Distributed Coordination of Simple Earthmover Robots for Terrain Modification

Jiahe Chen[1] and Kirstin Petersen[1]

***Abstract*— Toward autonomous robotic terrain modification in inaccessible environments, such as the moon or Mars, we investigate a distributed coordination algorithm based on Wasserstein geodesics for swarms of minimal earthmover robots. We use a simulation framework designed around a hardware instantiation of such robots, and investigate the efficacy of this algorithm on up to 9 agents working towards 100 randomly generated construction tasks. We find that our system has a success rate $\sim$90%, comparable to a centralized controller with global information when working on traversable terrain. More robots lead to higher convergence rates, faster convergence times, and less overall volume of moved material, although congestion can lead to longer cumulative distances traveled. The results indicate an interesting alternative to deployment of more sophisticated construction robots for terrain modification.**

I. Introduction

In inaccessible or harsh desert environments such as on the moon or Mars, it is desirable to have robots pave the way for future human involvement, e.g. by preparing landing sites, berms, or access paths [1], [2]. Such systems must operate autonomously and reliably in dusty and high-contrast scenes which limit the accuracy of state-of-the-art sensors like LiDARs and cameras, work under high radiation exposure which can affect processors and memory, and locomote over highly irregular terrain which may entrap classic wheeled, threaded, or legged ground vehicles. Moreover, the task may call for deployment of many robots over time, as construction challenges change and as funding and trust in the system grows, which mean that the coordination algorithms must be equally agnostic to the group size.

In past work [3], we introduced a starkly different approach to these high-end construction robots. Our suggested approach relies on large swarms of inexpensive, distributed robots with minimal actuation and sensing. Our robot design was inspired by a single-actuated wave mechanism first introduced in [4] which simply creates a traveling wave using a fixed helix surrounded by a chain of links. The robot consists of two such tails, such that when the tails are operated at different speed the robot moves while also moving granules underneath (Fig. 1). While this system is currently restricted to operation in the lab in a 2D vertical arena and uses packing peanuts as a proxy for sand, it has been shown capable of motion over granules, shuffling and transporting granules, and digging into granules. In [5], we worked in simulation to showcase the potential of such a system, introducing a centralized algorithm with global sensing to control multiple agents to perform terrain modifications from arbitrary, user-specified start and goal structures. Our algorithm was informed by Wasserstein distance, also known as the Earthmovers' distance. In this paper, we further that work by introducing a distributed Wasserstein-geodesics-based algorithm that can work with local sensor information to make a swarm of such simple agents perform desired terrain modification. Compared with the centralized planner, the distributed algorithm solely relies on sensing the steepness of the local structure in small range and inferring the x-coordinate of robot location, which serves as a big step towards real deployment. We investigate the efficacy of the proposed algorithm and find that deployment of more agents help increase the system efficiency at the expense of longer overall traveling distance. Comparison with the results of centralized algorithm shows that two algorithms work equally well on tasks with traversable start and goal terrain but the centralized algorithm has higher efficiency overall due to the global sensing and perfect planner-to-agent communication.

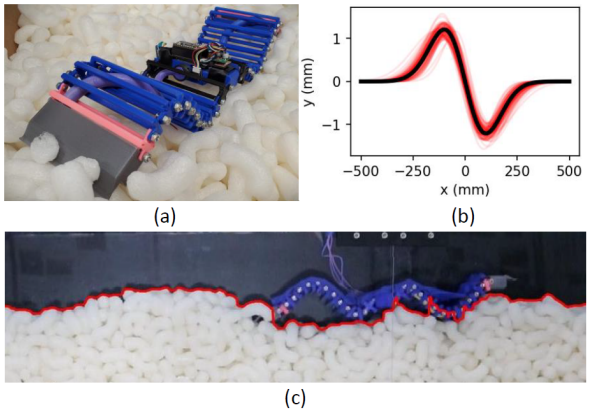

Fig. 1: (a) A double-tailed SAW robot on granules. (b) Construction function without noise (black line) and with noise (red line). (c) Automated surface tracking of a SAW robot pushing material forward.

Other related works on distributed collective robotic construction include [6] where a group of robots build a user-define 3D structure on a flat surface using customized solid bricks and [7] where robots build functional structures like a navigable ramp in unstructured environment using amor-

This work was funded by a Packard Fellowship for Science and Engineering and NSF #2042411.

[1]School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853, USA jc3472@cornell.edu, kirstin@cornell.edu

Manuscript 3728 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 16, 2024.

phous material. our work similarly targets construction of amorphous continuous structures but with a broader input as structures with arbitrary shapes can be given and a stricter output as a user-define structure is pursued.

The rest of the paper is structured as follows. In Sec. II, we recap the abstract model for terrain modification. We then review the optimal transport theory and Wasserstein geodesics and propose the distributed coordination algorithm that utilizes Wasserstein geodesics to decompose the construction problem into local subtasks that are easier to solve in Sec. III. The proposed algorithm is tested with 100 randomly generated tasks and the result is thoroughly examined in Sec. IV to characterize the efficacy of the algorithm.

II. Model

In this section, we briefly recap the abstract model for 2D construction with granular material, described in detail in our previous work [5]. The shape of the structure is described by a continuous, bounded, non-negative height function, $h : Q \to \mathbb{R}^+$, where $Q = [-L/2, L/2]$ is the construction area with length L. A valid height function must satisfy the steepness constraint $\max |h'(x)| \leq K$ due to the physical properties of granular material. We assume that the robot can freely move over the surface of the structure without affecting its shape. However, due to the motion constraint, the robot cannot climb up a hill with steepness greater than $K_R = 0.3$ otherwise it will get stuck. The robot can descend a hill, but cannot go back to its previous location after having descended. In simulation, robot will only travel to the current reachable area $Q_R \in Q$. The real robot has two construction modes. In one, the robot uses one tail to keep it stationary and another tail to push material forward. In the other, robot moves while manipulating the material. We focus on the first mode to decouple the locomotion and manipulation. The robot interaction with the structure is modeled by a construction function $f : Q \to \mathbb{R}$. Given an action location $\phi \in Q$, and a direction of pushing the material $\theta \in \{\text{left, right}\}$, the construction function of pushing material for 1 second is a first Gaussian derivative:

$$f_{\phi,\theta}(x) = \frac{z(\theta)a}{\sqrt{2\pi w^2}}(x-\phi)e^{-\frac{(x-\phi)^2}{2w^2}}, \tag{1}$$

where $a \neq 0$ and $w > 0$ are shape parameters set to fit the real robot behavior, and $z(\text{right}) = 1, z(\text{left}) = -1$. a and w are obtained through a curve fitting process described in [5]. To mimic the robot's action noise in reality, we add Gaussian noise to the shape parameters and action position. The construction function with noise is:

$$\hat{f}_{\phi,\theta}(x) = \frac{z(\theta)\hat{a}}{\sqrt{2\pi\hat{w}^2}}(x-\hat{\phi})e^{-\frac{(x-\hat{\phi})^2}{2\hat{w}^2}}, \tag{2}$$

where $\hat{a} \sim \mathcal{N}(a, a/10)$, $\hat{w} \sim \mathcal{N}(w, w/10)$, and $\hat{\phi} \sim \mathcal{N}(\phi, 10)$. These noise parameters are taken empirically. Let parentheses $(\cdot)$ denote the function application to points and brackets $[\cdot]$ denote the operator application to functions. Given a current structure h and a construction function f, the new structure after modification is:

$$M[f, h](x) = h(x) + \alpha\hat{f}(x), \tag{3}$$

where $\alpha \in [0, 1]$ depending on $\hat{f}$ and h is an attenuation factor that enforces the steepness constraint. Given an initial structure h_0, and a sequence of actions characterized by the position and direction parameters $(\phi_1, \theta_1), \ldots (\phi_n, \theta_n)$, the final structure after n actions is:

$$M[f_{\phi_n,\theta_n}, \ldots M[f_{\phi_1,\theta_1}, h_0]] = h_0(x) + \sum_{i=1}^{n} \alpha_i \hat{f}_{\phi_i,\theta_i}(x) \tag{4}$$

III. Algorithm

The construction problem can be formulated as follows based on Eq. 4. Given an initial and goal structure h_0 and h_1, how can we design a robot policy that can modify h_0 into a structure that is as close to h_1 as possible? In this section, we present a fully distributed coordination algorithm that abstracts the construction problem into multiple local subtasks by computing the intermediate local goal structures based on Wasserstein geodesics. These subtasks prove to be much easier to solve [5] and when multiple robots are deployed to solve them the structure can can converge to the goal structure with high probability as show in Sec. IV. The remainder of this section will review Wasserstein geodesics and the centralized planning algorithm.

A. Wasserstein Geodesics

Wasserstein metric and Wasserstein geodesics originate from the optimal transport theory which has been widely used in statistics, machine learning, and image processing. [8], [9], [10]. To apply optimal transport theory to our problem, we start by converting any two given structures h_0 and h_1 to probability distribution functions I_0 and I_1 by dividing by the integral over Q. We also assume that two structures have the same integral v. A transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ tells you where to move the mass from one location to another so that I_0 can morph into I_1. For a valid transport map, the amount of mass moved out of an area must be the amount of mass moved into another area. Therefore the measure-preserving (MP) requirement must be met:

$$\int_{T^{-1}(A)} I_0(x)\,dx = \int_A I_1(x)\,dx \tag{5}$$

If the cost of moving a unit mass is measured by a convex distance function $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, then the minimum cost based on Monge formulation is:

$$M(I_0, I_1) = \inf_{T \in MP} \int_{\mathbb{R}^n} d(x, T(x)) I_0(x) dx \tag{6}$$

The minimizer T^* is called the optimal transport map. The 2-Wasserstein distance is defined as:

$$W_2(I_0, I_1) = \left(\inf_{T \in MP} \int_{\mathbb{R}^n} |x - T(x)|^2 I_0(x) dx \right)^{1/2}. \tag{7}$$

Let $P(Q)$ be the set of probability densities supported on Q, then the metric space $(P(Q), W_2)$ is referred to as the 2-Wasserstein space. The 2-Wasserstein geodesics $I_t, t \in [0, 1]$

is the shortest path connecting I_0 and I_1 in 2-Wasserstein space:

$$\begin{aligned} I_t(x) &= \det(DT_t^{-1}(x))I_0(T_t^{-1}(x)), \\ T_t(x) &= (1-t)x + tT^*(x). \end{aligned} \tag{8}$$

In the case of 1D, the optimal transport map T^* has a closed-form solution: $T^*(x) = F_1^{-1} \circ F_0(x)$, where F_0 and F_1 are cumulative distribution functions of I_0 and I_1. This also leads to a closed-form equation of the 2-Wasserstein distance:

$$W_2(I_0, I_1) = \left(\int_0^1 \left| F_0^{-1}(t) - F_1^{-1}(t) \right|^2 dt \right)^{\frac{1}{2}}. \tag{9}$$

If we need to apply optimal transport theory to construction problem in 3D, then T^* in 2D is needed which can be estimated by numerical methods such as flow minimization, gradient descent, linear programming, and entropy regularization [11], [12], [13], [14]. Based on Eq. 8, we construct an operator for generating an intermediate structure at point t between h_0 and h_1.

$$\begin{aligned} G_t[h_0, h_1](x) &= \frac{h_0(T_t^{-1}(x))}{T_t'(T_t^{-1}(x))}, \\ T_t(x) &= (1-t)x + tF_1^{-1} \circ F_0(x). \end{aligned} \tag{10}$$

Let D_{W_2} be the distance function based on 2-Wasserstein distance and D_{L_2} be the distance function based on Euclidean distance. Compared with D_{L_2}, D_{W_2} captures the geometry of the data and can more accurately measure the progress towards the goal. This is better illustrated in Fig. 2. However, D_{L_2} is much easier to compute and will be more frequently used to make step-wise decisions in the planning algorithm.

$$D_{W_2}(h_0, h_1) = W_2(h_0/v, h_1/v), \tag{11}$$

$$D_{L_2}(h_0, h_1) = \left(\int_Q |h_0(x) - h_1(x)|^2 dx \right)^{\frac{1}{2}}. \tag{12}$$

Given the initial structure h_0, goal structure h_1, and current structure h, the construction progress is $1 - D_{W_2}(h, h_1)/D_{W_2}(h_0, h_1)$. Notice that by this definition, the construction progress could be negative when h is even more distant from h_1 than h_0.

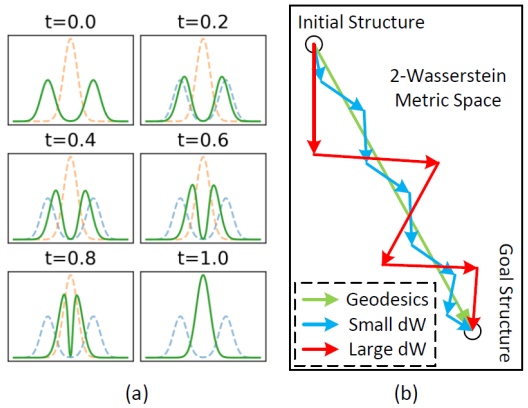

Fig. 2: (a) Wasserstein geodesics (green solid line) from h_0 (blue dotted line) to h_1 (red dotted line). (b) Effect of dW.

B. Distributed Coordination Algorithm

The distributed algorithm is shown in Alg. 1. We assume that the robot only knows its current position x_r and the steepness of the local structure within the range of its own length l_r. The availability of these sensor inputs matches the most recent hardware advances for the robot in the lab and it is reasonable to assume that the actual robots have access to certain location reference which grounds where the construction takes place and they can use the IMU data to infer the structure steepness. Given the current structure h, the shape of the local structure $\tilde{h}(x), x \in [x_r - l_r/2, x_r + l_r/2]$ can be reconstructed:

$$\tilde{h}(x) = \int_{x_r - l_r/2}^{x} h'(u)\, du. \tag{13}$$

Since the altitude of the structure is unknown to the robot, $\tilde{h}$ has an unknown offset to h. We also assume that the robot knows the goal structure h_1 and the local view of the goal structure $\tilde{h_1}$ corresponding to its current position. There is no long-range communication among robots but the robot can sense other robots' presence when in close distance. Once robots are deployed, the robot swarm spreads out to more efficiently explore the structure and avoid interference with each other. When a robot has isolated itself from other robots, the algorithm computes the Wasserstein geodesics from the current local structure $\tilde{h}$ to the goal structure $\tilde{h_1}$ and attempts to move the local structure by dW along the geodesics (line 7-8). Smaller dW is preferred since it leads to simpler subtasks and the outcome does not deviates significantly from the expectation. To reach the intermediate goal structure $\tilde{h_t}$, the algorithm uses a simple greedy approach to find an action plan (line 9-15). Once the plan is executed, the robot goes to the next location.

C. Centralized Planning Algorithm

The core part of the centralized algorithm from [5] is shown in Alg. 2 for comparison. Key differences include known global structure h and perfect communication between the central planner and robot swarm. Essentially in this system agents are used as untethered, distributed actuators.

IV. Results / Characterization

A. Simulation

We generated random structures based on a mixed Gaussian model to form a set of construction tasks to test the algorithm. A construction task is defined by a tuple (Q, h_0, h_1) where Q is the construction area and h_0, h_1 are valid height functions of initial and goal structures defined over Q. h_0 and h_1 must have the same integral over Q as well. The random height function is given by:

$$h(x) = \sum_{i=1}^{m} a_i e^{-\frac{(x-b_i)^2}{2c_i^2}}, \tag{14}$$

where $m \in Z, a_i \in \mathbb{R}, b_i \in \mathbb{R}^+, c_i \in \mathbb{R}$ are random variables drawn from uniform distributions. We generated 100 construction tasks with traversable initial and goal

Algorithm 1: Distributed multi-robot coordination algorithm based on Wasserstein geodesics.

Input: construction area Q, initial and goal structure h_0 and h_1, desired distance to move in every subtask dW, number of robots N, initial positions of each robot X_0, time limit T.

Set structure and robot positions $h \leftarrow h_0$, $X \leftarrow X_0$
Set action plan of each robot $S_i \leftarrow \emptyset$
repeat
  for *each robot* i **do**
    if $S_i = \emptyset$ **then**
      if *no robot nearby* **then**
        $t \leftarrow \min\{1, dW/D_{W_2}(\tilde{h}, \tilde{h_1})\}$
        $\tilde{h_t} \leftarrow G_t[\tilde{h}, \tilde{h_1}]$
        $A_i \leftarrow 0$
        repeat
          $\phi^*, \theta^* \leftarrow \operatorname{argmin}_{\phi,\theta} D_{L_2}(\tilde{h} + A_i + f_{\phi,\theta}, \tilde{h_t})$
          $L \leftarrow D_{L_2}(\tilde{h_t} + A_i + f_{\phi^*,\theta^*}, \tilde{h_t})$
          $A_i \leftarrow A_i + f_{\phi^*,\theta^*}$
          Add (ϕ^*, θ^*) to S
        until L *stops decreasing*;
      else
        Randomly walk on the structure
      end
    else
      Execute and remove an action from S_i
    end
  end
until *time limit* T *is reached*;

Algorithm 2: Centralized multi-Robot planning algorithm based on Wasserstein geodesics.

Input: Same as Alg. 1.

repeat
  $W \leftarrow D_{W_2}(h, h_1)$
  $t \leftarrow \min\{1, dW/W\}$
  $h_t \leftarrow G_t[h, h_1]$
  $A \leftarrow 0, S \leftarrow \emptyset$
  repeat
    $\phi^*, \theta^* \leftarrow \operatorname{argmin}_{\phi,\theta} D_{L_2}(h + A + f_{\phi,\theta}, h_t)$
    $L \leftarrow D_{L_2}(h + A + f_{\phi^*,\theta^*}, h_t)$
    $A \leftarrow A + f_{\phi^*,\theta^*}$
    Add (ϕ^*, θ^*) to S
  until L *stops decreasing*;
  for *each action* ϕ_j, θ_j *in* S **do**
    if ϕ_j *is reachable by any robot* **then**
      Execute the action with robot i s.t. $|x_i - \phi_j|$ is minimized
    end
  end
until W *stops decreasing or time limit* T *is reached*;

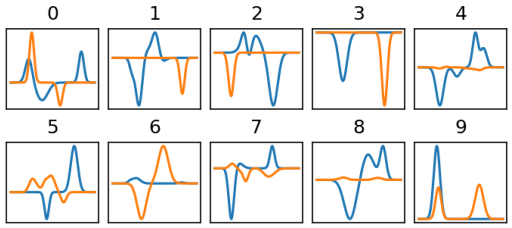

Fig. 3: Example of 10 tasks from the 100 randomly generated construction tasks. Initial structure h_0 (blue line) and a goal structure h_1 (orange line).

structure on a fixed Q that is 5 m (13 times the robot length) wide. The Wasserstein distance of h_0 and h_1 is at least 2 so that the difference is significant enough. 10 randomly selected tasks are shown in Fig. 3. These tasks do not have practical purposes, but are used to test the cross-task algorithm performance. For the simulation setup, each randomly generated task is simulated 5 times. The average Wasserstein distance between initial and goal structure is around 5 and the average Wasserstein distance between the local structure and local goal structure is around 1. Therefore we should use a smaller dW for the distributed algorithm. We set $dW = 0.25$ for distributed algorithm and $dW = 1$ for centralized algorithm. At the beginning of the construction, robots are uniformly distributed across the structure. The time limit T is set to 10,000 time steps (10,000 seconds).

B. Results

To test the construction performance of the algorithm, we measure several metrics including: achieved construction progress, success rate, construction time, robot traveling distance, and volume of moved material. We define that a construction task is completed when 90% construction progress is reached. Success rate is the number of successful simulations divided by the total number of simulations (500 in our case). Construction time is the time to reach 90% structure completion. Therefore it can only be measured for the ones that reach at least 90% of the goal. To eliminate the effect of the variance in the difficulty and size of random tasks, we normalize the construction time using Wasserstein distance between initial and goal structure. The same approach is applied to the robot traveling distance which measures the total distance robots traveled to complete construction. In reality, in addition to the energy consumed by locomotion, robot traversals may affect the structure to a small degree and have to be minimized. The volume of moved material is the total volume of material moved by the robot throughout a successful construction process, which is an important indicating factor for energy expenditure and construction efficiency. To eliminate the structural variance, we normalize it using the volume difference between initial and goal structure $\int_Q |h_0(x) - h_1(x)|\, dx$ which can be seen as the ideal least volume of material needs to be moved to complete the construction.

The characterization results of Alg. 1 are shown in Fig. 4. The success rate is 40%, 83%, 88%, 91%, and 90% for 1,

3, 5, 7, and 9 deployed robots respectively. Fig. 4(a) shows that when more than one robot are deployed the construction progress does not change significantly. However, to achieve that similar construction progress, Fig. 4(b, d) shows it takes less time and requires less material to be moved when more robots are deployed. Furthermore, Fig. 4(c) shows that when more robots are deployed, the traveling distance increases. This result is different from the one of the centralized algorithm shown in Fig. 7(c), and is caused by the crowding effect. Since robots tend to spread out to avoid interference, when there are too many on site, robots travels more to avoid each other.

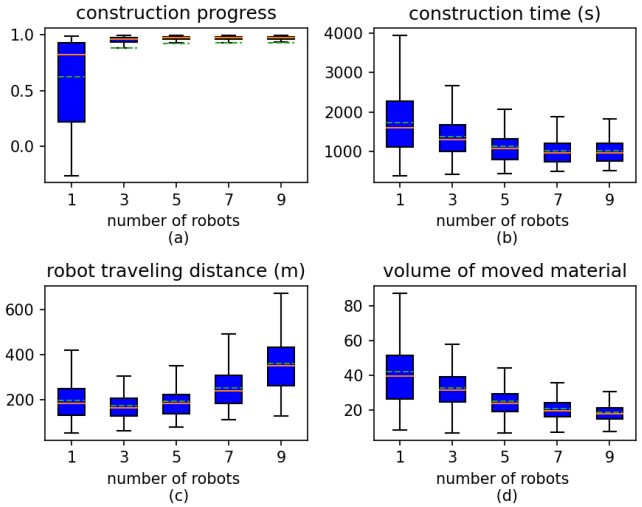

Fig. 4: Simulation results of constructing random tasks in Fig. 3 by Alg. 1. For each boxplot, medium is orange solid line and mean is green dashed line. (a) Achieved construction progress. (b) Normalized construction time. (c) Normalized robot traveling distance throughout construction. (d) Normalized volume of moved material throughout construction.

A major reason for unsuccessful construction is the untraversable structure created amid construction. Even though both the initial and goal structure are traversable, an untraversable structure can still emerge due to the limited local information available to the robots. Fig. 5 illustrates an example. Fig. 5(a) shows that untraversable structure can emerge at any time throughout the construction process. Upon the occurrence, part of the structure may be temporarily inaccessible for some robots, which may cause the construction to stall. Such phenomenon rarely happens for centralized algorithm due to the access to global information. Among all simulations, the total number of simulations where robots create untraversable structures is 411, 362, 334, 339, and 310 for 1, 3, 5, 7, and 9 robots. Among them, 35%, 77%, 84%, 87%, and 85% simulations are successful, which shows the resilience of Alg. 1 in overcoming untraversable structures.

Wrong decisions or bad actions can be recognized by the change of progress. A bad action causes decrease in progress. Bad actions emerge when the local intermediate

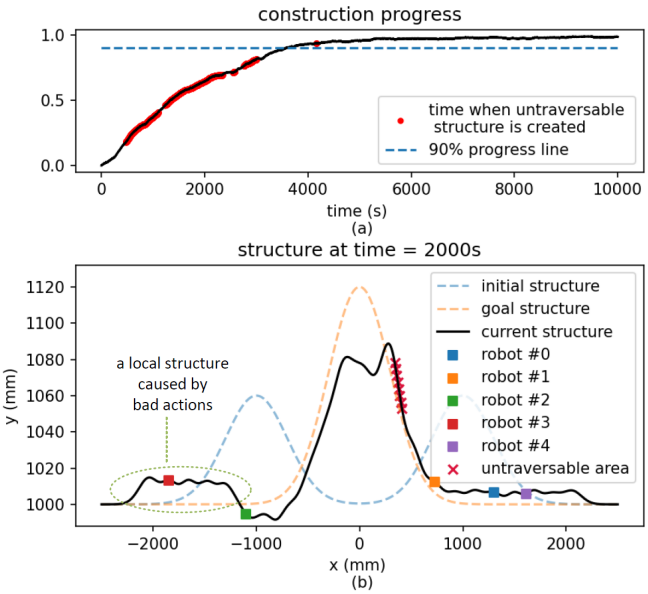

Fig. 5: (a) Construction progress over time for constructing the task shown in Fig. 2(a) by Alg. 1 with 5 robots. The moment when untraversable structure is created is marked by red cross. (b) The state of constructing the task at 2000s.

goal structure directs the robot to push material in wrong directions. An example can be seen in Fig. 5(b), where robots attempt to combine two mounds on each side to form a larger mound in the center. In the circled area, the robot created a flat structure to match the shape of the goal locally. By doing so the robot made wrong decision to push material to the left side, not to the center and caused decrease in progress. The remedy for this issue is to deploy more robots. Fig. 6(b) shows that when more robots are deployed, robots are less likely to take bad actions. This happens possibly because when multiple robots positioned across the structure modify the structure simultaneously, the overall construction process would be similar to modifying the structure based on the global Wasserstein geodesics, which is exactly what the centralized algorithm performs. This can be seen in Fig. 6(a), which shows that as more robots are deployed, the time evolution of construction progress of distributed algorithm approaches the one of centralized algorithm. The reduction in likelihood of taking bad actions explains why the construction process is more stable and rises to completion more quickly when there are more robots in Fig. 6(a). It also explains why untraversable structures emerge less frequently as number of robots increases.

Fig. 7 shows simulations results for centralized algorithm. The success rate is 63%, 65%, 67%, 69%, and 70% for 1, 3, 5, 7, and 9 robots respectively. Fig. 7(a) shows that when more than one robot is deployed, the distributed algorithm reaches a similar construction progress, but has a higher success rate. The lower success rate of centralized algorithm is likely caused by the early stopping when the algorithm could not find a way to move towards the goal (Alg. 2 (line 17)). Fig. 7(b-d) shows that centralized algorithm is

Manuscript 3728 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 16, 2024.

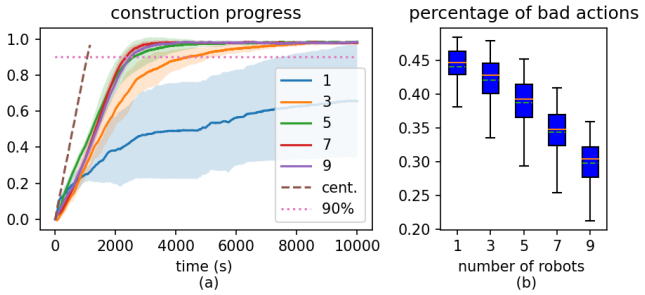

Fig. 6: (a) Construction progress over time for constructing the task shown in Fig. 2(a) by Alg. 1 with different number of robots (solid line) and by Alg. 2 (dashed line). Each setting is simulated by 10 times. (b) Percentage of bad actions among all actions for construction random tasks in Fig. 3 by Alg. 1. Medium is orange solid line and mean is green dashed line.

much more efficient and stable in terms of construction time, traveling distance and volume of moved material due to the more optimal planning based on global information and perfect communication.

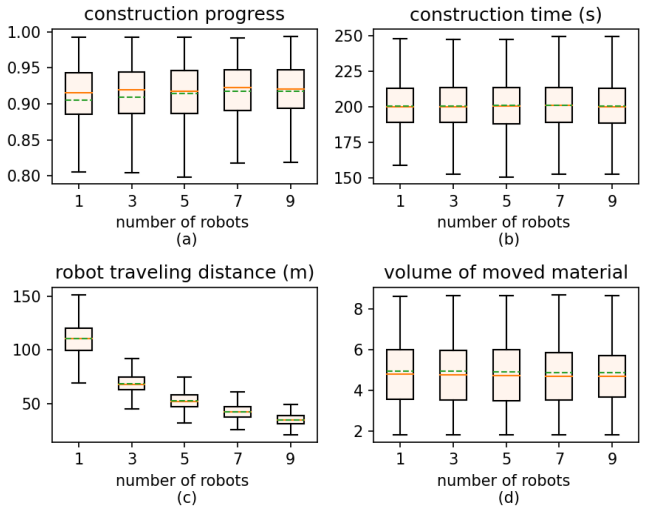

Fig. 7: Simulation results of constructing random tasks in Fig. 3 by Alg. 2. For each boxplot, medium is orange solid line and mean is green dashed line. (a) Achieved construction progress. (b) Normalized construction time. (c) Normalized robot traveling distance throughout construction. (d) Normalized volume of moved material throughout construction.

V. Discussion

A. Conclusion

In this paper we introduced a distributed algorithm based on Wasserstein geodesics, and showed how it can be used to coordinate user-specified terrain modifications in a 2D simulated environment by simple earth mover agents restricted to local sensing. We assumed minimal agents which know only their goal, the absolute distance to a reference point, the local slope of the terrain, and the presence of nearby agents; they do not know the overall state of the structure, nor the number of robots deployed.

Under the assumption that both the start and end goal involve structures which are traversable by the agents, we found that our distributed algorithm running on three or more agents could perform almost as well as a centralized algorithm with global sensors, also based on the Wasserstein geodesics. This is interesting, since these locally-informed agents may temporarily create untraversable structures and take actions that decrease the overall progress towards the goal. We found that beyond three agents, the convergence rate was not significantly improved, and crowding on occasion led to longer convergence times, however, the excess volume of material moved decreased, as did the frequency of bad actions. Notably, these results were achieved with significant action noise.

Our results indicate that high rates of success in these construction tasks may be feasible, even with the omission of global sensors and controllers, which can be disadvantageous due to adding significant complexity and cost, representing single point of failures, and imposing bandwidth limitations when many robots are deployed. While our algorithms are designed for the minimalist mechanisms for granule modifications described in the introduction, the algorithm works for any multi-robot system which perform non-additive terrain modification.

B. Future Work

There are many avenues for future work. Obviously, it would be interesting to find ways to improve the construction efficiency. Expanding on the work in this paper, we are investigating the potential for agents to start from and target untraversable structures, as this was shown to be possible using a centralized algorithm [5]. In cursory investigations, we found instances in which the agents with distributed coordination were capable of modifying traversable structures to untraversable, but we have yet to understand what characterizes such scenarios. Finally, our simulations would benefit from including more realistic physics, such as the impact of agent locomotion over the structure and how the convergence results would be affected by lowering the signal-to-noise ratio of the sensors.

References

[1] P. Desai, C. A. Galica, and N. Werkheiser, "Nasa lunar surface innovation initiative: Ensuring a cohesive, executable strategy for technology," in *International Astronautical Congress*, 2022.

[2] R. W. Moses and R. P. Mueller, "Requirements development framework for lunar in situ surface construction of infrastructure," in *Earth and Space 2021*, 2021, pp. 1141–1155.

[3] L. R. Huang, A. Zhu, K. Wang, D. I. Goldman, A. Ruina, and K. H. Petersen, "Construction and excavation by collaborative double-tailed saw robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3742–3748, 2022.

[4] D. Zarrouk, M. Mann, N. Degani, T. Yehuda, N. Jarbi, and A. Hess, "Single actuator wave-like robot (saw): design, modeling, and experiments," *Bioinspiration & biomimetics*, vol. 11, no. 4, p. 046004, 2016.

[5] J. Chen and K. Petersen, "2d construction planning for swarms of simple earthmover robots," in *The International Symposium on Distributed Autonomous Robotic Systems (DARS)*. Springer, 2024.

[6] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.
[7] N. Napp and R. Nagpal, "Distributed amorphous ramp construction in unstructured environments," *Robotica*, vol. 32, no. 2, pp. 279–290, 2014.
[8] J. Lellmann, D. A. Lorenz, C. Schonlieb, and T. Valkonen, "Imaging with kantorovich–rubinstein discrepancy," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2833–2859, 2014.
[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.
[10] A. Ramdas, N. García Trillos, and M. Cuturi, "On wasserstein two-sample testing and related families of nonparametric tests," *Entropy*, vol. 19, no. 2, p. 47, 2017.
[11] S. Angenent, S. Haker, and A. Tannenbaum, "Minimizing flows for the monge–kantorovich problem," *SIAM journal on mathematical analysis*, vol. 35, no. 1, pp. 61–97, 2003.
[12] R. Chartrand, B. Wohlberg, K. Vixie, and E. Bollt, "A gradient descent solution to the monge-kantorovich problem," *Applied Mathematical Sciences*, vol. 3, no. 22, pp. 1071–1080, 2009.
[13] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013.
[14] A. M. Oberman and Y. Ruan, "An efficient linear programming method for optimal transportation," *arXiv preprint arXiv:1509.03668*, 2015.