In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA'03)*
pp2293-2298, Taipei, Taiwan, September 14-19, 2003

Multi-Robot Task-Allocation through Vacancy Chains

Torbjørn S. Dahl, Maja J. Matarić, and Gaurav S. Sukhatme
Robotics Research Laboratory, Center for Robotics and Embedded Systems
Department of Computer Science, University of Southern California
tdahl|mataric|gaurav@usc.edu

Abstract

This paper presents an algorithm for task allocation in groups of homogeneous robots. The algorithm is based on *vacancy chains*, a resource distribution strategy common in human and animal societies. We define a class of task-allocation problems for which the vacancy chain algorithm is suitable and demonstrate how Reinforcement Learning can be used to make vacancy chains emerge in a group of Behavior-Based robots. Experiments in simulation show that the vacancy chain algorithm consistently outperforms random and static task allocation algorithms when individual robots are prone to distractions or breakdowns, or when task priorities change.

1 Introduction

Task-allocation (TA) and scheduling algorithms generally make a set of simplifying assumption about the world in order to find optimal allocation patterns [5]. Complex group dynamics such as varying levels of interference, are commonly not considered. Instead, it is assumed that tasks are independent, i.e., the allocation patterns do not influence the time it takes to complete an individual task. In traditional job shop scheduling situations these are fair assumptions, but in many multi-robot systems with complex group dynamics, the traditional simplifying assumptions do not hold true.

For a cooperative task such as transportation or foraging, the average task completion time depends on the number of robots that are allocated to the same task. Allocating a robot to a task may have either a positive or negative effect on a group's performance according to how much the robot contributes positively, in accomplishing tasks, or negatively, in increasing interference. Such dynamics can be difficult or impossible to model. By using learning however, it is possible, over time, to improve performance toward optimality.

Distributed TA algorithms for Multi-Robot Systems commonly use a flexible arbitration mechanism to allocate tasks according to estimates of eligibility or utility made locally by the participating robots [10, 3, 7]. Complex arbitration mechanisms however depend on fully connected communication networks and as such they face problems in scaling to large groups.

We demonstrate an algorithm for TA in groups of homogeneous robots, based on resource distribution through *vacancy chains* [6]. The algorithm uses local task selection, Reinforcement Learning (RL) for estimation of task utility, and reward structures based on the vacancy chain framework. The algorithm allocates tasks to robots in a way that is sensitive to the dynamics of the group, while also being completely distributed and communication-free.

2 The Vacancy Chain Process

A *vacancy chain* (VC) is a social structure through which resources are distributed to consumers. The typical example is a bureaucracy where the retirement of a senior employee creates a vacancy that is filled by a less senior employee. This promotion, in turn, creates a second vacancy to be filled, and so on. The vacancies form a chain linked by the promotions. The resources that are distributed in this example are the positions, and the consumers are the employees.

Chase [6] proposed that major human consumer goods also move through VCs and that such chains are common in other species including the hermit crab, the octopus, and various birds.

Chase listed three requirements for resource distribution through VCs:

1. The resource to be distributed must be reusable, discrete, and used by only one individual.
2. A vacancy is required before an individual takes a new resource unit, and individuals must need or desire new units periodically.
3. Vacant resource units must be scarce, and many individuals must occupy sub-optimal units.

We demonstrate that VCs can be used to optimize the performance of a group of robots when the given task conforms to the requirements listed above.

2.1 Vacancy Chains for Task Allocation

This work considers a particular subclass of the general multi-robot TA problem where each of the allocated tasks has a given value and can be repeated indefinitely. A task can have any number of robots assigned to it. Assigning a j'th robot to a task is called filling *service-slot* j. A particular number of homogeneous robots, j, servicing the same task, i, will have a corresponding task completion frequency, $c_{i,j}$, dependent on the degree to which they are able to work concurrently without getting in each other's way. The difference in completion frequency together with the task value, v_i, define the contribution of the last robot added or the last service-slot filled. We call this contribution, which can be negative, the *slot-value*, $s_{i,j}$. The formal definition is given in Equation 1.

$$s_{i,j} = (c_{i,j} - c_{i,j-1})v_i \quad (1)$$

By this definition, the group performance is optimized if each robot individually optimizes the value of the service-slot it occupies. This allows us to use distributed optimization algorithms without complex communication structures or centralized decision points. In a scenario where the service-slots are allocated optimally, a failure in a robot servicing a high-value task will result in an empty high-value service-slot that must be reallocated for optimal system performance. Expressed in the VC framework, a vacant, high-value service-slot is a resource to be distributed among the robots.

3 Prioritized Transportation

In the *basic transportation problem*, a group of robots traverse a given environment in order to transport items between sources and sinks. To perform optimally the robots must maximize the number of traversals in general. The basic transportation problem is one of the sub-problems of *foraging* [1, 9]. If the locations of sources and sinks are known, the foraging problem is reduced to a problem of transportation.

We define the *prioritized transportation problem* (PTP) as an extension of the basic transportation problem where the sources and sinks, the *targets* of the transportation, are divided into sets of different priority also called *circuits*. In PTP the robot group must strike the correct balance between target utility and robot interference in order to optimize its performance.

4 Experimental Setup

In order to demonstrate that our system optimized its performance by distributing tasks through a VC structure, we first show that the group structure and individual robot actions satisfy the definition of a VC process. We then show that this form of TA performs significantly better than random and static TA algorithms.

We performed the experiments in simulation on the *Player/Stage* software platform [8]. The robots in the experiments realistically simulated ActivMedia Pioneer 2DX robots with SICK laser range-finders and PTZ color cameras. Each robot wore colored markings so as to be recognized using ActiveMedia's Color-Tracking Software (ACTS). The experiments took place in a simulated 12x8-meter environment with two circuits. The sources and sinks were simulated laser-beacons, effectively bar-codes made from high-reflection material and recognizable by the laser range finder. We did not require actual objects to be carried. A minimum proximity to a target was interpreted as a delivery or a pick-up. Figure 1 shows a graphical rendering of the simulated environment in which the experiments took place, with the two circuits indicated by the dashed arows.

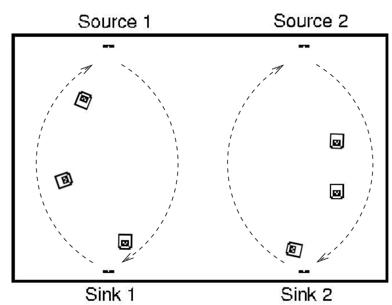

Figure 1: The Simulated Environment

4.1 Reward Structure

The robots received a reward, r_1, whenever they reached a target in circuit one, i.e., source one or sink one in Figure 1. Correspondingly, they received a reward, r_2, whenever they reached a target in circuit two. We call the circuit with the highest related reward the *high-value* circuit and correspondingly, the the circuit with the lowest related reward is called the *low-value* circuit. The robots also received an explicit punishment, p, whenever they were forced to avoid another robot. Robot avoidance was defined as the combination of a minimum laser proximity and presence of color markings.

Given an average frequency of interference, $\bar{i}_{n,m}$, for circuit n when serviced by m robots, the value, $s_{n,m}$, of a service-slot to a robot, in terms of reward, r_n, and punishment, p, is given in Equation 2.

$$s_{n,m} = r_n - p\bar{i}_{n,m} \tag{2}$$

To demonstrate the structures and structural adaptations that define VCs, we designed an initial setup with three robots servicing circuit one and three robots servicing circuit two. This setup imposed a set of constraints on the reward structure.

To keep more than three robots from servicing any of the circuits, it was necessary to make service-slot four on both circuits less attractive to the robots than service slot three on either circuit. This constraint is presented formally in Equation 3.

$$\forall(x, y).s_{x,4} < s_{y,3} \tag{3}$$

In order for a vacancy in the high-value circuit to be filled, service-slot three on the high-value circuit had to be more attractive than service-slot three on the low-value circuit. This constraint is expressed formally in Equation 4, where h denotes the high-value circuit.

$$\forall(x \neq h).s_{x,3} < s_{h,3} \tag{4}$$

We empirically estimated the values $\bar{i}_{1,3}$ and $\bar{i}_{2,3}$ to be 4 per traversal and $\bar{i}_{1,4}$ and $\bar{i}_{2,4}$ to be 6 per traversal. To satisfy the given constraints, we chose r_1 to be 17, r_2 to be 14, and p to be 3. Circuit one was made the high-value circuit.

5 The Adaptive Controller

All robots in our demonstration used the same adaptive, Behavior-Based controller. Based on individual experience they specialized to service a particular circuit. However, they retained an exploration rate, ϵ, of 0.1 to allow them to find and fill new vacancies.

Each controller in our experiments had a set of preprogrammed high-level *approach* behaviors and used Temporal Difference Q-learning to associate different input states with one of these. The Q-tables were initialized with random values between -0.1 and 0.1, the learning-rate, α, was set to 0.1, and the return discount factor γ was set to 0.95. For action selection we used a greedy-ϵ, strategy.

5.1 The Problem Space

We used a minimal input space with one bit indicating whether an object was currently being carried. The action space consisted of four high-level behaviors: *approach source one*, *approach sink one*, *approach source two*, and *approach sink two*. Only the two sink-oriented behaviors were applicable when a robot was carrying an object and correspondingly, only the two source-oriented behaviors were applicable when it wasn't.

Each of the high-level *approach* behaviors consisted of multiple lower level behaviors, such as *target location approach*, *visible target approach*, and *obstacle avoidance*. These low-level behaviors ensured that the robot made progress toward the targets specified by the high-level behavior without getting stuck or colliding with other robots.

6 Results

For each experiment we defined a convergence period and a stable period according to the stability of the system performance. Our student-t tests for statistical significance are all done on a 90% confidence level.

6.1 Initial Task Distribution

This set of experiments used six robots with randomly initialized Q-tables. We did 20 individual experiments of 5 hours, each averaging 3000 traversals or 500 traversals per robot. The convergence period was 1.25 hours.

To show the structure that emerged we look at the last target visited by each robot. This gives seven possible system states. We refer to each state using the notation $h : l$, where h is the number of robots whose last target was on the high-value circuit. Correspondingly, l is the number of robots whose last target was on the low-value circuit. The rows labeled A in Table 1 show the mean, μ, and standard deviation, σ, of the time the system spent in each of the states. The values are percentages of the total stable period. The rows labeled R describe the same values for a set of 20 control trials using a group of robots that randomly chose between the applicable high-level behaviors.

The row labeled T in Table 1 lists the number of different ways to choose a sample of size n from a population of size m, as a percentage of all possible samples, according to Equation 5. It is worth noticing that the time distribution produced by the six random controllers is closely aligned with the theoretical estimate, though the differences are statistically significant.

State		0:6	1:5	2:4	3:3	4:2	5:1	6:0
A	μ	0.1	2.8	19.3	44.5	27.6	5.3	0.4
	σ	0.1	1.7	5.4	6.0	7.0	2.6	0.4
R	μ	1.0	7.4	22.3	33.7	25.3	9.3	1.0
	σ	0.6	1.6	1.5	2.2	1.6	1.4	0.6
T		1.6	9.4	23.4	31.2	23.4	9.4	1.6

Table 1: Time Distribution with Six Robots

$$T = \frac{m!}{n!(m-n)!2^m} \tag{5}$$

The two time distributions given in Table 1 are presented as histograms in Figure 2 with the standard deviation indicated by the error bars for each state.

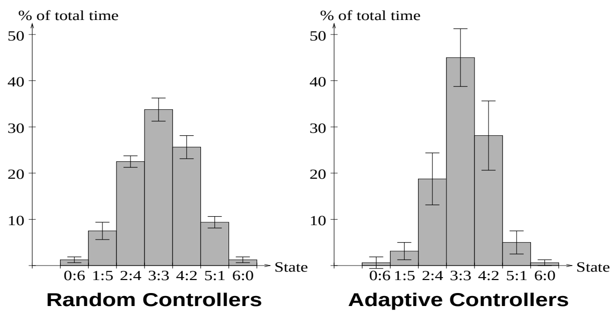

Figure 2: Time Distribution with Six Robots

The increase in the amount of time spent in state 3 : 3 is statistically significant. The time the adaptive group spends in state 3 : 3 is also significantly higher than the time spent in any of the other states.

Figure 3 presents the average performance of a group of robots controlled by the VC algorithm over both the convergence period and the stable period. This group's performance is indicated by the thick, solid line. The average performance of a group of six robots controlled by an algorithm that choses randomly between the high-level approach behaviors is indicated by the dashed line. The performance is calculated as the sum of the delivery frequencies for each circuit weighted by the value of the task. The values used for the performance plots presented here are 10.0 and 1.0 for the high-value and low-value circuits, respectively.

The performance data show that the performance of a group of robots controlled by the VC algorithm is significantly higher than the performance of a group controlled by a random choice algorithm. Together, the time distribution data and the performance data show that the adaptive controllers improve the group's performance by adopting a dedicated allocation pattern.

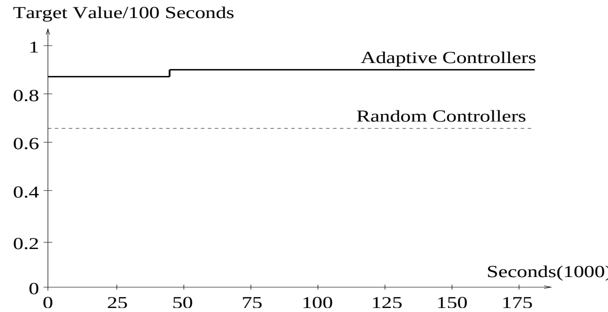

Figure 3: Performance for Six Robots

6.2 Vacancy from Robot Breakdown

The second set of experiments used five robots with Q-tables taken from the end of the initial task distribution experiment. We randomly removed one of the robots dedicated to the high value circuit, thus creating a vacancy on that task. We did 20 experiments with a duration of 7.5 hours. The convergence period was 2.5 hours.

The converged controllers kept the system in state 3 : 2 for a significantly larger amount of time than a group of five random controllers. The values of the time distribution in the stable period are given in Table 2 and a graphical presentation is provided in Figure 4.

State		0:5	1:4	2:3	3:2	4:1	5:0
A	μ	0.3	6.5	35.7	46.2	10.4	0.6
	σ	0.3	3.8	9.0	8.6	4.7	0.6
R	μ	2.2	13.9	31.0	32.9	16.1	2.8
	σ	0.8	2.6	2.2	2.3	2.6	0.4
T		3.1	15.6	31.3	31.3	15.6	3.1

Table 2: Time Dist., Breakdown with Vacancy

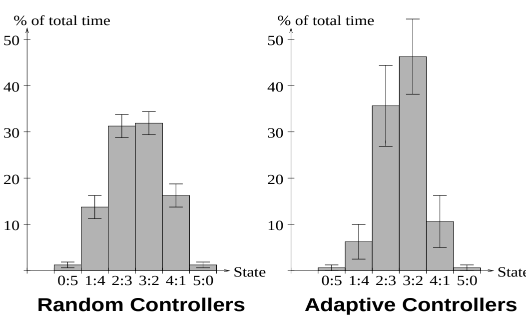

Figure 4: Time Dist., Breakdown with Vacancy

These results show that the group has adapted its structure from one that promotes the 3 : 3 state to one that promotes the 3 : 2 state. Such a change implies that a robot from the low-value circuit has filled the vacancy we created in the high-value circuit.

The performance data presented in Figure 5 show that on the removal of a robot from the high-value circuit, the performance drops sharply. After the re-convergence period, however, the performance rises again to a level that is significantly higher than the performance of five random controllers and also significantly higher than the mean performance, over 20 trials, of a group of robots controlled by a static TA algorithm optimized for six robots. The average performance of the static group is indicated by the thin solid line.

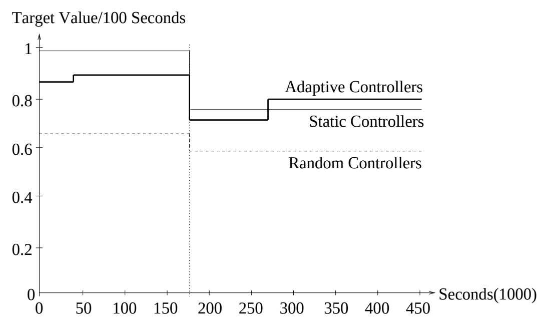

Figure 5: Performance, Breakdown with Vacancy

6.3 Breakdown without Vacancy

This set of experiments was similar to the one presented in Section 6.2, but this time we removed a robot from the low-value circuit. We ran 20 trials for this experiment. The convergence time was 2.5 hours.

The time distribution during the stable period of this experiment, presented in Table 3, was not significantly different from the distribution produced during the experiment where a vacancy was created by the removal of a robot.

State	0:5	1:4	2:3	3:2	4:1	5:0
μ	0.3	6.7	34.6	47.1	10.5	0.7
σ	0.3	3.7	9.2	9.4	3.8	0.4

Table 3: Time Dist., Breakdown without Vacancy

Performance fell significantly when the robot was removed, but remained significantly higher than the performance of five random controllers. There was no significant difference in the performance during the stable period of this experiment and the stable period during the experiment where a vacancy was created. Also, there was no significant difference in performance between the convergence and stable periods. This consistency in the performance reflects the fact that the group structure remained unchanged.

This result demonstrates that our algorithm produces the group structure required for VC distribution, independent of which robot should fail.

6.4 Changing Sub-Task Values

In this set of experiments we started with five robots using Q-tables from the stable period of the experiment presented in Section 6.2. These Q-tables defined three policies servicing high-value service-slots and two policies servicing low-value slots. We then reassigned the values of the tasks by switching them so that the high-value circuit became the low-value circuit and vice versa. This created a scenario where the three policies previously servicing high-value service-slots were now servicing low-value slots, and where the two policies previously servicing low-value slots were now servicing high-value slots. The new setup had a vacancy on the new high-value circuit. We ran 20 individual trials using this setup. The convergence time was 50 minutes.

The time distribution for the stable period, given in Table 4, had significantly higher values for states 3 : 2 and 4 : 1 than the time distribution produced by the experiment where a vacancy was created by the removal of a robot. Compared to the time distribution produced by five robots using a random TA algorithm, the distribution produced in this latest experiment had a significantly higher value for state 3 : 2 and significantly lower values for *all* other states.

State	0	1	2	3	4	5
μ	0.2	5.4	30.3	48.7	14.4	0.9
σ	0.5	3.9	5.2	6.1	1.6	0.3

Table 4: Time Dist., Changed Sub-Task Values

When the task values were changed, the performance fell significantly. After the re-convergence period, however, it was back up to a level that was not significantly different from the initial level.

This experiment showed that the VC TA algorithm is not sensitive to how a vacancy is created, whether by robot failure or by a change in task values.

7 Related Work

Learning has been used to increase the applicability of both centralized and distributed TA and scheduling algorithms. Zhang and Dietterich [12] used RL to learn a centralized policy for payload processing. Being centralized, this and similar approaches do not scale easily to large groups of robots. Blum and Sampels [2] used

Ant Colony Optimization to construct an environmentally embedded, pheromone based, approximate solution to the problem of First Order Parallel Shop Scheduling. Like most scheduling algorithms, this approach assumes independent task completion times. Brauer and Weiss [4] use a distributed RL mechanism for Multi-Machine Scheduling (MMS) where each machine estimates locally the optimal receiver of the material it has processed. This approach, like ours, uses local action selection and utility estimates. The MMS problem however does not contain the complex group dynamics of the transportation problem.

Balch [1] studied performance-related reward functions for robot using Q-learning. The problem Balch considered was *multi-foraging*, where robots gathered pucks of different colors. Balch also looked at *progressive reward functions* or *shaping* and concluded that robots which used local performance-related reward functions or progressive reward functions, performed equally well and better than the robots that were using the global reward function. Our results build on Balch's work and further explore the use of local performance based reward functions for optimizing group performance.

Stone and Veloso [11] have studied mechanisms for dynamic role assignment, but provide specialized, preprogrammed rather than general algorithms for choosing individual roles in order to optimize group performance.

8 Conclusions and Future Work

Our experiments have shown that the VC TA algorithm produces allocation patterns that comply with the definition of task distribution through VCs. They also show that in doing this, the algorithm performs better than random and static TA algorithms. Finally, our experiments show that our algorithm works regardless of whether a vacancy is produced by removing a robot or by changing the values of the tasks.

In the future we aim to extend the VC algorithm to also be able to allocate tasks efficiently in heterogeneous groups of robots while keeping the applicability, scalability, and robustness properties.

Acknowledgments

This work is supported in part by DOE RIM Grant DE-FG03-01ER45905 for "Multi-Robot Learning in Tightly-Coupled, Inherently Cooperative Tasks" and in part by ONR DURIP Grant 00014-00-1-0638.

References

[1] T. R. Balch. Reward and Diversity in Multirobot Foraging. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99) Workshop: Learning About, From and With other Agents*, Stockholm, Sweden, July 31 - August 6 1999.

[2] C. Blum and M. Sampels. Ant colony optimization for fop shop scheduling: A case study on different pheromone representations. In *Proceedings of the 2002 Congress on Evolutionary Computing (CEC'02)*, pages 1558–1563, Honolulu, Hawaii, May 12 - 17 2002. IEEE Press.

[3] S. Botelho and R. Alami. M+ : a scheme for multi-robot cooperation through negotiated task allocation and achievemen. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA'99)*, pages 1234–1239, Detroit, Michigan, May 10-15 1999.

[4] W. Brauer and G. Weiß. Multi-machine scheduling - a multi-agent learning approach. In *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 42–48, Paris, France, July 4 -7 1998.

[5] P. Brucker. *Scheduling Algorithms*. Springer, Second edition, 1998.

[6] I. D. Chase, M. Weissburg, and T. H. Dewitt. The vacancy chain process: a new mechanism of resource distribution in animals with application to hermit crabs. *Animal Behavior*, 36:1265–1274, 1988.

[7] B. P. Gerkey and M. J. Matarić. Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, October 2002.

[8] B. P. Gerkey, R. T. Vaughan, K. Støy, A. Howard, G. S. Sukhatme, and M. J. Matarić. Most valuable player: A robot device server for distributed control. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, pages 1226–1231, Wailea, Hawaii, October 29 - November 3 2001.

[9] D. Goldberg and M. J. Matarić. Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks. In T. Balch and L. E. Parker, editors, *Robot Teams: From Diversity to Polymorphism*, pages 315–244. A K Peters Ltd, 2001.

[10] L. E. Parker. L-ALLIANCE: Task-Oriented Multi-Robot Learning in Behaviour-Based Systems. *Advanced Robotics, Special Issue on Selected Papers from IROS'96*, 11(4):305–322, 1997.

[11] P. Stone and M. Veloso. Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.

[12] W. Zhang and T. G. Dietterich. A reinforcement learning approach to job-shop scheduling. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1114–1120, Montréal, Canada, August 20 - 25 1995. Morgan Kaufmann.