Improving the performance of self-organized robotic clustering: modeling and planning sequential changes to the division of labor

Jung-Hwan Kim and Dylan A. Shell

Abstract— **Robotic clustering involves gathering spatially distributed objects into a single pile. It is a canonical task for self-organized multi-robot systems: several authors have proposed and demonstrated algorithms for performing the task. In this paper, we consider a setting in which heterogeneous strategies outperform homogeneous ones and changing the division of labor can improve performance. By modeling the clustering dynamics with a Markov chain model, we are able to predict performance of the task by different divisions of labor. We propose and demonstrate a method that is able to select an open-loop sequence of changes to the division of labor, based on this stochastic model, that increases performance. We validate our proposed method on physical robot experiments.**

I. Introduction

Studies of self-organized multi-robot systems (MRS) consider multiple agents, each with limited individual capabilities, but with the capacity for synergistic interaction in order to perform tasks collectively. Unlike the more common intentional distributed robot teams, the group's functionality emerges through feedback mediated by the environment and is the product of action rather than representation or calculated reasoning [1]. Self-organized MRS have several potential advantages: simple hardware allows for the production of cheap, specialized, and robust units which exploit economies of scale. However, since the robots in self-organized MRS have limited sensing and manipulation capabilities, it can be difficult to improve the speed of collective performance. It is already known that merely increasing the number of robots will not improve the speed of the system above a certain threshold because of the interference between team members [2], [3]. Principled methods for maximizing system performance (in terms of speed and/or quality) remains challenging for self-organized robot swarms.

In our previous work [4], [5], we introduced a novel approach for *object clustering*, one of the most widely studied task domains for self-organized MRS. The approach we demonstrated consisted of two complementary behaviors: *twisting* and *digging* (Fig. 1 illustrates both). Each robot was assigned with one of these behaviors for the duration of a clustering experiments. With a mix of robots executing the two complementary behaviors, the robots detached the objects from the boundary and successfully generated a single central cluster as shown in Fig. 2. Certain mixes of the behaviors outperformed other mixes aand in different respects. For example, the mix of 2T3D (2 Twisters and 3 Diggers) had reliable performance compared to other cases

Jung-Hwan Kim and Dylan A. Shell are with Department of Computer Science and Engineering, Texas A& M University, College Station, Texas, USA. jnk3355,dshell at cse.tamu.edu

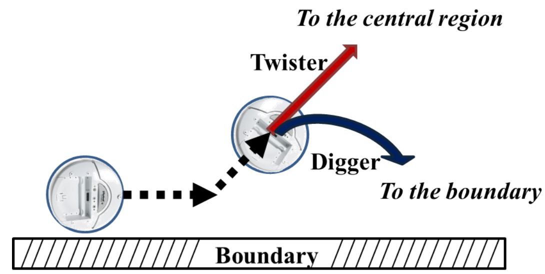
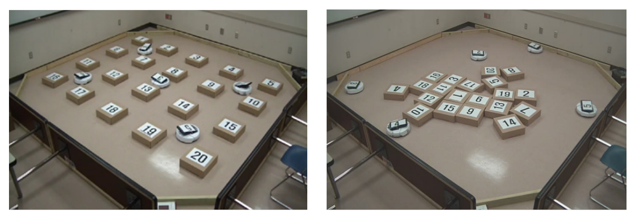


Fig. 1: Trajectories of Twisters and Diggers on the boundary region. Basically the trajectories differ by the way they move away from the boundary wall.



(a) (b)

Fig. 2: The clustering process. (a) initial configuration and (b) final configuration.

while mix 1T4D (1 Twister and 4 Diggers) formed a cluster efficiently in the shortest observed time although it failed in one of its trials. This suggests that, given a preference between reliability and efficiency, an appropriate mix (or distribution of labor) could be determined. In this paper, we attempt to address the question of how to maximize the system's performance by computing a policy for altering the robot division of labor as a function of time.

This research considers a sequencing strategy based on the hypothesis that since clustering performance is influenced by the division of labor, it can be improved by sequencing different divisions of labor. We construct a model in order to predict clustering behavior (in terms of likelihood of success and speed) and propose a method that uses the model's predictions to select a sequential change in labor distribution. Both of these aspects are performed off-line at *design time*. The model is calibrated with values from experiments in which robots maintain a constant distribution of labor. Then the analysis step is conducted in order to produce a labor policy for the robots. This is then executed on-line at *run time*. The system under study involves robots that are unable observe the environment's current state; fortunately, although there is a deal of stochasticity, task performance does have a degree of predictability. We constructed a Markov chain model which abstracts away many of the details of the robots but which captures the important geometric state for the clustering task. The model is used to predict task progress as a function of time, which allows for planning and evaluation of different sequences of workload division.

This paper is organized as follows. After discussing related work on self-organized MRS for object clustering in Section II, we present our model and sequencer in Section III. We validate the result with physical robot experiments in Section IV. Section V concludes.

II. Related work

Object clustering is a widely studied canonical task for self-organized MRS. Deneubourg *et al.*'s classic paper [6] introduced an distributed algorithm inspired by ants' brood sorting and applied it to a simulated MRS. Clustering was achieved with a simple algorithm with only a local density sensor and without direct communication between robots. Beckers *et al.* [7] conducted a physical robot experiment and demonstrated clustering without needing a density sensor by employing a binary threshold sensor. They also explained the emergence of clusters on the basis of the geometry of the clusters. Beside this research, many authors inspired from Deneubourg *et al.* [6] proposed clustering algorithms that use similar approaches [8], [9].

Self-organized MRS robots typically do away with adaptive planning, representation, or calculated reasoning at runtime. In contrast, producing desirable behavior in such systems often focuses on design decisions, employing theory and analysis off-line. One successful approach is to model such systems mathematically as a stochastic processes, which can be a natural fit given the non-determinism often inherent to such systems. The Rate Equation [10], [11], [12], [13] has been used as a useful tool for analysis of collective dynamics of swarm robotic systems. For the clustering task, Martinoli [14] proposed a probabilistic model of an object collection method by quantifying the geometry of clusters and verified it through physical experiments as well as simulations. Thereafter, Kazadi *et al.* [15] provided a mathematical model of clustering dynamics by analyzing conditions where cluster formation occurs, and introduced a characteristic function which described cluster growth properties.

The previous work in robotic clustering mentioned above either focused on empirical demonstrations or considered a simple model in which environmental effects (like boundaries) play no role. In this paper, we develop a practical model that we calibrate with actual data from initial experiments, and then use in order to make predictions about behavior in order to produce a division of labour policy to improve overall clustering performance.

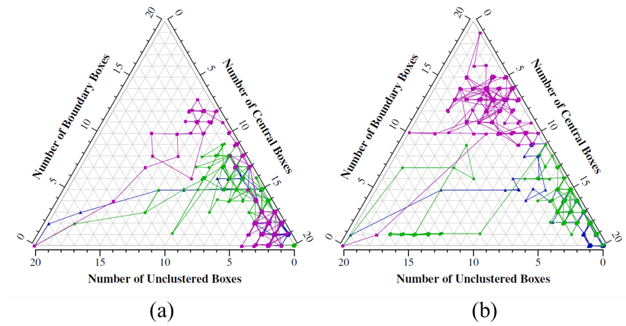


Fig. 3: Ternary plots detailing the cluster dynamics for each trial for two divisions of labor: (a) 1T4D and (b) 2T3D. The system begins in the lower left, with no clusters formed. The goal is for the system to form a single central cluster of 20 boxes, which is the state represented on the lower right corner. Boxes that collect on the boundary show a degree of hysteresis.

III. Approach: From a stochastic model to planned sequences

As mentioned in Section I, objects can be successfully clustered using a mix of robots, each employing one of the two complementary behaviors. Fig. 4 shows the box cluster dynamics for each of the three 90 minute runs of five physical robots for mixes 1T4D and 2T3D on a ternary plot. The axes of the ternary plot reflects the fact that groups of boxes behave in qualitatively different ways depending on whether they are part of a cluster on the boundary, or are part of a cluster in the center, or are not part of any cluster. The spread in each trial reflects changes in the clustering configuration and gives an indication of how goal-directed the cluster formation dynamics are. The plot also illustrates how fluctuations and randomness in the system become manifested as stochasticity in the evolution of the task-performance measure. This view suggested that a discrete-time Markov chain model may allow one to predict the configuration of clustering based on the current transition probabilities.

We observed that certain labor mixes outperformed others. As shown in Fig. 4, the blue trial for 1T4D was extremely efficient, while the magenta trial ended with some boxes on the boundary. The reliability (but comparatively longer time, visible in the meandering trajectory) is visible in the 2T3D case as all the paths converge to the lower right corner. These observations suggest that an appropriate sequence of the different labor divisions might improve clustering performance. That is, by planning the sequence of labor mixes, the system can produce reliable quality and fast object clustering performance too.

In the remainder of this section, a state transition matrix is first computed from empirical data obtained from calibrated experiments. Then, given an initial state condition, the state after n time-steps can be predicted by using the model. Based on a Markov chain model of single strategies, we can further find a better strategy composed of the sequence of different

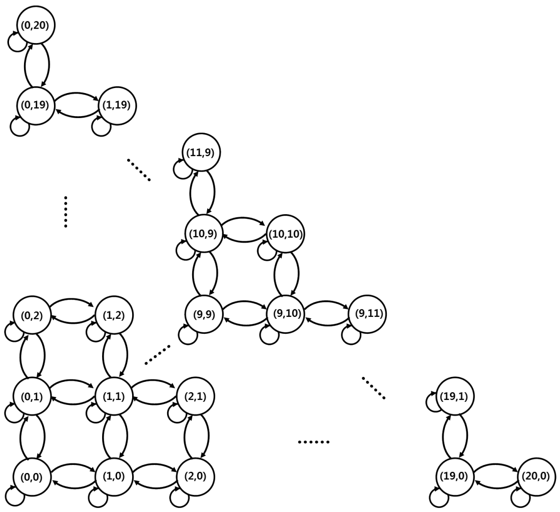


Fig. 4: State diagram for a clustering task.

strategies.

A. The transition matrix

In our problem, the system state is describes the progress of the clustering task. During the clustering process, each box in the workplace may be part of a central cluster, a boundary cluster, or neither. We define the state in the Markov chain model as the number of boxes in central cluster(s) and the number of boxes in boundary cluster(s) $S_t = \{N_c(t), N_b(t)\}$, where $N_c(t)$ and $N_b(t)$ are the number of boxes in central clusters and boundary clusters respectively at time t. Then, $N_c(t) + N_b(t) = N_0 - N_u(t)$ where N_0 is the total number of boxes and $N_u(t)$ is the number of boxes that do not belong to any of the clusters. The number total states is $d = \frac{N_0(N_0+1)}{2}$, and the matrix describing transitions between states has dimension $d \times d$.

As a simplification, we assume that the environment may stay in the same state or change to another state by one-state increments or decrements. Then a state transition can only occur in five directions such as $(i, j) \rightarrow (i, j)$, $(i, j) \rightarrow (i+1, j)$, $(i, j) \rightarrow (i-1, j)$, $(i, j) \rightarrow (i, j-1)$, and $(i, j) \rightarrow (i, j+1)$. The transitions between states is illustrated as a right-angled triangle in Fig. 4.

For each edge, a transition probability is computed by the frequency counts of the boxes moving between states in each time interval. In order to measure the frequency of each state transition, we define an alternative formula which assigns a certain weight in the transited state. The total weight of 1 is assigned when one transition occurred in a time interval. If the transition of the state is varies with a single increment, decrement, or stayed in the same, the total weight of 1 is allotted to the transition.

Let S_{t_0} be the starting state (i_0, j_0) at time t_0, and S_{t_n} be the state (i_n, j_n) after n time intervals from t_0. If we assume that the state transition occurs along edges in the state diagram, the number of steps to approach from S_{t_0} to S_{t_n} is computed by the difference of absolute values of the state grid, $|i_n - i_0| + |j_n - j_0|$. Let x be $|i_n - i_0|$ and y be $|j_n - j_0|$.

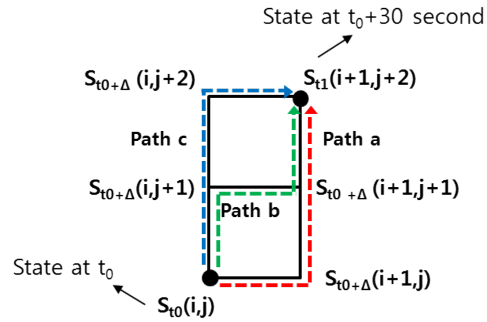


Fig. 5: All paths from S_{t_0} to S_{t_1}. ($t_1 - t_0 = 30$ seconds).

TABLE I: The weight of the state transition.

Edges	Weight	Note
$S(i,j) \rightarrow S(i,j+1)$	2/9	Edges included in two paths
$S(i+1,j+1) \rightarrow S(i+1,j+2)$	2/9	
$S(i,j+1) \rightarrow S(i+1,j+1)$	1/9	Edges included in one path
$S(i,j+1) \rightarrow S(i,j+2)$	1/9	
$S(i+1,j) \rightarrow S(i+1,j+1)$	1/9	
$S(i,j) \rightarrow S(i+1,j)$	1/9	
$S(i,j) \rightarrow S(i+1,j+2)$	1/9	
Total	1	Assigned in one transition

With empirical data, it is possible that $x > 1$ or $y > 2$ in a single time interval. If this is the case, the weight is divided and we consider all paths that reach from the current state to the next state via transitions, and assign a weight proportional to the number of possible routes connecting the states. Fig. 5 illustrates all paths that approach S_{t_1} from S_{t_0} after one time interval. The number of the shortest paths in an $x \times y$ grid map type is $(x + y)!/x!y!$. All edges of each path have the weight divided by the number of edges in the shortest path, $x + y$. In other words, the weight of the edge in a selected path is as follows,

$$W_{edge} = \frac{x!y!}{(x+y)!} \times \frac{1}{x+y}. \quad (1)$$

In addition, since the edges can be selected multiple times as a path, the final weight of the edges will be

$$W_{total} = \frac{x!y!}{(x+y)!} \times \frac{1}{x+y} \times N_s, \quad (2)$$

where N_s is the number of times selected as a path.

The weight of all edges of the state transition in Fig. 5 is shown in Table I. With the rule above assigning weights, a transition matrix is generated by integrating the weighted frequencies of all state transitions that occur over the duration of the calibration experiments. The weighted frequencies are then normalized to calculate the transition probability. That is, if a transition from one state to another state occurs frequently, the probability of the transition is large. In our scenario, the matrix has 231 states, where each state has transition probabilities for 5 directions. We order the 231 states along the rows and columns of the transition matrix as $(0,0), (0,1), \cdots, (0,20), (1,0), \cdots, (1,19), \cdots, (19,0), (19,1), (20,0)$.

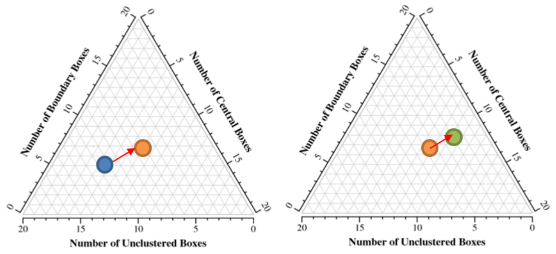


Fig. 6: The state transition in 30 seconds.

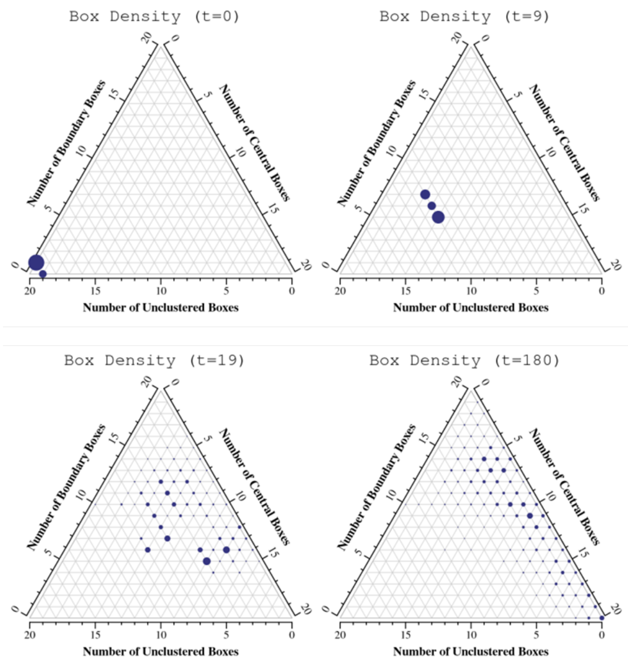


Fig. 7: Variation of the probability distribution of states in n time-steps (n=0, 9, 19 and 180). The probability of large point is relatively higher than the probability of small point.

We constructed a model for all combinations of twisters and diggers, producing transition probability matrices for six divisions of labor from calibrated experiments with 0T5D, 1T4D, 2T3D, 3T2D, 4T1D and 5T0D.

B. Prediction of state transition

After the transition matrix is obtained, S_{t_n} can be predicted by a discrete-time Markov chain [16]. Let M be the state transition matrix of our system. The ij-th entry m_{ij} of M provides the probability of going from state i to state j in one time-step. Then the n-step transition matrix can be determined by $M^{(n)} = (m_{ij})^n$. Thus, we can predict the state distribution of S_{t_n} by

$$P\left\{S_{t_n} = (i_n, j_n) \,|S_{t_0} = (i_0, j_0)\right\} = S_{t_0} M^n. \tag{3}$$

Fig. 7 illustrates the variation of the probability distribution of states at particular time intervals, where the Markov chain provides the possible states at each time step. The distribution spreads out because the number of entries having non-zero probability grows gradually with each transition, as time increases.

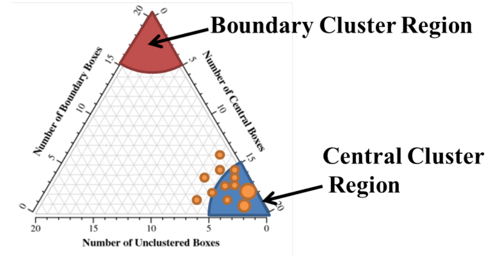


Fig. 8: An example of the clustering result of a sequence having high performance.

C. Selecting sequence of strategies

Having constructed a Markov chain model that predicts clustering task performance for each of the twister vs. digger mixes, we now turn to selection of the sequence of labor mixes which achieves the best performance of the clustering task; we seek a sequence that is both reliable and efficient. As a proof of this concept in this paper, we consider the most basic sequence comprised of only two mixes but, as will be seen below, more complex varieties follow the same procedure directly.

From the Markov chain Theorem, the state distribution of the sequence of two mixed strategies after n time-steps where they switch at time k is

$$M_{seq} = \left[M_A\right]^k \left[M_B\right]^{n-k}, \tag{4}$$

where M_A and M_B are the transition matrices of labor mix A and labor mix B, respectively, and k is the time at when the strategy is switched where $0 \leq k \leq n$. With Eqs. 3 and 4, the probability of a configuration during the clustering task, given the switching time, can be predicted. For example, if the initial configuration is $(0, 0)$ in which is no boxes in the central clusters or the boundary clusters, the initial vector, $X_0 = [1, 0, \cdots, 0]$, here X_0 has size $1 \times d$. That is, the probability distribution of the final state after n time-steps can be computed by $X_0 M_{seq}$. We can use the probability distribution of the final states to determine the best strategy for the clustering task.

To quantify the clustering performance, we introduced a performance metric. Given an initial configuration, a perfect central cluster has state $(N_0, 0)$, and ought to be assigned a high weighting factor for quantifying the clustering performance. Smoothing this function, weights are assigned up to clusters composed of more than 90% boxes in a central cluster. For example, since we use 20 boxes in our experiment, we consider up to the states, (18,0), (18,1), and (18,2) for measuring the clustering performance. Let $P\left\{S_{t_{180}} = (i, j) | S_{t_0} = (0, 0)\right\}$ be $M_{seq}(i, j)$. Then, the performance metric is defined as follows.

$$Performance\ Metric = \sum_{u=\lfloor 0.9\times N_0 \rfloor}^{N_0} \frac{u}{N_0} \sum_{v=0}^{N_0-u} M_{seq}(u,v). \quad (5)$$

Fig. 8 provides an example of a result from a sequence having high performance. For example, as shown in Fig. 8, if the boxes are located in the central region of the arena at the final time step, we can assume that the sequence of mixed strategies produce a good result with high probably.

IV. Physical robot experiments

We first describe how the Markov chain model is built based on data obtained from calibration runs. Next, we validated our Markov chain model by comparing the model prediction with physical robot experiments.

A. The Markov chain model

In order to build the Markov chain model, we conducted a calibrated run for all possible combination of Twister(T) and Digger(D) with 5 robots: 0T5D, 1T4D, 2T3D, 3T2D, 4T1D, and 5T0D. Each trial lasted 90 minutes, with 20 boxes. All experiments were videotaped and annotated with $n = 180$ moments by observing frames every 30 seconds. For each division of labor, a total number of 540 transitions between states was observed. From this we obtain the state transition matrix.

In order to find the best sequence of strategies having the maximum clustering performance, we compared the performance by varying the switching time from $k = 0$ to $k = 180$. Fig. 9 shows the performance metric for all sequence of strategies. The Markov chain model predicts that the best sequence of strategies was switching from 2T3D to 0T5D, and it outperformed the clustering performance of a single strategy between 22 to 89 min. The switching sequences of 2T3D→1T4D and 1T4D→0T5D also outperformed the clustering performance of a single strategy between 55 to 89 min, and between 17 to 89 min, respectively. Note that the end points of each line shows the performance of a pure strategy where no switching occurs.

B. Model validation

The model suggests that the best strategy is a sequence where 2T3D is switched to 0T5D after 25 minutes (shown in Fig. 9). We examined the ordering by comparing the clustering performance predicted by the Markov chain with an actual experiment. We selected the two best sequences: the sequence from 2T3D to 0T5D and the sequence from 2T3D to 1T4D, and carried out physical experiments for both cases, switching at 25 minutes. Each set of sequences was conducted five times under the same initial configuration with 5 robots and 20 boxes. We assumed that the average size of a single central cluster at the final step, 180 time-step, is a good measurement of the task performance.

Table II shows the average size of a single central cluster at the final step in each sequence, and represents the clustering results of each experiments after 90 minutes. Since the average size of a single central cluster in the sequence from 2T3D to 0T5D is larger than the average size of the sequence from 2T3D to 1T4D, the result supports that idea that the ordering predicted from the model at 25 minutes, is also reflected in physical experiments.

This observation is further confirmed with a statistical test. We assumed that the gap in performance between two sequences in physical experiment is identical to the difference of the statistical mean. In order to test a statistical hypothesis, we conducted the two-sample t-test with unequal variance based on experimental data. The two-sample t-test is used to determine if two population means are equal or not. The two-sample t-test is defined as

$$H_0 : \mu_a = \mu_b,$$
$$H_a : \mu\ are\ not\ equal.$$

We then select the level of significance to be used in the test as 0.05. After performing the hypothesis test, we could get the P-value. Since the P-value, 0.0425, is below 0.05 in one-tailed test, we can reject the null hypothesis of no difference between the means from the two samples in favor of the alternative. In other words, we accept that the mean of the size of a single central cluster between the sequence from 2T3D to 0T5D and the sequence from 2T3D to 1T4D are unequal with 95% confidence. That is, the difference of the performance between two sequences in physical experiments has significant difference. Consequently, the ordering of the clustering performance predicted by the Markov chain model corresponds to the ordering of the clustering performance by physical robots.

TABLE II: Experimental results and Two-sample t test.

		2T3D → 0T5D (Switching @25min)	2T3D→ 1T4D (Switching @25min)
Trials	1	18 boxes	11 boxes
	2	19 boxes	20 boxes
	3	19 boxes	9 boxes
	4	18 boxes	17 boxes
	5	20 boxes	14 boxes
Average size of a single central cluster at the final step		18.8 boxes	14.2 boxes
$P(T \leq t)$ one-tail		0.0425	

V. Conclusion

We conclude that the Markov chain model is useful at predict the performance of self-organized robots performing an object clustering task, and the model permits planning of a sequence of changes to the division of labor. The experiments suggest that the model's predictions of the relative performance of different switched strategies of the labor mix are useful for reasoning about the performance of real robots.

This work showed that a sequence of one division of labor followed by another improves clustering performance over a single strategy. Future work will include sequences of more

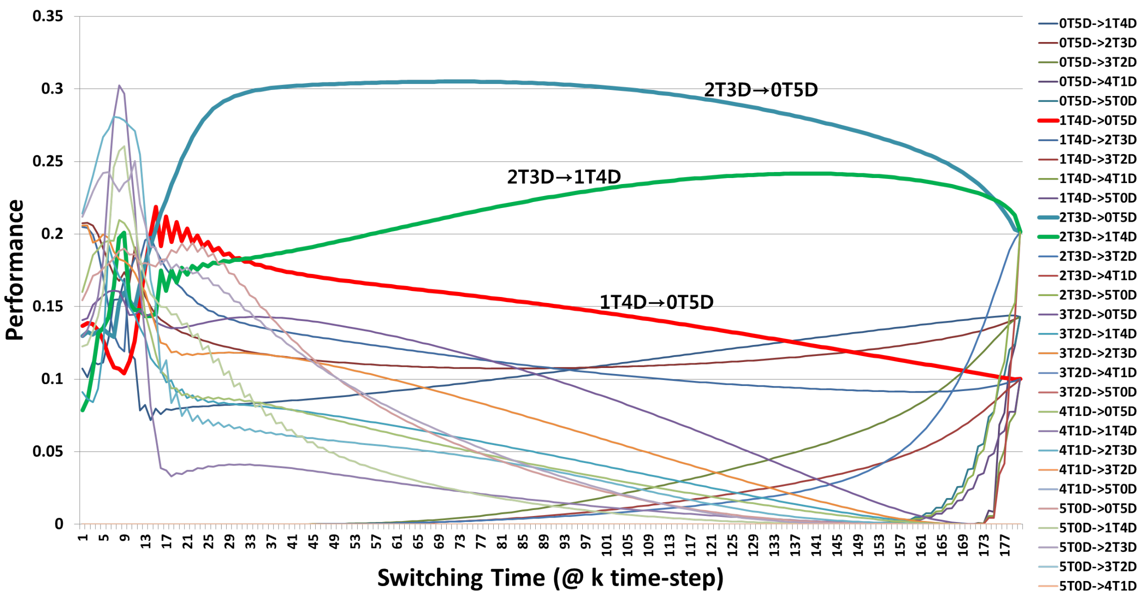


Fig. 9: A comparison of predicted performance via the Markov chain model with varying switching time.

than two strategies. In addition, since the results verify the utility of the stochastic model with the object clustering, we plan to extend our method to adapt to different self-organized cooperative systems such as a transportation tasks or a monitoring task system.

REFERENCES

[1] L. E. Parker, "Multiple Mobile Robot Systems," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 40.

[2] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127–141, Sept. 2002.

[3] A. Lein and R. T. Vaughan, "Adaptive multirobot bucket brigade foraging," in *In Proceedings of the Eleventh International Conference on Artificial Life (ALife XI)*. MIT Press, 2008, pp. 337–342.

[4] J.-H. Kim, Y. Song, and D. Shell, "Robot spatial distribution and boundary effects matter in puck clustering," Mar. 2011, AAAI Spring Symposium Series: Multirobot Systems and Physical Data Structure.

[5] Y. Song, J.-H. Kim, and D. A. Shell, "Self-organized clustering of square objects by multiple robots," in *Proceedings of the 8th international conference on Swarm Intelligence*, ser. ANTS'12. Springer-Verlag, 2012, pp. 308–315.

[6] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "The dynamics of collective sorting robot-like ants and ant-like robots," in *Proc. of Simulation of Adaptive Behavior (SAB)*, 1991, pp. 356–363.

[7] R. Beckers, O. Holland, and J. Deneubourg, "From Local Actions to Global Tasks: Stigmergy and Collective Robotics," in *Proc. of Artificial Life IV*, 1994, pp. 181–189.

[8] M. Maris and R. Boeckhorst, "Exploiting physical constraints: Heap formation through behavioral error in a group of robots," in *Proc. of Conference on Intelligent Robots and Systems (IROS)*, 1996, pp. 1655–1660.

[9] O. Holland and C. Melhuish, "Stigmergy, self-organization, and sorting in collective robotics," *Artif. Life*, vol. 5, no. 2, pp. 173–202, 1999.

[10] K. Lerman and A. Galstyan, "Two paradigms for the design of artificial collectives," in *Collectives and the Design of Complex Systems*, K. Tumer and D. Wolpert, Eds. Springer New York, 2004, pp. 231–256.

[11] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *Int. Journal of Robotics Research*, vol. 23, pp. 415–436, 2004.

[12] K. Lerman, A. Galstyan, A. Martinoli, and A. J. Ijspeert, "A macroscopic analytical model of collaboration in distributed robotic systems," *Artificial Life*, vol. 7, pp. 375–393, 2001.

[13] W. Agassounon, A. Martinoli, and K. Easton, "Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes," *Auton. Robots*, vol. 17, no. 2-3, pp. 163–192, Sept. 2004.

[14] A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots," *Robotics and Autonomous Systems*, vol. 29, no. 1, pp. 51–63, 1999.

[15] S. Kazadi, A. Abdul-Khaliq, and R. Goodman, "On the convergence of puck clustering systems," *Robotics and Autonomous Systems*, vol. 38, no. 2, pp. 93–117, 2002.

[16] S. M. Ross, *Introduction to Probability Models, Eighth Edition*, 8th ed. Academic Press, Jan. 2003.