マルチエージェント強化学習における報酬配分の自律的学習

大分大学　　○柴田 克成，杉坂 政典
東京工業大学　伊藤 宏司

Autonomous Learning of Reward Distribution in Multi-Agent Reinforcement Learning

○Katsunari Shibata and Masanori Sugisaka,　Oita University　　shibata@cc.oita-u.ac.jp
Koji Ito,　Tokyo Institute of Technology

Abstract: The autonomous learning of reward distribution between two agents has been proposed. By this learning, each agent who learns its action by reinforcement learning decides how much reward the agent gives to the other one. In this paper, conflicts in multi-agent system are classified, and the condition, under which the reward distribution is required, is introduced at first. Then the reward distribution learning is extended to more than two-agent case. Finally, this algorithm is applied to a three-agent problem, and some interesting strategies among the agents could be observed.

1．まえがき

近年､マルチエージェントシステムにおける利害の衝突回避や協調行動の創発的獲得に対し、強化学習の有効性が示されている。このとき、強化信号（以後報酬と呼ぶ）は各エージェントに配分されるが、一般的に、その配分割合は学習前に決定される[1][2]。しかし、適切な配分は問題によるところが大きいため、問題に関する事前知識がないと決定することは困難である上、あらかじめ決定することで、強化学習が持つ自律性、適応性といった優れた能力を阻害する可能性が大きい。

そこで､筆者らの一部は、２エージェントシステムにおいて、各エージェントが、強化学習で自らの行動を学習するだけでなく、自分が得た報酬を他のエージェントに分配する割合を学習によって獲得する方法を提案し、２エージェントの競合問題に適用した[3]。

本稿では､まず､マルチエージェント強化学習における利害衝突の強さを定義し、報酬分配が必要な問題を明確にする。そして、報酬分配の学習を３エージェント以上の場合に拡張し、３エージェントの問題に適用した結果を報告する。

2．利害衝突の分類

本章では､マルチエージェントシステムにおける利害衝突の強さを、衝突回避のために譲歩したときの報酬の大きさに基づいて分類する。ただしここでは、報酬は、システムとして得られるのではなく、各エージェントごとに個別に得られるものとする。

動的計画法(Dynamic Programming)に基づく強化学習では、同じ報酬でも短時間に得られた方が良いということも考慮し、時刻 t での理想的な状態評価値 $V^*(t)$は、将来得られる報酬 r と割引率γから

$$V^*(t) = \sum_{i=1}^{\infty} \gamma^{i-1} r(t+i) \tag{1}$$

と表される。ここで、$V_{conflict}$ を衝突(コンフリクト)がある場合、$V_{altruistic}$ を他のエージェントに譲歩した場合、$V_{selfish}$ を他のエージェントが譲歩した場合のそれぞれの状態評価値とする。そして、譲歩したエージェントの状態評価値が、譲歩されたエージェントの状態評価値と同じ場合を「弱い衝突」とする。つまり、

$$V_{selfish} = V_{altruistic} > V_{conflict} \tag{2}$$

と書かれる場合である。「中間の衝突」は、

$$V_{selfish} > V_{altruistic} > V_{conflict} \tag{3}$$

の場合であり、譲歩したエージェントの状態評価値は、譲歩されたエージェントより小さいが、衝突が続く場合よりは大きい。最後のケースは、譲歩したエージェントの状態評価値が、衝突が続く場合の評価値と同じかそれ以下の場合である。これを「強い衝突」と呼び、

$$V_{selfish} > V_{conflict} \geq V_{alturistic} \tag{4}$$

と表すことができる。たとえば、複数のエージェントが協力して荷物を運んで、その報酬を一つのエージェントだけがもらった場合も広い意味で利害の衝突と考えることができるが、この場合は「強い衝突」に分類される。

「弱い衝突」および「中間の衝突」の場合、譲歩しても､衝突し続ける場合よりも得られる報酬が大きいため、個々のエージェントの利益に基づいて学習を行うことで利害の衝突は回避できる[4]。しかし､「強い衝突」の場合、譲歩したエージェントにとってはメリットがないため、譲歩することを学習できない。このような場合、報酬の分配が必要となる。しかし、報酬を分配した際に、各エージェントの状態評価値が、衝突時よりも大きくなければならない。そのためには、全エージェントの状態評価値の平均が衝突時より大きくなることが必要条件となる。２エージェントの場合の式は以下のようになる。

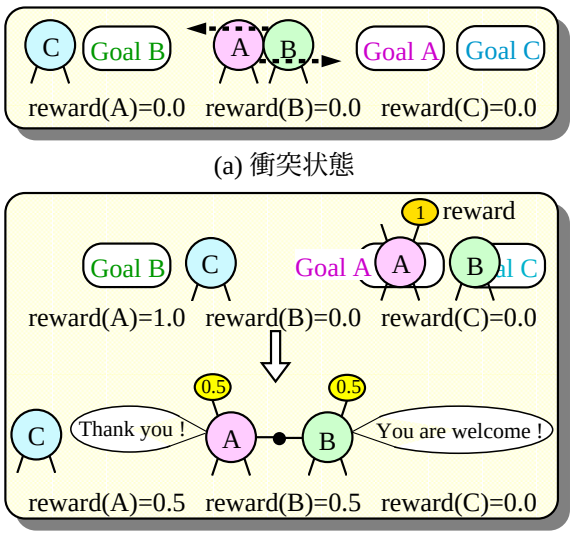
$$V_{selfish} + V_{alturistic} > 2V_{conflict} \tag{5}$$

3. 学習の原理とアルゴリズム

本稿では、「強い衝突」に焦点を当て、学習によって自律的に報酬分配を獲得させる。報酬の分配は、一見利他的に見える。しかし、前述の必要条件が成り立っていれば、少なくとも完全に均等に分配すれば、全エージェントがコンフリクト状態よりも状態評価値が良くなるため、利己的な評価基準によって報酬の分配を学習することができる。これを、例を用いて簡単に説明する。

図 1 (a)のように、A, B, C の 3 つのエージェントがそれぞれのゴール A, B, C を目指していたとする。この場合、エージェント A と B はコンフリクト状態にあり、お互いに譲らないため、両エージェントともにゴールに到達できない。しかし、ここで、図 1 (b)のように B がたまたま道を譲り、A がゴールし、報酬 1 を獲得したとする。このままでは、B は報酬をもらえないため、道を譲らずに A が道を譲るのを待った方が得ということになり、道を譲るという行動を学習することはできない。したがって、学習が進むにつれ、逆に、お互いに道を譲らないことを学習し、ますますデッドロック状態から抜け出すことができなくなる。ところが、A が、もらった報酬の半分を B にあげたとすれば、デッドロック状態で報酬が得られないことと比較すれば、A, B ともに得ということになる。つまり、A にとっては、B に報酬を分配すると短期的には損になるが、長期的には得ということになる。しかし、A が C に報酬を分配しても、見返りはなく、自分の報酬が減少するだけであるので、A が C に報酬を分配することは学習しない。

そこで、エージェント i から j への報酬分配率 $dist_{ji}$



(a) 衝突状態

(b) 報酬を分配した場合

図 1　報酬分配を利己的な評価関数で学習できる原理

を乱数を用いて変化させ、その後の N 試行の間固定させる。そして、過渡状態を除くため、その後半 $N/2$ 試行の間に得られる強化信号 r とステップ数 $step$ の総和を

$$total\_rein_j = \sum_{n=N/2+1}^{N} \sum_{i=1}^{A} dist_{ji} r_i(n) \quad (6)$$

$$total\_step = \sum_{n=N/2+1}^{N} step(n) \quad (7)$$

ただし、A: エージェントの数

と計算する。本稿では、N=1000 とした。そして、(1)式との整合性を考慮し、エージェント j に対し

$$R_j = total\_rein_j \, \gamma^{total\_step} \quad (8)$$

という長期的な評価を導入し、各エージェントごとに、分配率を変化させる前後でこの値を比較する。そして、増加した場合は加えた値を新たな分配率とし、そうでない場合は加える前の値に戻す。この時、分配率は、常に

$$\sum_{j=1}^{A} dist_{ji} = 0 \quad (9)$$

という関係を満たす必要があるので、分配値の変化量は、

$$\Delta dist_{ji} = rnd_{ji} - rnd_{(j+1)\%A,\, i} \quad (10)$$

とした。また、分配値 $dist$ が 0 より小さくなった場合は、その分を他のエージェントへの分配値に均等に分配した。

ここでは、各エージェントは、エージェント 0,1,2.. の順に 1 ステップ 1 エージェントが行動するものとし、状態遷移は決定論的とした。学習は、基本的には Q-learning に基づく。例としてエージェント j=0 の学習則を示す。まず、時刻 t でエージェント j の行動後の状態評価値 $V_j(s_j(t+1))$を、次の自分の順番までに他のエージェントがゴールする確率 $Popp_j$、他のエージェントがゴールしたときの期待報酬 $\check{r}opp_j$、および、他のエージェントが行動後の最大 Q 値の期待値 $maxQ_j$ を用いて、

$$V_j(s_j(t+1)) = Popp_j(s_j(t+1))\, \check{r}opp_j(s_j(t+1)) + (1 - Popp_j(s_j(t+1)))\, \gamma^{A-1}\, maxQ_j(s_j(t+1)) \quad (11)$$

と計算する。ここで、αを学習係数として

$$Popp_j(s_j(t+1)) \leftarrow (1-\alpha)\, Popp_j(s_j(t+1)) + \alpha \quad \textit{if another agent reaches its goal}$$
$$\leftarrow (1-\alpha)\, Popp_j(s_j(t+1)) \quad \textit{otherwise} \quad (12)$$

$$\check{r}opp_j(s_j(t+1)) \leftarrow (1-\alpha)\, \check{r}opp_j(s_j(t+1)) + \alpha\, \gamma^{i-1}\, dist_{ji}\, r_i(t+i) \quad \textit{if the agent i reaches its goal} \quad (13)$$

$$maxQ_j(s_j(t+1)) \leftarrow (1-\alpha)\, maxQ_j(s_j(t+1)) + \alpha \max_k (Q_j(s_j(t+A), a_k) \quad (14)$$

とし、この V を用いて以下のように Q 値を学習する。

$$Q_j(s_j(t),a(t)) \leftarrow (1-\alpha)\, Q_j(s_j(t),a(t)) + \alpha\{dist_{jj} r_j(s_j(t),a(t)) + \gamma V_j(s_j(t+1))\} \quad (15)$$

4. シミュレーション

ここでは、3エージェントのシミュレーション結果を示す。図5の左上の図に示すように、6つのマス目を持つ円の環境に3つのエージェントを配置し、それぞれちょうど反対側にゴールを配置した。最初に動くエージェントは、各試行ごとに、A, B, C と順番に変化させ、その後の動く順番は A の次は B、B の次は C、C の次は A とした。各エージェントは、"時計回りに動く"、"反時計回りに動く"、"動かない"の3つの行動から一つを、Q 値を用いたボルツマン分布にしたがって確率的に選択した。ただし、隣に他のエージェントがいる場合には、そちらに動く行動を選択しても移動できない。ゴールには、右回りでも左回りでも到達することは可能であるが、いずれの場合も、他のエージェントの協力が必要である。割引率γ=0.96 とし、報酬 r=1.5 とした。1回のシミュレーションで 500000 回の試行を行った。行動選択時の温度は学習の前半で 1.0 から 0.01 に徐々に下げ、後半は 0.01 で固定した。分配率に加える乱数も学習全体を通して、±0.05 から±0.005 に徐々に下げていった。乱数系列を変えて 100 回のシミュレーションを行った。学習前の初期報酬分配率は、自分自身に 1.0、その他に 0.0 とした。この問題では、ゴール到達までの最小ステップ数は 7 であり、これを実現するには、最初に動くエージェントがゴールしなければならない。しかしながら、実際には確率的な要素があるため、完全に 7.0 にはならない。

最後の 1000 回の試行でのゴールまでの平均ステップ数を観察した。100 シミュレーションのうち、最大は 11.41、最小が 7.26、平均が 8.45 であった。ゴールに到達したエージェントの報酬分配率を見ると、平均で自分自身に 0.60、そのエージェントがよく動く方向にいるエージェントに対して 0.21、その他のエージェントに 0.19 と、進行方向のエージェントへの分配率が多少大きくなる傾向があった。ただし、学習後のエージェントが動く方向は、時計回りのみの場合や反時計回りのみの場合もあったが、最初に動くエージェントや確率によって動く方向が変化する場合も多かった。100 回のうち 73 回のシミュレーションでは、常に同じエージェントがゴールに到達していた(ケース1)。このうちの 71 回では、自分自身への報酬分配率が一番小さかったのが、ゴールに到達したエージェントであった。また、14 回のシミュレーションでは、約 1/3 の試行で一つのエージェントが、残りの約 2/3 の試行で別のもう一つのエージェントがゴールに到達していた(ケース2)。また、11 回のシミュレーションでは、最初に動いたエージェントがゴールに到達していた(ケース3)。その他の 2 回のシミュレーションでは、学習があまり安定せず、上記の3つのケースのいずれにも属さなかった。また、ゴールするエージェントの自分自身への報酬分配率を各ケースごとに平均すると、ケース1が 0.58、ケース2が 0.62、ケース3が 0.66 と順に大きくなる傾向があった。学習の進行によるエージェント A の報酬分配率、各エージェントのゴール確率、ゴール到達までの平均ステップ数の学習による変化を図2から4に示す。この例は、ケース1で、ほとんどエージェント A がゴールしていた。この例のように、温度が 0.01 になる 250000 回ぐらいから学習は安定する場合が多いが、250000 回過ぎても安定せず、ゴール確率が大きいエージェントが何回か入れ替わる場合もあった。また、図4のステップ数はいったん上昇しているが、他のシミュレーションでは、単調減少する場合が多かった。

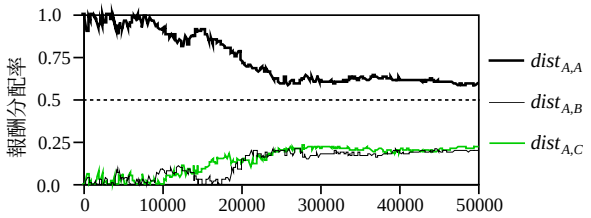


図2　エージェント A の報酬分配率 $dist_{iA}$ の変化

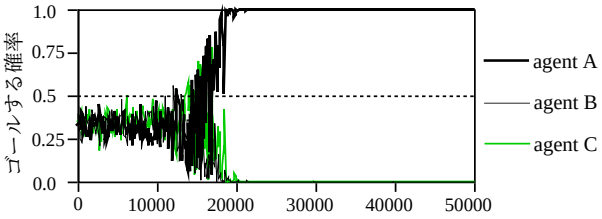


図3　各エージェントのゴール到達確率の変化

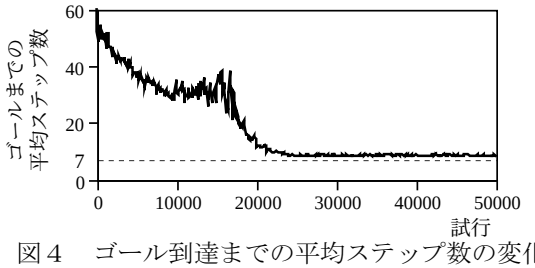


図4　ゴール到達までの平均ステップ数の変化

ゴール後のエージェントの行動を観察すると、エージェント間での駆け引きが見られた。図5(a)の例1（図2から4は、この場合の学習過程）では、まずエージェント B が最初に動き、全体が時計回りで行動し、B が最初にゴールに近づく。しかし、C が B のゴールから動かず、B のゴールのじゃまをして、結果として、A がゴールしている。この時の報酬分配率を見ると、C にとっては、A がゴールした方が B がゴールするより多くの報酬が得られることがわかる。したがって、C は B のゴールをはばみ、A がゴールすることを学習したと解釈することができる。また、例2では、エージェント全体が反時計回

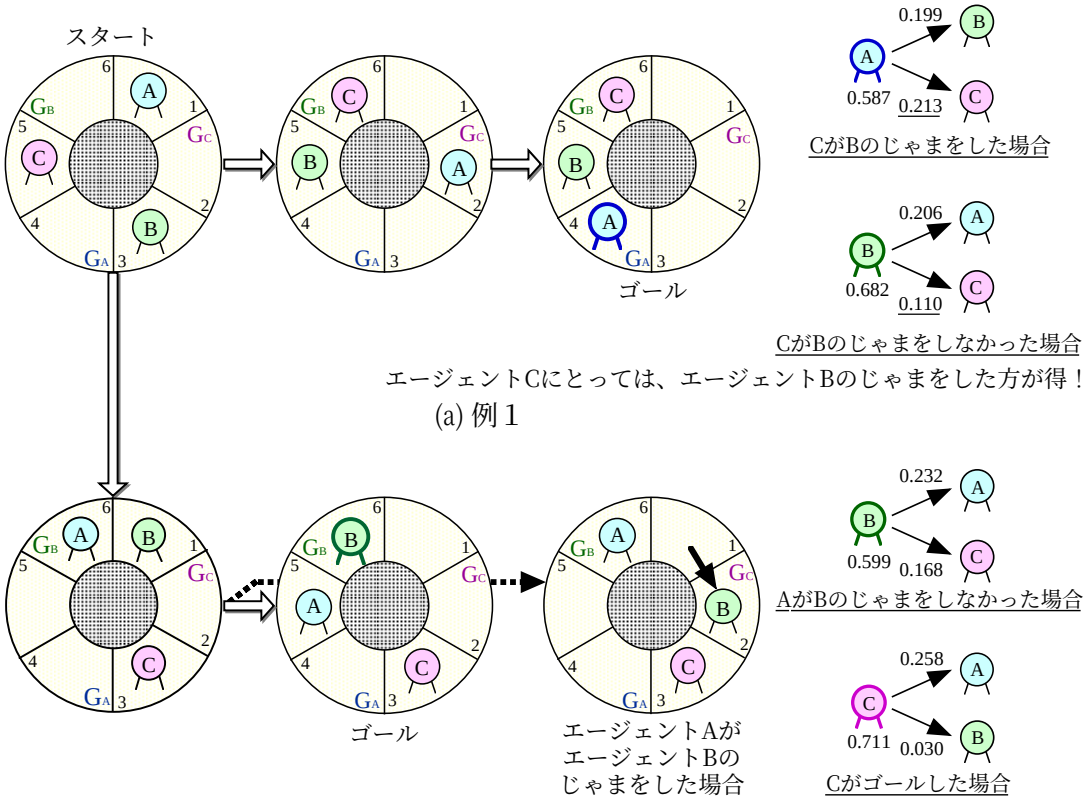


図５　３エージェント問題の学習後の行動と報酬の分配例

りで回り、A はＢをじゃますることなく、B がそのままゴールした。この時、もし、A が B をじゃますると、B は逆戻りして C のじゃまをした。これは、C がゴールしても B はほとんど報酬をもらえないためと考えられる。そして、結果として、C もゴールできないことがわかる。したがって、この場合は、A はＢに道を譲った方が得であると学習したと解釈できる。

5. 結論と今後の課題

マルチエージェント強化学習において、報酬の分配が必要とされる問題を明らかにし、報酬分配の学習則を３つ以上のエージェントの場合に拡張した。そして、３エージェントの問題に適用し、報酬分配が学習できることを確認した。また、２エージェント問題と比較すると、相手のエージェントが２ついるため、学習後の解がいくつか存在し、学習が多少不安定になるが、エージェント間の駆け引きを観察することができた。

また、本学習アルゴリズムは、タスクごとに学習が必要であり、また、エージェント数が増大すると学習が困難になる。他のエージェントの行動を見て、その行動に対して報酬を分配するなどの対策が必要である。

謝辞

本研究の一部は、日本学術振興会未来開拓学術研究プロジェクト「生物型適応システム」(JSPS-RFTF96I00105)、および、文部省科研費(#10450165)の補助の下で行われた。

参考文献

[1] 白川英隆，木村元，小林重信，”強化学習による協調的行動の創発に関する実験的考察”，第 25 回知能システムシンポジウム予稿集, pp. 119-124 (1998)

[2] Ono, N. and Fukumoto, K., ″Multi-agent Reinforcement Learning: A Modular Approach″, Proc. of ICMAS-96, pp.252-258 (1996)

[3] 柴田克成、伊藤宏司，″2エージェント強化学習における報酬分配の自律学習″，ロボティクス・メカトロニクス講演会(ROBOMEC) ’99, 1A1-28-036, (2pages) (1999)

[4] Shibata, K. and Ito, K., ″ Emergence of Individuality and Sociality by Reinforcement Learning in Multi-Agent Systems ″, Proc. of AROB(Int’l Sympo. on Artificial Life and Robotics) 5th ’00, Vol 2, pp.589-592 (2000)