Reaching Distributed Equilibrium with Limited ID Space*

Dor Bank, Moshe Sulamy, and Eyal Waserman

Tel-Aviv University, Israel

Abstract. We examine the relation between the size of the *id* space and the number of rational agents in a network under which equilibrium in distributed algorithms is possible. When the number of agents in the network is not a-priori known, but the *id* space is limited, a single agent may duplicate to gain an advantage but each duplication involves a risk of being caught. Given an *id* space of size L, we provide a method of calculating the threshold, the minimal value t such that agents know that $n \geq t$, such that the algorithm is in equilibrium. We apply the method to Leader Election and Knowledge Sharing, and provide a constant-time approximation $t \approx \frac{L}{5}$ of the threshold for Leader Election.

Keywords: Rational agents · Game theory · Leader election.

1 Introduction

We consider the model of distributed game theory [2, 1, 3, 5, 8, 4, 9], in which the participants are rational agents, and may deviate from the algorithm when it increases their personal gain. The goal is to design distributed algorithms that are *in equilibrium*, that is, where no agent has an incentive to cheat.

Previous works [3, 5, 8, 4, 9] assumed that n, the number of agents in the network, is a-priori known to all agents. When n is not a-priori known, in some distributed algorithms an agent may cheat by duplicating itself (perform a Sybil Attack [7]) in order to gain an advantage. We consider the case where the *id* space is limited and any duplication involves a risk of detection, i.e., guessing an *id* that might already be taken by some other agent.

For the *id*-space $ID = \{1, 2, ..., L\}$, and when all agents a-priori know that n, the true number of agents in the network, distributes uniformly $n \sim U[t, L]$, what is the minimal threshold t we must provide the agents for the algorithm to reach equilibrium?

2 Model

The model is a standard synchronous message-passing model of a 2-vertex connected network of $n \geq 3$ nodes, each node representing an agent.

* This research was supported by the Israel Science Foundation (grant 1386/11).

Each agent a-priori know its input (if any), its *id*, the *id*-space $\{1, 2, ..., L\}$ and the threshold $t \in \mathbb{N}$ s.t $3 \leq t \leq n \leq L$. We assume the prior over any unknown information is uniformly distributed over all possible values. We assume all agents start the protocol together. If not, we can use the Wake-Up building block [5] to relax this assumption.

Each rational agent $\mathcal{A}$ wants to maximize its utility function $u_{\mathcal{A}} : \mathcal{O} \to \mathbb{R}$ where $\mathcal{O}$ is the set of all possible outputs to the algorithm. A rational agent participates in the algorithm but may deviate from it if a deviation increases its *expected* utility, while assuming all other agents follow the protocol.

To differentiate from Byzantine faults, all utility functions must satisfy the *Solution Preference* [5] property, which ensures agents never prefer an outcome in which the algorithm fails over one in which it terminates correctly. An algorithm is said to be *in equilibrium* if no agent, at any point in the algorithm execution, can unilaterally increase its utility by deviating from the algorithm.

2.1 Duplication

Since n is not a-priori known to agents, an agent $\mathcal{A}$ can deviate by simulating m imaginary agents. Each duplicated agent must be assigned an *id* and duplication involves a risk of choosing an *id* that already exists, rendering it non-unique, and causing the algorithm failure. We assume m and the *id*s of all m duplicated agents must be chosen at round 0, before the algorithm starts.

2.2 Leader Election

Each agent $\mathcal{A}$ outputs $o_{\mathcal{A}} \in \{0, 1\}$, $o_{\mathcal{A}} = 1$ if $\mathcal{A}$ was elected leader, and $o_{\mathcal{A}} = 0$ otherwise. The set of legal output vectors is defined as: $O_L = \{\underline{o} \mid \exists \mathcal{A} : o_{\mathcal{A}} = 1, \quad \forall \mathcal{A}' \neq \mathcal{A} : o_{\mathcal{A}'} = 0\}$

We assume a *fair* leader election[3] where, at the beginning of the algorithm, each agent has an equal chance to be elected leader, and assume agents prefer to be elected leader.

2.3 Knowledge Sharing

In the problem (from [4], adapted from [5]), each agent $\mathcal{A}$ has a private input $i_{\mathcal{A}}$ and a function q, where q is identical at all agents. An output is *legal* if all agents output the same value. An output is *correct* if all agents output $q(I)$ where $I = \{i_1, \ldots, i_n\}$. The function q satisfies the Full Knowledge property[5, 4], which states that when one or more input values are not known, any output in the range of q is *equally* possible. We assume that each agent $\mathcal{A}$ prefers a certain output value $p_{\mathcal{A}}$. Following [4], in this paper we only discuss Knowledge Sharing in ring graphs.

3 Solution Basis

Equation 1 defines the necessary condition for equilibrium in the distributed problem in the presence of rational agents:

$$\sum_{k=t}^{L} e_0(k) \geq \max_m \sum_{k=t}^{L-m} p_m(k) e_m(k) \tag{1}$$

Where $e_m(k)$ is the expected utility of an agent simulating m false duplicates, when k true agents participate in the network; $p_m(k)$ is the probability of successfully choosing m *id*s that are not yet taken, generally $p_m(k) = \frac{\binom{L-k}{m}}{\binom{L-1}{m}}$. We are interested in the minimal threshold t that satisfies Equation 1, and it can be calculated in $O(L^3)$ running time, by trying all values for t.

3.1 Enhancements

Linear Threshold For most algorithms there exists L_0 such that for any $L > L_0$, there exists a pivot value t_0 such that for any $t \geq t_0$ the algorithm is in equilibrium, and for any $t < t_0$ it is not in equilibrium. In such cases we can use binary search to improve the running time to $O(L^2 \log L)$.

Limited Duplications For some algorithms there exists a specific duplication number m', such that if there exists m for which agent has an incentive to deviate, then it also has an incentive to deviate with m' duplications. For such algorithms we only need to examine a single duplication value, improving the running time to $O(L^2)$.

For algorithms that satisfy both enhancements, the running time is improved to $O(L \log L)$.

4 Contributions

Here we summarize our contributions. Details and full proofs are provided in the full paper [6].

4.1 Leader Election

The Leader Election algorithm [3, 5] satisfies both enhancements. Thus, the minimal threshold can be found in $O(L \log L)$ time.

Particularly, whenever an agent has an incentive to deviate by duplicating m agents, it also has an incentive to deviate by duplicating 1 agent. Thus, to check for equilibrium it suffices to check the case $m = 1$.

Furthermore, we prove a constant-time approximation of the Leader Election threshold that shows the minimal threshold t for equilibrium is in the range $0.2L < t < 0.21L$.

4.2 Knowledge Sharing

The Knowledge Sharing algorithm [5, 4] (in a ring) satisfies only the "Linear Threshold" enhancement. Thus, the minimal threshold can be found in $O(L^2 \log L)$ time.

5 Acknowledgment

We would like to thank Yehuda Afek for helpful discussions and his course on Distributed Computing which has inspired this research, and to Sivan Schick for his contributions to this paper.

References

1. Abraham, I., Alvisi, L., Halpern, J.Y.: Distributed computing meets game theory: Combining insights from two fields. SIGACT News **42**(2), 69–76 (Jun 2011). https://doi.org/10.1145/1998037.1998055, http://doi.acm.org/10.1145/1998037.1998055
2. Abraham, I., Dolev, D., Gonen, R., Halpern, J.Y.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: PODC. pp. 53–62 (2006)
3. Abraham, I., Dolev, D., Halpern, J.Y.: Distributed protocols for leader election: A game-theoretic perspective. In: DISC. pp. 61–75 (2013)
4. Afek, Y., Rafaeli, S., Sulamy, M.: Cheating by Duplication: Equilibrium Requires Global Knowledge. ArXiv e-prints (Nov 2017)
5. Afek, Y., Ginzberg, Y., Landau Feibish, S., Sulamy, M.: Distributed computing building blocks for rational agents. In: Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing. pp. 406–415. PODC '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2611462.2611481, http://doi.acm.org/10.1145/2611462.2611481
6. Bank, D., Sulamy, M., Waserman, E.: Reaching Distributed Equilibrium with Limited ID Space. ArXiv e-prints (Apr 2018)
7. Douceur, J.R.: The sybil attack. In: Revised Papers from the First International Workshop on Peer-to-Peer Systems. pp. 251–260. IPTPS '01, Springer-Verlag, London, UK, UK (2002), http://dl.acm.org/citation.cfm?id=646334.687813
8. Halpern, J.Y., Vilaça, X.: Rational consensus: Extended abstract. In: Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing. pp. 137–146. PODC '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2933057.2933088, http://doi.acm.org/10.1145/2933057.2933088
9. Yifrach, A., Mansour, Y.: Fair Leader Election for Rational Agents. PODC '18