Dynamic Traveling Repairmen Bounty Hunters

Extended Abstract

Drew Wicke
George Mason University
Fairfax, VA
dwicke@gmu.edu

Ermo Wei
George Mason University
Fairfax, VA
ewei@cs.gmu.edu

Sean Luke
George Mason University
Fairfax, VA
sean@cs.gmu.edu

ABSTRACT

Vehicle routing problems such as the multiagent dynamic traveling repariman problem (DTRP) are of interest to many fields and of increasing practical importance in light of advances in autonomous vehicles. DTRP is NP-hard, making approximation methods attractive. However, current approaches do not adequately consider issues special to DTRP, such as discontiguous-space scenarios or alternatives to equitably partitioning the task space. We tackle this problem in a novel way, using a multiagent task allocation technique called *bounty hunting*. In bounty hunting, agents compete to perform tasks non-exclusively in return for reward, and rapidly learn which agents are more adept at various tasks, implicitly splitting up the task space. We demonstrate that bounty hunting can perform efficiently in discontiguous environments, and Pareto-dominates the state-of-the-art heuristic technique, and is particularly good in large-scale scenarios.

KEYWORDS

Multiagent Systems; Task Allocation; Bounty Hunting

ACM Reference Format:
Drew Wicke, Ermo Wei, and Sean Luke. 2018. Dynamic Traveling Repairmen Bounty Hunters. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018*, IFAAMAS, 3 pages.

1 INTRODUCTION

In this paper we discuss a new approach to the distributed or multiagent version of the *dynamic traveling repairman problem* or DTRP [1]. In the multiagent DTRP, m agents are tasked to travel to and from various locations in order to service customers. The customers are generated by a Poisson point process in an area G with rate λ and mean service time of $\bar{s}$. The goal is to minimize the average waiting time of the customers. While abstract, the DTRP is applicable to a wide and increasing range of real-world problems, most notably in routing, autonomous vehicles, and logistics.

The DTRP is NP-hard and so much of the literature has focused on heuristic approximation methods. Surprisingly, presently the most popular and best-performing single-agent-case solution is to simply follow a *Nearest Neighbor* (NN) policy [2]. In the distributed case, it has been proven that by equitably partitioning the space G into k regions, each assigned to a unique agent, and by following some optimal single-agent policy, the overall system will be near-optimal in general [4]. The average waiting time of the tasks can be described as $T \sim \gamma^2 \frac{\lambda G}{m^2(1-\rho)^2}$ where $\rho = \frac{\bar{s}\lambda}{m}$ is the load factor and γ must be determined experimentally.

Very little research has been done in building equitable partitions. Current approaches attempt an equitable partition of the space by running a distributed gradient descent algorithm on the weights of a power diagram [3]. These methods assume that the agents are resources to be allocated to the generated regions, and that the agents share internal information amongst themselves in order to create the equitable partitions. This may fail if not all of the agents are following the same algorithm, or are otherwise unwilling or unable to share information.

Additionally, these methods assume a contiguous space. While some algorithms handle Poisson point processes that are spatially nonuniform, current theory does not address discontiguous spaces. However, in many real-world scenarios there will be spaces where no tasks are generated, creating discontiguous task regions.

We will demonstrate the use of *bounty hunting* as an effective alternative heuristic solution to the DTRP. Bounty hunting is a novel alternative to the use of auctions in multiagent task allocation [5–7], in which a *bail bondsman* offers an ever-increasing *bounty* (or reward) for various tasks up for grabs, and multiple *bounty hunters* compete to finish these tasks. The bounty on a task is awarded only to the bounty hunter who completes it first. There is no task exclusivity: multiple bounty hunters can commit to the same task at the same time. However, tasks cannot be simultaneously serviced by more than one agent. This means that our bounty-hunting variation will not explicitly partition the space at all, unlike previous methods.

Bounty hunting only works if each agent can adapt so as to learn which tasks are worth pursuing. At that point agents will have largely ceded tasks to one another, effectively (in the DTRP case) creating a loose partition of the space. However this partition is dynamic, allowing for flexibility that the hard-partitioning methods cannot provide. Furthermore, this approach works better if the bounty hunters are equipped with the ability to *signal* to other bounty hunters that they intend to work on specific tasks.

We will begin by presenting our version of the bounty-hunting model and signaling method. We examine a discontiguous environment, comparing our bounty-hunting method to NN. Moving to the multiagent case, we show how our partition-free approach can significantly outperform existing hard-partitioned methods. We conclude that bounty hunting Pareto-dominates the NN approach.

2 METHODS

2.1 Bounty-Hunting Model

The bounty-hunting model has previously been studied and formally defined for the dynamic multiagent task allocation problem [5, 7]. Here, we consider how the bounty hunters learn when

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

to signal to the other agents that they are attempting a task. Bounty hunters decide what task to work towards at each time step, and may abandon a task that they are working on or traveling toward in order to undertake another task. They keep track of the distance to the task that they were traveling toward when they abandon this task, in order to develop an expected distance. This distance indicates the range within it becomes more probable that the agent will complete the task without abandoning it. When the distance between the agent and the task i is less than or equal to the point of average task abandonment, the agent will signal to nearby agents their intention of completing the task i. Signaling gives a more reliable indication of the agent's intentions while allowing the agents to have the freedom to abandon tasks. The bounty hunters use these signals and past interaction with agents in order to determine the probability α of successfully completing the task.

The bondsman sets the bounty rate, R and an initial starting bounty B_0 so the bounty at time t is $B_i(t) = B_0 + R \cdot t$. Distance to a task is defined as $\|h(t) - l_i\|$; the agent's current location is $h(t)$, and the task location is l_i. The average service time $\bar{s}$ for tasks is learned through exponential averaging: $\tilde{s} = (1 - \psi)\tilde{s} + \psi s$, where s is the service time of the finished job, and ψ is the learning rate (in our experiments $\psi = 0.05$), and $\tilde{s}$ is the agent's estimate of $\bar{s}$. Finally, the set of available tasks is defined as $I(t)$. We can then calculate the expected bounty received for task i, $U_i(t)$, and the task that the agent will work on, $I^*(t)$, as:

$$U_i(t) = \alpha\left(\frac{B_i(t)}{\|h(t) - l_i\| + \tilde{s}} + R\right) \qquad I^*(t) = \underset{\forall i \in I(t)}{\operatorname{argmax}}\, U_i(t)$$

Therefore, the bounty hunters learn to not go after tasks other agents have signaled while maximizing their total expected bounty per time step. When $R = 0$ the bounty hunters follow a NN discipline, traveling to the closest task.

2.2 Equitable Partitions Nearest Neighbor

This method is used to compare against the bounty hunter approach. Here we split the area G into m equitable partitions and assign each agent to a partition. Because we consider uniformly distributed tasks in a square region we are able to manually partition the space. However, if the tasks are distributed by some other distribution in a convex space, then the space may be equitably partitioned through a gradient descent algorithm of a power diagram as defined in [3]. Then each agent follows the single-agent NN policy, traveling from their current task to service the next closest task (with the ability to abandon tasks).

3 EXPERIMENTS

3.1 Discontiguous Tasks

This experiment illustrated the role that the bounty rate plays in an environment where the NN approach struggles.

For this experiment the bounty rate was set to 5.0. We created two regions of 40×40 that were centered at (20,20) and (150,150) in a virtual space. The agent started in the region centered at (20,20). Tasks had a mean task-generation rate of $\lambda_1 = \lambda_2 = \frac{1}{32}$. This experiment was run for 1,000,000 time steps.

Table 1: Discontiguous Tasks: Average waiting time for the NN approach and bounty hunting with a bounty rate of 5.0.

$\bar{s}$	Nearest Neighbor	Bounty Hunting
8.00	9305.95	2325.55
9.00	12211.07	3189.18
10.00	16086.59	4617.20
11.00	21221.22	7123.05
12.00	27554.38	11684.93
13.00	39723.68	21157.16

Results. As we see in Table 1, the NN method performed very poorly compared to the bounty hunting approach. This was because the bounty rate motivated the agents to more frequently service tasks that were further away.

3.2 Sixty-Four Agents

In order for the system to scale, we had to limit the communication between the agents. Specifically, the bounty hunters could only communicate with other agents within a radius of 40 units, and were aware of tasks within a radius of 40. For this experiment we increased the size of the environment to a 320×320 area and we set $\lambda = \frac{64}{80} = 0.8$ and $\bar{s} = 65, 60, 55, 50, 45, 40$, and yielding load factor values of $\rho = 0.8125, 0.75, 0.6875, 0.625, 0.5625, 0.5$. We set the base bounty to a fixed $B_0 = 500$, and each experiment was run for 300,000 time steps.

Results. To compare the efficiency of each approach in heavy traffic, we determined the coefficient γ of the average waiting time T of the tasks. To do so, we plotted the experimentally determined average waiting time of the tasks (for each value of ρ) versus $\frac{\lambda G}{m^2(1-\rho)^2}$ and determined the slope of the resulting line using least squares with a degree-one polynomial. We found that bounty hunting with a bounty rate of 0.0 had $\gamma = .62$ and the NN approach had a $\gamma = .84$. This confirmed that bounty hunting outperformed NN.

4 CONCLUSIONS AND FUTURE WORK

We have proposed a multiagent systems technique to solve the dynamic traveling repairman problem. Bounty hunting addresses a failure in current heuristic approaches to the DTRP, largely due to the unique flexibility of the bounty rate. It performs efficiently in a discontiguous environment, thanks to the ability of the bounty rate to bring distant tasks "near." We examine additional metrics for the DTRP and how bounty hunting performs compared to the NN approach in the following technical report [8].

Furthermore, the ability of bounty-hunting agents to learn to not go after tasks other agents have signaled enables bounty hunting to not only match but in some cases outperform the NN equitable-partitions approach, without explicitly splitting up the space. Thus bounty hunting Pareto-dominates NN. Additionally, bounty hunting's flexibility scales well. Bounty hunting therefore offers an effective alternative to equitable partitions in heavy-load settings.

For future work we hope to prove theoretically that the bounty hunters split up the space equitably in an online fashion, and to explore bounty hunting's application to other vehicle routing problems, such as pick-up and delivery.

REFERENCES

[1] Dimitris Bertsimas and Garrett van Ryzin. 1989. The dynamic traveling repairman problem. (1989).

[2] Dimitris Bertsimas and Garrett van Ryzin. 1991. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Operations Research* 39, 4 (1991), 601–615.

[3] Marco Pavone, Alessandro Arsie, Emilio Frazzoli, and Francesco Bullo. 2009. Equitable partitioning policies for robotic networks. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2356–2361.

[4] Marco Pavone, Emilio Frazzoli, and Francesco Bullo. 2011. Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *IEEE Trans. Automat. Control* 56, 6 (2011), 1259–1274.

[5] Drew Wicke, David Freelan, and Sean Luke. 2015. Bounty Hunters and Multiagent Task Allocation. In *International Conference on Autonomous Agents and Multiagent Systems*.

[6] Drew Wicke and Sean Luke. 2017. Bounty Hunting and Human-Agent Group Task Allocation. In *AAAI Fall Symposium on Human-Agent Groups: Studies, Algorithms and Challenges*.

[7] Drew Wicke, Ermo Wei, and Sean Luke. 2016. Throwing in the Towel: Faithless Bounty Hunters as a Task Allocation Mechanism. (2016). IJCAI workshop on Interactions with Mixed Agent Types.

[8] Drew Wicke, Ermo Wei, and Sean Luke. 2018. *Bounty Hunters as Dynamic Traveling Repairmen.* Technical Report GMU-CS-TR-2018-2. Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030-4444 USA.