

第2回弓ヶ浜セミナー (2014年1月22日開催)  
**自律分散システムに関する身近な話題と  
これまで・これからの研究紹介**

英文タイトル: Some selected issues in autonomous distributed systems, previous and future studies of the author

徳光 政弘\*,  
Masahiro TOKUMITSU

概要

自律分散システムは複数のエージェントから構成され、各エージェントは他のエージェントと相互作用する。エージェントは時々刻々と変化する環境に適応して行動する必要がある。例えば、コンピュータネットワークでウィルスが蔓延している場合には、変化する状況に応じて修復する必要がある。本稿では、自律分散システムの中心的な問題であるエージェントの環境適応を取り上げ、エージェントの戦略を定量的に評価する適応的戦略を紹介する。

1 はじめに

自律分散システムは、自律的に意思決定をして行動する複数のエージェントから構成される。エージェントは他のエージェントとネットワークを通じ、相互作用して目的の動作をする。しかし、相互作用の結果、ネットワークでのウィルス蔓延 [2] や、データ転送速度 [6] 等のシステム全体の性能低下が発生することもある。

自律分散システムの設計と性能評価は重要な課題である。自己修復ネットワークモデル [3, 5] は、利己的に動作するノードが相互に修復するモデルであるが、利己的に動作するノードがネットワークの多くを占めると、ネットワーク全体の修復性能が低下することが示された。利己的に動作するエージェントはシステム全体の性能の低下を招くこともあるため、利己的に振る舞うエージェントを協調させ、システム全体の性能を向上させることが重要となる。また、システム全体の性能を評価するだけでなく、性能を決定する要因の一つであるエージェントの戦略が定量的に評価する必要がある。

自律分散システムは環境の変動に対して時空間的に適応する必要がある。例えば、コンピュータネットワークにおけるウィルス蔓延について考える。ネットワーク上のウィルス拡散を防ぐ方法の手段として、エージェント

が相互に修復する自己修復ネットワークモデル [3] が提案されている。エージェントはウィルスが発生した場合には、修復を行う必要がある。一方で、ネットワーク上にウィルスが存在しない場合には、修復の動作は無駄であるから、その行動は控える必要がある。この例のように、エージェントはそのネットワーク環境に応じて、自身にとって最適な行動を選択する必要がある。

本稿では、エージェントの行動を決定する指針である戦略に対して、その戦略の環境適応の性能を評価する手法である適応的戦略について紹介する。適応的戦略 [7] は、エージェントが他のエージェントと相互作用する中で、自身の戦略が他のエージェントのものと比較して、環境の変化に対して適切であったかどうかを定量的に評価できる。本手法は、Axerlod[1] による囚人のジレンマの総当り戦略トーナメントにおける TFT (Tit-for-Tat、しっぺ返し戦略) の成功も定量的に評価可能である。TFT 戦略は、対戦相手と協調することで高い利得を獲得し、トーナメントで優勝した。本稿で紹介する適応的戦略は、他の戦略と協調する TFT 戦略の振る舞いも定量的に特徴付けることができる。

2 適応的戦略

適応的戦略は二つの考え方、(1) 協調的であること、(2) 自己寛容性、これら二要素を性能評価に含めている。(1) の要請は、変化する環境に対して他の戦略と協調す

\* 原稿受理 平成 25 年 12 月 6 日

\*\* 電子制御工学科

ることで獲得する利得の低下を防ぐこと。(2)の要請は、自身と似ているか同一の戦略に対しても協調的に振る舞うことである。これら2要素が適応的戦略の性能評価に反映される。

$S$  は戦略集合を表すとする。 $N$  は戦略集合  $S$  の濃度を表すとする。 $i, j, k$  は戦略集合  $S$  の要素である戦略の番号を表すとする。戦略番号  $i$  の戦略を  $s_i$  と表すとする。戦略  $s_i$  が  $s_j$  に対して得る期待利得を  $E_p[s_i|s_j]$  と表す。戦略  $s_i$  が全戦略に対して得る期待利得を  $E_m[s_i]$  と表すとする。

適応的戦略の観点から見た戦略の強さを適応量とし、戦略  $s_i$  の適応量を  $E[s_i]$  と表すとする。次式で示すように、適応量はゲームにおけるエージェントの相互作用の結果で表される。

$$E[s_i] = \frac{1}{NM} \sum_{s_j \in S} E_p[s_i|s_j] E_m[s_j] \quad (1)$$

ただし、戦略  $s_j$  が得る期待利得  $E_m[s_j]$  は次式で表される。

$$E_m[s_j] = \frac{1}{NM} \sum_{s_k \in S} E_p[s_j|s_k] \quad (2)$$

適応量は式 (1) で定義されているように、その値は0から1に収まるように正規化される。適応量の値が1に近ければ、他の戦略と協調的に振る舞い、0に近ければ利得を得ることができなかつたことを示す。

### 3 シミュレーション

自己修復ネットワークモデルを例に、 $kC$  戦略 [3] をエージェントに適用し、適応的戦略の観点で性能評価を行った。 $kC$  戦略は空間型戦略 [4] で、エージェント周囲の他のエージェントが前のステップに選択した行動 (協調または裏切り) により、そのステップの行動が決定される。 $k$  は寛容度を表し、0から9の離散値を取りうる。 $k$  の値ごとに戦略を用意し、総当り戦でシミュレーションを行った。

表1は、故障率ごとに最高利得を獲得した戦略と適応量をそれぞれ示している。故障率0.02において、最大の適応量を得ている戦略は7C、最大の利得を獲得している戦略は6Cである。戦略7Cが獲得した利得は0.399であった。戦略7Cは最大の利得は得ていないものの、比較的高い利得を得ている。適応的戦略から評価すると、7C戦略は6C戦略が高い利得を得ることができない対戦相手の戦略に対しても、他の戦略に比べて高い利得を得たことを示している [7]。

表1 故障率ごとの最高利得を獲得した戦略と適応量 [7]

故障率	戦略	リソース量	戦略	適応量
0.00	6C	0.501	6C	0.118
0.01	6C	0.456	6C	0.094
0.02	6C	0.414	7C	0.081
0.03	6C	0.388	7C	0.071
0.04	7C	0.380	7C	0.066

## 4 おわりに

自律分散システムに関する身近な話題である環境適応に関して、エージェントの戦略を評価する手法の適応的戦略について紹介した。適応的戦略の観点でエージェントの戦略を定量的に評価することで、他のエージェントとどの程度協調しているかを調べることができる。今後は、適応的戦略の考え方に基づいた自律分散システムの設計手法の研究が必要である。

## 参考文献

- [1] R. Axelrod. *The Evolution of Cooperation, Basic Books*. New York, 1984.
- [2] Z. Dezsó and A.-L. Barabási. Halting viruses in scale-free networks. *Phys. Rev. E*, 65:055103, 2002.
- [3] Y. Ishida. A critical phenomenon in a self-repair network by mutual copying. *Lecture Notes in Computer Science*, 3682:86–92, 2005.
- [4] Y. Ishida and T. Mori. Spatial strategies in a generalized spatial prisoners dilemma. *Artificial Life and Robotics*, 9(3):139–143, 2005.
- [5] M. Oohashi and Y. Ishida. *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, chapter A Game Theoretic Approach to Regulating Mutual Repairing in a Self-Repairing Network, pages 281–286. Springer, Netherlands, 2007.
- [6] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.
- [7] M. Tokumitsu and Y. Ishida. Adaptive strategies: A novel game-theoretic analysis for autonomous distributed systems in dynamic environments. *International Journal of Innovative Computing, Information and Control*, 10(1):373–387, 2014.