Towards Completely Decentralized Mustering for StarCraft*

(Extended Abstract)

Zachary Suffern
Georgia Tech
Atlanta, Georgia
zsuffern0614@gatech.edu

Craig Tovey
Georgia Tech
Atlanta, Georgia
ctovey@gatech.edu

Sven Koenig
USC
Los Angeles, California
skoenig@usc.edu

ABSTRACT

We study decentralized agent coordination with performance guarantees by developing a primitive for mustering teams of agents of minimum acceptable team sizes for StarCraft using randomization to accurately estimate the size of the team.

Categories and Subject Descriptors

Computing Methodologies[**Artificial Intelligence**]: Distributed Artificial Intelligence - *Cooperation and Coordination*

General Terms

Algorithms

Keywords

agent mustering, agent coordination, randomization, StarCraft

1. INTRODUCTION

It can be impossible for autonomous agents to perform as effectively as they theoretically would under omniscient centralized control. The standard performance measure of effectiveness is the "price of anarchy" [1], which is the ratio of the productivity of an idealized centralized system to that of the decentralized one. However, the system outcome is binary in many situations: the team either succeeds or fails. The resulting price of anarchy could then be only 1 or ∞, which is too crude a range. Instead, let α (and β) be the resource level (here: number of agents) needed for success with centralized (and decentralized, respectively) control. Then, we define β/α to be the *price of decentralization.*

We use the popular game StarCraft as testbed [2] because properties like robustness and adaptability to change can only be evaluated experimentally, not analytically. StarCraft is a well-known preexisting domain with an API and all the complexity we need. Creating our own experimental testbed would inevitably be biased. Decentralized StarCraft agents need the capability to coordinate autonomously with each other, for example in the context of the self-assembly of teams of agents of minimum acceptable team sizes (= mustering). Our long-term challenge for this testbed is for each agent to control itself based only on $O(1)$ sensor range, $O(1)$ length messages sent within a $O(1)$ communication range, using only $O(1)$ memory and computation, so as to achieve a $O(1)$ price of decentralization. We require our methods to scale to very large numbers of agents without degrading performance guarantees, and to adjust without intervention to damage, reduction or addition of agents. In the StarCraft environment, we do not expect to compete well against human experts in scenarios with only hundreds of agents (to which the game is currently limited) but we do expect to compete well in scenarios with tens or hundreds of thousands of agents, even against a team of cooperating human players.

1. **Initiation:** An initiation message consisting of the minimum acceptable team size f is sent to one or more agents.
2. **First Reception:** When an agent receives an initiation message, it checks its memory to determine if it has previously received an initiation message. If not, it generates and stores k distinct random integers in the range $[0,M]$. Otherwise it retrieves its set of k random integers.
3. **Transmission:** If the received initiation message does not contain two sets of k integers, the agent sends an initiation message to each nearby agent consisting of f and two copies of its set of k integers. If the received initiation message does contain two sets of k integers, the agent constructs an initiation message consisting of f and the sets of the k smallest and the k largest distinct integers from the union of the received sets and its own set.
4. **Threshold Test:** Let (μ^-,μ^+) be the mean values of the k (smallest, largest) integers in the constructed initiation message. If

$$\mu^- \leq \frac{M(k+1)}{2}(1-(1-\sqrt{p})^{\frac{1}{(f-1)k}})$$

and

$$\mu^+ \geq M(1-\frac{k+1}{2}(1-(1-\sqrt{p})^{\frac{1}{(f-1)k}})),$$

the agent determines that the mustering is complete and sends a muster-complete message to all nearby agents. Otherwise, if the constructed initiation message is different from the most recently transmitted initiation message, the agent transmits it to each nearby agent.
5. **Completion:** Upon reception of a muster-complete message, an agent transmits the muster-complete message to all nearby agents and starts working on the given task.

Figure 1: Agent-Mustering Protocol

2. AGENT MUSTERING

In agent mustering, each agent must decide whether or not to work on a given task. The agents as a group must not decide to work on it unless enough of them do, and they must act simultaneously. The fundamental difficulty in mustering lies in assessing the actual number of agents n using only $O(1)$ memory and $O(1)$ length messages. No agent can sense all of the other agents. Each agent could have a unique identifier, but n identifiers cannot be stored in $O(1)$ memory. A single message counting the number of agents

*The research at Georgia Tech was supported by NSF Grant CMMI 1335301 and a David M. McKenney Family Professorship. The research at USC was supported by NSF under Grants 1409987 and 1319966.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*
Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

could average $\Omega(n^2)$ time to reach all agents. If there were multiple counting messages, agents could not detect which one they had previously incremented. Our solution to these challenges is the agent-mustering protocol shown in Figure 1, with preset parameters M, p and k. Not shown is a mechanism that prevents agents from sending messages after a certain amount of time, which can be set to a worst-case value $10qfT$, where T is the sum of the message process and transmit times and q is the maximal number of nearby agents for each agent. In StarCraft and many other applications where message passing is nearly instantaneous (for example, by infrared or radio) and processing is asynchronous, any small constant time bound (for example, one second) will work. Each message contains less than $2k \log M$ bits of information. Yet, for $f \geq 25$, the following theorem shows that the price of decentralization is less than 5 with probability more than 95% because $4.65f$ agents will start to work (but $f-1$ agents won't start to work) on the given task with probability more than 95%.

THEOREM 1. *With $M \gg 14f > 150$, $k = 1$ and $p = 0.05$, the probability that the agent-mustering protocol makes an incorrect decision with fewer than f or more than $12f$ agents is less than p. For large values of f (≥ 25), with $M \gg 5fk^2$, $k = 9$ and $p = 0.05$, the probability of an incorrect decision with fewer than f or more than $4.65f$ agents is less than p.*

PROOF. Let n be the actual number of agents. For both statements, the probability of at least one tied value among kn values is approximately $(kn)^2/2M$. The conditional probability that there is a tie among the k largest or k smallest values, given that there is a tie, is approximately $2k/kn = 2/n$. Therefore, the probability of a tie affecting the agent-mustering protocol is approximately k^2n/M, which by assumption is negligibly small. For $k = 1$ and $n < f$ the probability of an incorrect decision is maximum at $n = f-1$. Approximate the random integers as uniformly distributed variables on $[0,1]$ scaled by M. In this approximation, the probability that $\mu^- \leq \alpha M$ equals $1-(1-\alpha)^{(f-1)}$. Setting $\alpha = 1-(1-\sqrt{p})^{\frac{1}{(f-1)}}$ makes the probability of this event $\sqrt{p}$. With $f > 10$, the correlation between the first and last order statistics of n independent uniformly distributed variables is negligible. Therefore, we can accurately approximate by treating the event that μ^+ is not smaller than its threshold as independent. By symmetry of the uniform distribution, the probability of this event is also $\sqrt{p}$. The probability of both events occurring is therefore approximately $(\sqrt{p})^2 = p$, as desired. For $k = 1$ and $n \geq 12f$, the probability of an incorrect decision is maximum at $n = 12f$. Then, $P(\text{incorrect decision}|n = 12f)$ $= P(\mu^- \geq \alpha M$ or $\mu^+ < (1-\alpha)M | n = 12f) \leq 2P(\mu^- \geq \alpha M | n = 12f)$ $= 2(1-\alpha)^{12f} < (1-\alpha)^{12(f-1)} = (1-\sqrt{p})^{12} < (0.7764)^{12} < 0.05$.

The proof of the bound for $f \geq 25$ is more complicated. We sketch it here. Let X_i be the ith order statistic of n i.i.d. $U[0,1]$ variables. Then, X_i is known to have a $\beta(i, n-i+1)$ distribution with mean $E[X_i] = \frac{i}{n+i}$ and variance $\frac{i(n-i+1)}{(n+1)^2(n+2)}$. Supposing that we used the median rather than the mean of the first k order statistics in the threshold test step, μ^- would have a beta distribution with parameters $(\frac{k+1}{2}, n - \frac{k-1}{2})$. As $n \to \infty$, this distribution converges to a standard gamma distribution with parameter $\frac{k+1}{2}$. At $k = 9$, the left and right 5% tails of the distribution occur at 1.97 and 9.15, which gives a ratio of 4.65.

We now explain why using the mean is superior to the median in the threshold test step.

LEMMA 1. *For any set of random variables $Y_1, \ldots, Y_m$, it holds that $\sigma_{\sum_{j=1}^m Y_j} \leq \sum_{j=1}^m \sigma_{Y_j}$.*

Proof: This follows from the Cauchy-Schwartz inequality and some algebraic manipulation. ■

From the properties of beta distributions above, the standard deviations of the X_i are proportional to $\sqrt{i(n+1-i)}$. The first and second derivatives (with respect to i) are therefore proportional to $(n+1-2i)/\sqrt{i(n+1-i)}$ and

$$\frac{-2\sqrt{i(n+1-i)} - (n+1-2i)\frac{1}{2}(n+1-2i)/\sqrt{i(n+1-i)}}{(i(n+1-i))},$$

Trial	f	k	Agents (n)	Runs	Failures
1	25	9	24	10,000	47 ($<$ 0.5%)
2	25	9	115	10,000	43 ($<$ 0.5%)

Figure 2: Experimental Results 1

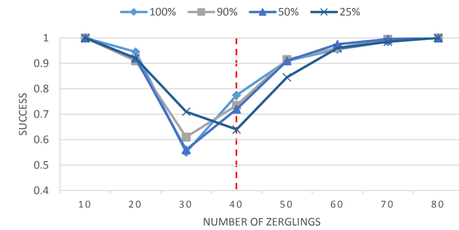

Figure 3: Experimental Results 2

respectively. The former is ≥ 0 for $i < n/2$. The latter is < 0 for $i < n/2$. Hence, the deviations are a concave function of i. Therefore, for odd m,

$$\sum_{i=1}^{m} \sigma_{X_i} \leq m\sigma_{X_{\frac{m+1}{2}}}.$$

Combining this inequality with the lemma gives $\sigma_{\frac{1}{m}\sum_{i=1}^m X_i} \leq \sigma_{X_{\frac{m+1}{2}}}$. This implies that the variability of the mean of the first m order statistics is less than the variability of their median, which is the reason why we use the mean rather than the median to determine the threshold. □

3. EXPERIMENTAL EVALUATION

Experiment 1: We simulated 10,000 trials of the agent-mustering protocol for $f = 25$ and, in one case, $n = f-1 = 24$ and, in the other case, $n = 4.6f = 115$. In all trials, we used $M = 1{,}000{,}000$ and, as in Theorem 1, $p = 0.05$ and $k = 9$. An incorrect decision (= failure) for $n < f$ is the decision to start the given task; a incorrect decision for $n \geq f$ is the decision not to start the given task. Figure 2 shows that the fraction of incorrect decisions is about one tenth of that assured by Theorem 1. Hence, Theorem 1 appears to be a conservative guarantee, which is an encouraging outcome.

Experiment 2: We also simulated 200 trials each of the agent-mustering protocol for $m = 1{,}000{,}000$, $f = 20, 40$ and 80 and n varied from $f/2$ to $2f$. Figure 3 shows the results for $f = 40$ with n on the horizontal axis. As expected, the percentage of correct decisions (= success) is high at the extremes and low when n is close to f. The results for other values of f were very similar. For all values of f tested with $n = f/2$ and with $n = 2f$, at least 90% of the decisions were correct. Therefore, our protocol in effect achieved a price of decentralization equal to 2 with probability $\geq 90\%$. We also tested the resilience of the agent-mustering protocol for communication reliabilities 0.90, 0.50, and 0.25 (shown in different colors). A communication reliability of r was defined as the probability that an agent fails to send any messages when it attempts to transmit messages to nearby agents. We thought this to be a more stringent and realistic model of failure than to fail to transmit each message with a given independent failure probability. Figure 3 shows that our mustering algorithm is strongly resistant to communication failures since the probabilities of making correct decisions are quite similar for all values of r tested. Correct decisions for $n < f$ tend to be slightly more likely for small r because the apparent number of agents is decreased when some of them fail to communicate. Even so, the percentages of correct decisions for $n > f$ are not greatly lower even for $r = 0.25$.

REFERENCES

[1] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[2] G. Robertson and I. Watson. A review of real-time strategy game AI. *AI Magazine*, 35(4):75–204, 2014.