規範の導入と α-domination 戦略に基づく分散強化学習手法を提案した．資源共有問題はエージェントの利己的な性能追求がシステムの性能を低下させる競合の問題から，分散環境ではシステム性能を向上するために協調的な制御が同時に要求されるが，優先順位付多基準意思決定問題にモデル化して提案手法を適用することにより，高いシステム性能を獲得できることを DDB の課題で実験的に示した．

本手法は競合回避を実現する行動規範として譲歩に限定するものではない．競合回避に貢献する任意の報酬が設計できれば適用することができる．このとき，完全な競合回避のための行動規範を設定することは必ずしも要

求されず，ある程度行動を絞り込むヒントとなれば良いと考えられる．また，互いに情報交換できない MAS におけるエージェントの利己的な性能追求による競合とシステム性能の低下の問題は，資源共有問題に限らず様々な実問題に存在し，それらの制御規則の獲得を困難にする．例えばエージェントが何らかの制約を持つ MAS であれば，システム性能と制約充足の間には少なからず競合が存在する．提案手法はこのような課題に広く拡張できる可能性があると考えている．また，提案手法は２つ以上の行動規範に基づく多基準意思決定にもそのまま適用することができる．例えば複数の目的や制約充足などを扱うより複雑な問題への適用が考えられる．

今後は上記のことを考慮し，本論文で扱わなかった動的環境を扱う課題に提案手法の適用を考えている．

参 考 文 献

[1] P. Bernstein and E. Newcomer: *Principles of Transaction Processing*, Morgan Kaufmann Publishers (1997) (「トランザクション処理システム入門」，大礒，小野沢，木下，中山，早瀬(訳)．日経 BP (1998))

[2] S. Fujita: A quorum based k-mutual exclusion by weighted k-quorum systems; *Information Processing Letters*, Vol. 67, No. 4, pp. 191–197 (1998)

[3] Z. Gabor, Z. Kalmar and C. Szepesvari: Multi-criteria reinforcement learning; *Proceedings of the 15th International Conference on Machine Learning*, pp. 197–205 (1998)

[4] C. Guestrin, M. Lagoudakis and R. Parr: Coordinate reinforcement learning; *Proceedings of 19th International Conference on Machine Learning*, pp. 227–234 (2002)

[5] J. Hu and M. P. Wellman: Experimental results on Q-learning for general-sum stochastic games; *Proceedings of the 17th International Conference on Machine Learning*, pp. 407–414 (2000)

[6] K. Ikeda, H. Kita and S. Kobayashi: Failure of pareto-based MOEAS: Does non-dominated really mean near to optimal?; *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 957–962 (2001)

[7] S. Mannor and N. Shimkin: The steering approach for multi-criteria reinforcement learning; *Advances in Neural Information Processing Systems 14*, pp. 1563–1570 (2002)

[8] 大金: 分散データベースにおける協調機構の実現; 東京工業大学知能システム科学専攻修士学位論文 (2003)

[9] 大金，木村，小林: 分散データベースにおける協調機構の実現; *Proceedings of 30th SICE Symposium on Intelligent Systems*, pp. 73–78 (2003)

[10] M. J. Osborne and A. Rubinstein: *A Cource in Game Theory*, The MIT Press (1994)

[11] L. Peshkin, K. Kim, N. Meuleau and L. Kaelbling: Learning to cooperative via policy search; *Proceedings of 16th Conference on Uncertainty in Artificial Intelligence*, pp. 489–496 (2000)

[12] J. G. Schneider, W. K. Wong, A. Moore, and M. Riedmiller: Distributed value functions; *Proceedings of the 16th International Conference on Machine Learning*, pp. 371–378 (1999)

[13] P. Stone and M. Veloso: Multiagent systems: A survey from a machine learning perspective; *Autonomous Robots*, Vol.8, No.3 pp. 345–383 (2000)

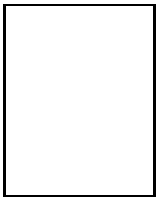
[14] R. S. Sutton and A. Barto: *Reinforcement Learning: An Introduction*. A Bradford Book The MIT Press, (1998)

[15] 魚田, 小碇: データベース, 日科技連 (1993)

[16] C. J. C. H. Watkins and P. Dayan: Technical Note: *Q*-Learning, *Machine Learning 8*, pp. 279–292 (1992)

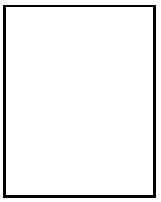
著 者 略 歴

青木 圭（あおき けい）



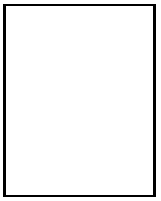
2002 年東京工業大学大学院知能システム科学専攻修士課程修了．同年４月博士課程在籍中．マルチエージェント，強化学習，協調システムに関する研究に従事．

池田 心（いけだ こころ）



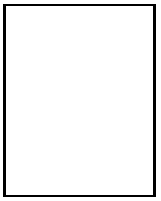
1993 年 3 月，東京大学理学部数学科卒業．同年４月，東京工業大学大学院総合理工学研究科修士課程入学．2000 年 10 月，同博士課程進学，2003 年 3 月，同修了．博士（工学）．同年 4 月京都大学学術情報メディアセンター助手，現在に至る．主に最適化，高度な意思決定システム，ゲームの研究に従事．

木村 元（きむら はじめ）



1997 年東京工業大学大学院知能科学専攻博士課程修了．同年４月日本学術振興会ＰＤ研究員．1998 年 4 月，東京工業大学総合理工学研究科助手．2004 年 4 月，九州大学工学研究院助教授．現在に至る．人工知能，特に強化学習に関する研究に従事．

小林 重信（こばやし しげのぶ） (正会員)



1974 年東京工業大学大学院経営工学専攻博士課程終了．同年４月，同大学工学部制御工学科助手．1981 年８月，同大学大学院総合理工学研究科助教授．1990 年８月，教授．現在に至る．問題解決と推論制御，知識獲得と学習などの研究に従事．