2D Construction Planning for Swarms of Simple Earthmover Robots

Jiahe Chen[1] and Kirstin Petersen[1]

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

Abstract. We present a model for simple earth mover robots modifying continuous 2D granular structures and propose a multi-robot planner which can modify structures according to user-defined goals. The proposed planner decomposes a long-horizon construction task into subtasks that are easier to solve using optimal transport theory and Wasserstein geodesics. We test the algorithm with 100 randomly generated construction tasks and demonstrate the algorithm with both single and many robots. The simulations show that the multi-robot algorithm can achieve 85% construction progress on average even under the presence of action noise and motion constraints. In addition, the deployment of multiple robots can reduce the overall traveling distance by 70% compared with the single-robot case, which is important as motion also affects the structure.

1 Introduction

One of the most celebrated use cases of robotics is to prepare a remote, hostile landscape for colonization, whether in the Arctic, the deep sea, or space. Government agencies, planning an imminent return to the moon followed by trips to Mars, are driving innovations in this space, calling for new robot designs capable of using in-situ materials to prepare landing pads, roads, foundations, and berms [1, 2]. State-of-the-art robots towards these goals are typically based on advanced actuation and high-speed communication, and sophisticated sensors [3–7]. However, the extreme conditions of extraterrestrial terrain conditioning poses high risk of clogged mechanisms, dust-covered or saturated sensors, and robot entrapment; hence resilience and long-term autonomy is critical. An interesting alternative is to deploy swarms of simpler, more robust robots that can continue working even in the presence of lost individuals [8].

In previous work [9], we introduced and tested a conceptual design of such simple robots. These robots were based on double-tailed single-actuator wave mechanisms (Fig. 1(a)). The mechanism consists of fixed, mechanical helices surrounded by rectangular chains, such that rotation causes a sinusoidal motion, capable of traversing a number of terrains [10, 11]. Now just by operating the two tails with differing speeds, we demonstrated the ability of the robot to push, transport, degrade, level, compact, and dig into granular materials as well as climb over other robots. While these demonstrations were restricted to simple tethered, 2D manipulation of large granules (packing peanuts), they were especially exciting because of the robustness of the construction behaviors, many performed with open-loop control. We further speculated on several

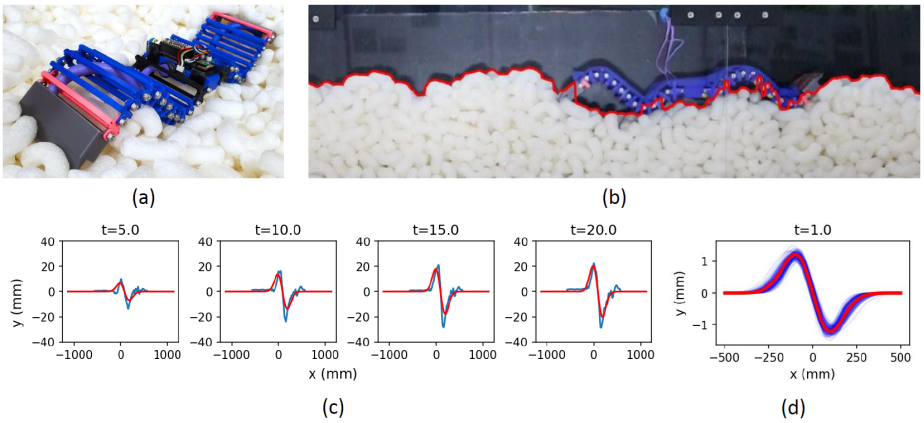

Fig. 1: The construction function (Eq. 2) is obtained through curve fitting with real data. (a) A double-tailed SAW robot on granules. (b) Automated surface tracking of a SAW robot pushing material forward. (c): Fitting the change of surface over time (blue line) with the construction function (red line). (d): Construction function without noise (red line) and with noise (blue line).

pathways towards 3D operation [9]. In this paper, we build on this technology by introducing an abstract mathematical model for the robot mechanism and a construction planner based on Wasserstein geodesics, also aptly known as the earth movers distance.

Over the last few decades, collective robotic construction has emerged as a distinct scientific field with its own unique challenges, methods, and opportunities [12]. Consequently, related work on planners is numerous and matches the wide range of goal structures (buildings, dams, roads, rafts, etc.), building material (bricks, cement, filament, sandbags, sand, etc.), and platform constraints (mobile ground-locked robots, climbing robots, UAVs, etc.). The closest related work stems from [13], which introduced the concept of navigability constraints, permitting simple robots to add material to modify an environment guided only by their own ability to overcome discontinuities and slopes. Later follow-on articles [14, 15], extended upon this work with a variety of building materials, including foam bricks, sandbags, and rocks. Our work similarly targets continuous amorphous structures, but we focus on non-additive construction. Due to the simplicity of our robots, each agent cannot carry or add material from elsewhere, but has to push material around to form a structure. Therefore a new model is needed to describe this modification process. Compared to [13], our construction problem has a broader input (any given initial structure) and a stricter output (a user-defined goal structure), which adds more variations to the construction task and difficulties in defining the problem. However, our system requires global sensing and centralized control. Essentially, we are picturing a construction system with global depth sensors and communication, where the robots are used as untethered, distributed actuators. To address the challenges mentioned above, we first develop an abstract model to describe the non-additive modification (Sec. 2). Based on the model we propose a planning algorithm that utilizes Wasserstein geodesics to decompose a construction problem into subtasks

that are easier to solve (Sec. 3). We explore the algorithm with a single perfect robot (Sec. 4) and then we include realistic factors like motion constraints and traveling distance into consideration (Sec. 5). Finally, we propose a multi-agent planning algorithm to overcome these issues (Sec. 6). Our proposed method is a first step towards landscape preparation with simple and robust multi-robot systems. The concept of using Wasserstein geodesics for subtask abstraction can also be used to help solve other problems involving continuous structures such as the manipulation of deformable objects. [16].

2 Abstract Model

2.1 Problem Formulation

The construction problem is formulated as follows. Given initial and user-defined goal structures, how can we design a robot motion and action strategy such that the shape of the final structure is as close to the goal structure as possible? The real robot has two construction modes. In one, one tail is stationary and keeps the robot in place while the other moves material forward. In the other, locomotion and material manipulation happens simultaneously. Here, we focus on the former one to separate locomotion and manipulation. For now, we assume that the robot can freely move without affecting the structure shape. Real constraints are considered in Sec. 5. We also assume that robots have global sensing and follows plans from a central controller. The remainder of this section presents mathematical models of the continuous structure and modifications, the height function and function operator concepts are inspired by [13].

2.2 Mathematical Model

Model for continuous structures: We model the 2D construction similar to the original robot environment [9]. The construction area Q can be any connected, compact, and finite subset of $\mathbb{R}^1$. Here we set it as an area with length L, $Q = [-L/2, L/2]$. The height function is a continuous, bounded, non-negative function, $h : Q \to \mathbb{R}^+$. The graph of $h, (x, h(x))x \in Q$ describes a structure. Due to the physical properties of granular material, the maximum steepness of the structure cannot exceed a constant $K = 0.42$, informed by real robot experiments [9]. Therefore, the height of a valid structure must satisfy:

$$\max |h'(x)| \leq K \tag{1}$$

Construction function: The robot action on the structure can be modeled by a class of real-valued bounded construction functions $f : Q \to \mathbb{R}$. The integral of a valid construction function over Q must be 0 so that the volume of the structure is preserved: $\int_Q f(x)\, dx = 0$. We use the first Gaussian derivative to model a robot pushing material. Let $Q_f \subset Q$ denote the area where the action can be taken. Given an action location $\phi \in Q_f$, and a direction of pushing the material $\theta \in \{\text{left}, \text{right}\}$, the construction function of pushing material for 1 second is:

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

where $a \neq 0$ and $w > 0$ control the size of the moved mass, and $z(\text{right}) = 1, z(\text{left}) = -1$. a and w are set to fit the real robot behavior. The fitting process is shown in Fig. 1(b)-(d). The area where the action is permitted is set to $Q_f = [-(L/2-5w), (L/2-5w)]$, so that $f_{\phi,\theta}$ is always a valid construction function (assuming $L \gg 10w$). In reality, the robot's action is always noisy. We add Gaussian noise to the shape parameters and action position to mimic this effect. The construction function with noise is:

$$\hat{f}_{\phi,\theta}(x) = \frac{z(\theta)\hat{a}}{\sqrt{2\pi\hat{w}^2}}(x - \hat{\phi})e^{-\frac{(x-\hat{\phi})^2}{2\hat{w}^2}}, \tag{3}$$

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. Even though the noise seems small, the accumulated error effect becomes significant over many actions.

Model for modification: Let $\mathcal{H}$ denote the space of all valid height functions on Q. Let $\mathcal{F}$ denote the space of all valid construction functions. The modification process is modeled by the modification operator $M : \mathcal{H} \times \mathcal{F} \to \mathcal{H}$. 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:

$$M[f, h](x) = h(x) + \alpha\hat{f}(x), \tag{4}$$

where $\alpha \in [0, 1]$ is an attenuation factor that enforces the steepness constraint of the structure (Eq. 1) and $\hat{f}$ is the construction function with noise (Eq. 3). In reality, when the robot keeps digging at the same place and the steepness of the hole exceeds K, the material around will roll back. In addition, the robot has limited weight and cannot displace material heavier than its own weight. Therefore the robot action has a diminishing effect when the steepness of the modified structure approaches K, which is captured by α. Let $x^* = \text{argmax}\,|h'(x) + \hat{f}'(x)|$. α is given by:

$$\alpha = \begin{cases} 1 & \text{if} \quad |h'(x^*) + \hat{f}'(x^*)| \leq K \\ \frac{K - h'(x^*)}{\hat{f}'(x^*)} & \text{if} \quad h'(x^*) + \hat{f}'(x^*) > K \\ \frac{-K - h'(x^*)}{\hat{f}'(x^*)} & \text{if} \quad h'(x^*) + \hat{f}'(x^*) < -K \end{cases} \tag{5}$$

Given an initial structure h_0, and a sequence of actions characterized by the position and direction parameters $(\phi_1, \theta_1), (\phi_2, \theta_2), (\phi_3, \theta_3) \ldots$, the final structure after n actions is:

$$M[f_{\phi_n,\theta_n}, M[f_{\phi_{n-1},\theta_{n-1}}, \ldots M[f_{\phi_1,\theta_1}, h_0]] = h_0(x) + \sum_{i=1}^{n} \alpha_i \hat{f}_{\phi_i,\theta_i}(x) \tag{6}$$

3 Planning Algorithm

Based on Eq. 6, we can reformulate the construction problem as follows. Given an initial structure h_0 and a goal structure h_1 which must have the same integral over Q, how can we find a sequence of actions $\{f_{\phi_i,\theta_i}\}$ that can modify h_0 into a structure

that is as close to h_1 as possible? In this section, we present a planning algorithm that can automatically abstract the construction problem into multiple subtasks which are much easier to solve based on Wasserstein geodesics. In each subtask, the algorithm generates an intermediate goal structure h_t based on the current structure, and then uses a greedy approach to find an action sequence that can modify the current structure to h_t. The remainder of this section reviews optimal transport theory and Wasserstein distance and then presents the basic framework of the planning algorithm.

3.1 Optimal Transport and Wasserstein Metric

Optimal transport theory was originally formulated by Monge to study the most efficient way to transform a distribution of mass to another. [17] In recent years it has been widely used in statistics, machine learning, and image processing. [18–20] Here we briefly introduce optimal transport theory and the Wasserstein metric without going into details of measure theory and Riemannian geometry. Consider two probability density functions I_0 and I_1 defined over $\Omega \subset \mathbb{R}^n$. A transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ tells 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:

$$\int_{T^{-1}(A)} I_0(x)\,dx = \int_A I_1(x)\,dx. \tag{7}$$

If T is differentiable and bijective, by applying change of variables, we have:

$$\det(DT^{-1}(x))I_0(T^{-1}(x)) = I_1(x), \tag{8}$$

where $\det(DT^{-1}(x))$ is the Jacobian of T^{-1}. 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 optimal transport problem based on Monge formulation is:

$$M(I_0, I_1) = \inf_{T \in MP} \int_{\mathbb{R}^n} d(x, T(x))I_0(x)dx. \tag{9}$$

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

$$W_2(I_0, I_1) = \left(\inf_{T \in MP} \int_{\mathbb{R}^n} |x - T(x)|^2 I_0(x)dx \right)^{1/2}. \tag{10}$$

Let $P(\Omega)$ be the set of probability densities supported on Ω, then the metric space $(P(\Omega), W_2)$ 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:

$$I_t(x) = \det(DT_t^{-1}(x))I_0(T_t^{-1}(x)), \quad T_t(x) = (1 - t)x + tT^*(x). \tag{11}$$

Eq. 10 also suggests that the 2-Wasserstein geodesics has the following properties:

$$W_2(I_0, I_t) = tW_2(I_0, I_1). \tag{12}$$

In the case of 1D, the optimal transport map, 2-Wasserstein distance, and 2-Wasserstein geodesics all have closed-form solutions.

$$T^*(x) = F_1^{-1} \circ F_0(x), \tag{13}$$

$$W_2(I_0, I_1) = \left(\int_0^1 \left| F_0^{-1}(t) - F_1^{-1}(t) \right|^2 dt \right)^{\frac{1}{2}}, \tag{14}$$

$$I_t(x) = \frac{I_0(T_t^{-1}(x))}{T_t'(T_t^{-1}(x))}, \quad T_t(x) = (1-t)x + tT^*(x), \tag{15}$$

where F_0 and F_1 are cumulative distribution functions of I_0 and I_1. In future work where we need to study 3D structures, T^* in higher dimensions can be estimated using numerical methods such as flow minimization, gradient descent, linear programming, and entropy regularization. [21–24]

3.2 Measure of Distance and Intermediate Structures

We apply the Wasserstein metric in 1D to measure the distance between structures and generate intermediate structures. The height function h is converted to a probability distribution function I by dividing by the integral over Q. Consider two arbitrary height functions h_a and h_b with the same integral v over Q. Let $\mathcal{Q}$ denote the space of real-valued, bounded functions on Q and let $\mathcal{Q}^+$ denote the non-negative subset of $\mathcal{Q}$. Thus we have $\mathcal{H} \subset \mathcal{Q}^+ \subset \mathcal{Q}$. Eq. 14 can be used to measure the distance between them. Let $D_{W_2} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ be the distance function based on 2-Wasserstein distance. Let $D_{L_2} : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ be the distance function based on Euclidean distance.

$$D_{W_2}(h_a, h_b) = W_2(h_a/v, h_b/v), \tag{16}$$

$$D_{L_2}(h_a, h_b) = \left(\int_Q |h_a(x) - h_b(x)|^2 dx \right)^{\frac{1}{2}}. \tag{17}$$

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. 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. We construct an operator $G_t : \mathcal{Q}^+ \times \mathcal{Q}^+ \to \mathcal{Q}^+$ for generating an intermediate structure at point t between two given structures based on Eq. 15.

$$G_t[h_a, h_b](x) = \frac{h_a(T_t^{-1}(x))}{T_t'(T_t^{-1}(x))}, \quad T_t(x) = (1-t)x + tF_b^{-1} \circ F_a(x), \tag{18}$$

where F_a, F_b are cumulative distribution functions of h_a and h_b. Notice that T_t^{-1} always exists since all height functions are non-negative and continuous. The generated intermediate goal may not be a valid structure. However, it does not affect the planning since in each subtask the algorithm does not have to reach the exact intermediate goal.

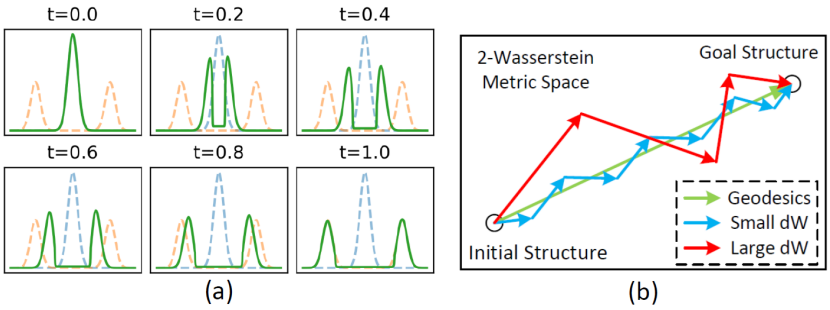

Fig. 2: (a) Wasserstein geodesics (green solid line) from h_0 (blue dotted line) to h_1 (red dotted line). (b) Effect of dW on how Alg. 1 follows the geodesics.

3.3 Planning Algorithm based on Wasserstein Geodesics

The basic planning algorithm is shown in Alg. 1. The algorithm decomposes the construction problem into subtasks, in which it computes the shortest path (geodesics) from the current to the goal structure and then attempts to move the structure by dW over that path (lines 3-5). If the distance to the goal is less than dW, the algorithm attempts to move towards the goal directly. The algorithm then uses a greedy approach to determine the action sequence that leads to the intermediate goal. Notice that since the prediction process does not capture the noise, the attenuation factor, and robot constraints, the execution result (line 13) might be very different from the predicted outcome. dW determines the resolution of decomposition. This effect is illustrated in Fig. 2. Smaller dW leads to a path closer to the geodesics and thereby less mass movement and fewer actions. Smaller dW also leads to simpler subtasks which require fewer actions to accomplish. Therefore the greedy approach can better predict the action sequence and the algorithm can more efficiently solve the problem. The trade-off is more computation and more frequent communication with the centralized controller.

4 Single Robot Execution

In this section, we present the simulation results for Alg. 1 executed by a single robot without constraints, to show that with an idealized robot, the algorithm can accomplish almost any construction task within the physical limits of the granular material. For comparison, we also tested a naive greedy algorithm (Alg. 2) which does not have subtask abstraction.

4.1 Random Structure Generation

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

Algorithm 1: Planning algorithm based on Wasserstein geodesics.

Input: construction area Q, initial height function h_0, goal height function h_1, desired distance to move in every subtask dW.

Set current structure $h \leftarrow h_0$
repeat
  Get distance to the goal (Eq: 16): $W \leftarrow D_{W_2}(h, h_1)$
  Set the point on the geodesics: $t \leftarrow \min\{1, dW/W\}$
  Get intermediate goal (Eq: 18): $h_t \leftarrow G_t[h, h_1]$
  $A \leftarrow 0, S \leftarrow \emptyset$
  repeat
    Find optimal action: $\phi^*, \theta^* \leftarrow \text{argmin}_{\phi,\theta} D_{L_2}(h + A + f_{\phi,\theta}, h_t)$
    Get distance (Eq: 17) to the intermediate goal: $L \leftarrow D_{L_2}(h + A + f_{\phi^*,\theta^*}, h_t)$
    $A \leftarrow A + f_{\phi^*,\theta^*}$
    Add optimal action (ϕ^*, θ^*) to S
  until L *stops decreasing*;
  Execute the action sequence S (Eq. 6) and update h.
until W *stops decreasing*;

Algorithm 2: Naive greedy algorithm.

Input: construction area Q, initial height function h_0, goal height function h_1.

Set current structure $h \leftarrow h_0$
repeat
  Get distance to the goal (Eq: 17): $L \leftarrow D_{L_2}(h, h_1)$
  Find optimal action: $\phi^*, \theta^* \leftarrow \text{argmin}_{\phi,\theta} D_{L_2}(h + f_{\phi,\theta}, h_1)$
  Execute the action based on the discrete dynamics (Eq. 6) and update h.
until L *stops decreasing*;

structures defined on Q. The random height function is given by:

$$h(x) = \sum_{i=1}^{m} a_i e^{-\frac{(x-b_i)^2}{2c_i^2}}, \tag{19}$$

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 on a fixed Q that is $5\,\text{m}$ (13 times the robot length) wide. 20 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.

4.2 Simulation Results

The simulation results are shown in Fig. 4. The average progress achieved by Alg. 1 and Alg. 2 are $94 \pm 3\%$ and $8 \pm 12\%$ respectively. This shows that greedily following locally optimal actions often cannot solve the full construction problem. The average number of steps to complete 95% of a structure is 2000 ± 1185 with each step taking 1 second. Since the construction area is only $5\,\text{m}$ wide, the large number of actions

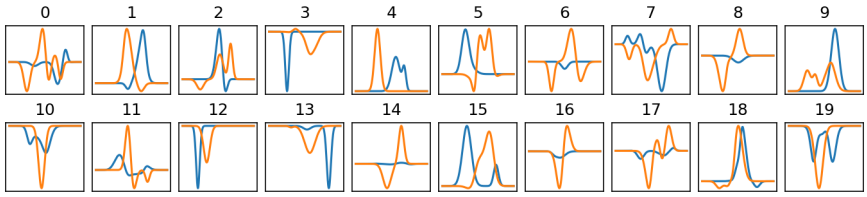

Fig. 3: Example of 20 tasks from the 100 randomly generated construction tasks. Each task comprises an initial structure h_0 (blue line) and a goal structure h_1 (orange line). Notice that h_0 and h_1 have the same volume of material.

shows the long-horizon nature of the construction problem, which again strengthens the importance of subtask abstraction. It also suggests that other methods with poor sampling efficiency like reinforcement learning may be inefficient.

5 Towards Real Deployment

In the following sections, we start taking the constraints of real robots and the cost of navigation into account. For a real robot, there is a maximum steepness $K_R = 0.3$ it can climb before becoming stuck [9] (it can still descend a slope with steepness $< -K_R$). After imposing this constraint, the robot will not travel to any location outside the reachable area. Given the x-coordinate of robot location x_R, the reachable area $Q_R \subset Q$ is the largest closed interval $[a, b]$ that includes x_R and satisfies following conditions:

$$\begin{aligned} &h'(x) \leq K_R \quad \text{for} \quad b \geq x > x_R, \\ &h'(x) \geq -K_R \quad \text{for} \quad a \leq x < x_R, \\ &\partial_+ h(x_R) \leq K_R \quad \text{or} \quad \partial_- h(x_R) \geq -K_R, \end{aligned} \tag{20}$$

where $\partial_+ h(x_R)$ and $\partial_- h(x_R)$ are right and left derivative at robot location. If the robot receives a command to go to a location outside Q_R, it will simply ignore it. Notice that after a robot travels to a location in Q_R, it may not be able to go back to its previous location. This constraint makes the construction process much more unpredictable as new actions can unintentionally change the accessible area for the robot.

The cost of traversal is the second effect we need to consider. To overcome finite battery capacity we need to minimize the traveling distance while accomplishing the construction task. Although following the Wasserstein geodesics minimizes the mass moving distance, it is unrelated to the traveling distance of the agent who executes it. A good example is the construction task shown in Fig. 2(a), where we want to split one mound in the middle into two mounds on each side. If we use a single robot to follow the geodesics, it will travel back and forth between two mounds, resulting in wasted energy. This problem can be addressed by introducing multiple (simple and therefore inexpensive) robots to execute the algorithm presented in Sec. 6. Another reason why we need to minimize the traveling distance is robot motion will change the structure in

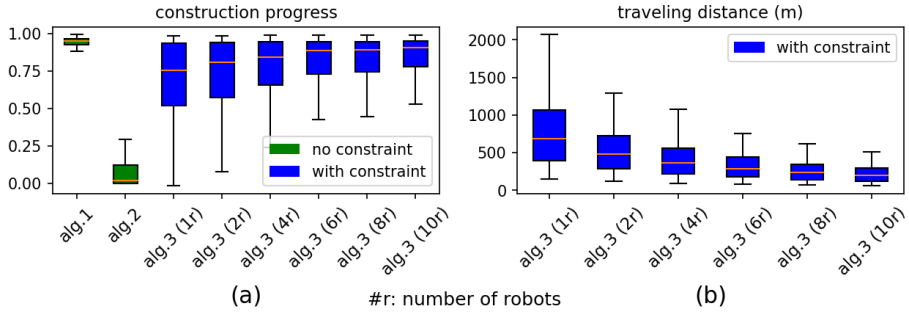

Fig. 4: Simulation results of solving 100 random construction tasks. Each task is simulated 5 times. (a) Construction progress. (b) Total traveling distance of all robots.

reality. The degree of change depends on many factors including the robot's speed, the structure's shape, and the granules' material properties. For example, the change would be subtle if the robot travels on a flat surface at low speed. Modeling this change is left for future work. Here, we focus on minimizing the robot's traveling distance.

6 Multi-Robot Execution

In this section, we show that by adding more robots, the system can overcome the robot's motion constraints to achieve higher construction progress and reduce the traveling distance. We propose the multi-robot planning algorithm based on Wasserstein geodesics (Alg. 3). The proposed algorithm comprises a flattening phase (lines 2-8), and then a construction phase (lines 10-27). Given robot position x_R, let $Q_T \subset Q$ denote the traversable area of a robot over which it can freely move back and forth. Q_T is the largest closed interval that includes x_R and satisfies: $\forall x \in Q_T, |h'(x)| \leq K_R$. We also define the untraversable area $Q_U \subset Q$ as any area that satisfies: $\forall x \in Q_U, |h'(x)| > K_R$. In the flattening phase, each robot randomly moves around its traversable area. If the local steepness exceeds threshold K_{TH}, the robot takes an action to flatten the structure and then moves to the next location. By repeating this process, each robot can potentially expand its traversable area, ultimately making the entire structure traversable. In the construction phase, the algorithm applies Alg. 1 to find the action sequence to accomplish each subtask. For each action, the algorithm assigns the closest agent who can reach the action location to execute it or skips this action if such a robot does not exist. With multiple robots on site, this strategy increases the chance of completing the structure under constraints and lowers the traveling distance.

6.1 Simulation Results

The simulation results are shown in Fig. 4. We first compare the single-robot execution with and without the motion constraint. Fig. 4(a) shows that after adding the constraint, the average progress drops to $69 \pm 27\%$. Notice that for all tasks, the maximum steepness

Algorithm 3: Multi-Robot planning algorithm based on Wasserstein geodesics. A robot swarm with N robots is identified by $\{1, \ldots, N\}$. The x-coordinates of their positions are denoted by $X = [x_1, \ldots, x_N]$. The reachable area and traversable area of robot i are denoted by $Q_{R,i}$ and $Q_{T,i}$ respectively.

Input: construction area Q, initial height function h_0, goal height function h_1, desired distance to move in every subtask dW, number of robots N. Steepness threshold for taking an action to flatten the structure K_{TH}.

Set current structure $h \leftarrow h_0$, and deploy robots uniformly over untraversable area Q_U
repeat
  for *each robot* i **do**
    Choose a random location $\tilde{\phi} \in Q_{T,i}$ that satisfies: $|h'(\tilde{\phi})| > K_{TH}$
    if $h'(\tilde{\phi}) > K_{TH}$ **then** $\tilde{\theta} \leftarrow$ left **else** $\tilde{\theta} \leftarrow$ right;
    Execute action $\tilde{\phi}, \tilde{\theta}$ (Eq. 6) and update h and X
  end for
until $\forall i, Q_{T,i} = Q$;
Robots spread out to uniformly cover the entire structure
repeat
  Get distance to the goal (Eq: 16): $W \leftarrow D_{W_2}(h, h_1)$
  Set the point on the geodesics: $t \leftarrow \min\{1, dW/W\}$
  Get intermediate goal (Eq: 18): $h_t \leftarrow G_t[h, h_1]$
  $A \leftarrow 0, S \leftarrow \emptyset$
  repeat
    Find optimal action: $\phi^*, \theta^* \leftarrow \text{argmin}_{\phi,\theta}\, D_{L_2}(h + A + f_{\phi,\theta}, h_t)$
    Get distance (Eq: 17) to the intermediate goal: $L \leftarrow D_{L_2}(h + A + f_{\phi^*,\theta^*}, h_t)$
    $A \leftarrow A + f_{\phi^*,\theta^*}$
    Add optimal action (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**
      Choose execution robot i s.t. $\phi_j \in Q_{R,i}$ and $|x_i - \phi_j|$ is minimized
      Execute the action (Eq. 6) and update h and X
    end if
  end for
until W *stops decreasing*;

of the initial or goal structure is close to the structure constraint K, which exceeds the maximum steepness the robot can climb K_R. We then compare single-robot execution to multi-robot execution with the motion constraint. Fig. 4(a) shows that as more robots are added, Alg. 3 achieves higher progress. The highest average progress is $85 \pm 13\%$ for ten robots, a significant increase from the single-robot case. To study the effect of the number of robots on the traveling distance, we execute Alg. 3 with motion constraint and analyze 25 tasks with construction progress $> 90\%$. Fig. 4(b) shows as more robots are added, the total traveling distance of all robots decreases. The average traveling distance drops from $795 \pm 540\text{m}$ for one robot to $237 \pm 156\text{m}$ for ten robots, a 70% decrease. We also found that the initial traveling distance for flattening the structure

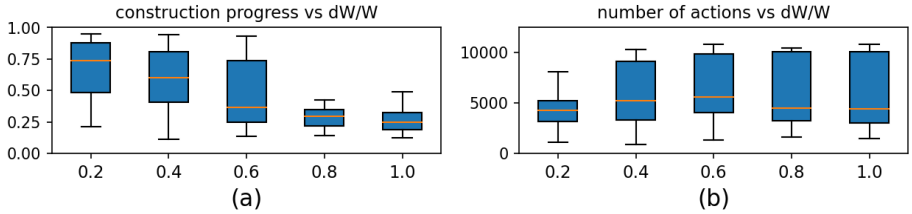

Fig. 5: Simulation results of solving task #98. Each setting of dW is simulated 100 times. W is the distance between the initial structure and goal structure. (a) Construction progress vs dW. (b) Number of actions vs dW.

is only up to 8% of the total traveling distance, which is almost negligible. Under the influence of the motion constraint, the effect of dW is amplified as the algorithm has a much higher chance of solving simpler subtasks. We study this effect by solving task #98 which has the lowest progress even being built by multiple robots. The distance between the initial and goal structure of task #98 is $W = 15.39$. We solve this task with 10 robots and $dW = 0.2W \sim W$. When $dW = W$, the algorithm always attempts to solve the task in one shot. Fig. 5 shows as dW increases, the progress drops, and more actions are taken. This illustrates the importance of subtask abstraction when solving the long-horizon construction problem.

7 Discussion

In this paper, we developed an abstract model for modifying 2D granular structures based on data from a simple robot presented in [9]. Based on this model, we proposed a planning algorithm that utilizes Wasserstein geodesics to solve the construction problem of modifying a given structure to a user-defined new structure. A robot can follow this plan with a simple greedy approach. To reduce the robot's traveling distance and address the robot's motion constraint, we extended this work to a multi-robot algorithm. This strategy allows the system to build structures that are infeasible for a single robot.

There are many avenues to further our work. For the model, the effect of robot motion on the structure can be included. For algorithms, more advanced methods like reinforcement learning may be used to solve subtasks. Many subtasks are similar and the learned policy for one subtask may be transferable to solve other subtasks. Agents could also be more intelligent and cooperate to overcome the motion constraint. Finally, we are interested in extension to 3D and decentralized operation. For 3D construction, the main challenges are finding suitable construction functions and computing the optimal transport map and Wasserstein geodesics. To decentralize the system, one potential solution is to let each agent compute Wasserstein geodesics locally.

Acknowledgments

This project was funded by the Packard Fellowship for Science and Engineering and the National Science Foundation (NSF) Grant #2042411.

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] R. Lee, B. Younes, A. Pletta, J. Harrington, R. Q. Wong, W. Whittaker *et al.*, "Cratergrader: Autonomous robotic terrain manipulation for lunar site preparation and earthmoving," *arXiv preprint arXiv:2311.01697*, 2023.

[4] S. Hong, P. Shyam, A. Bangunharcana, and H. Shin, "Robotic mapping approach under illumination-variant environments at planetary construction sites," *Remote Sensing*, vol. 14, no. 4, p. 1027, 2022.

[5] J. Thangavelautham and Y. Xu, "The design of autonomous robotic technologies for lunar launch and landing pad (llp) preparation," in *2022 IEEE aerospace conference (AERO)*. IEEE, 2022, pp. 1–13.

[6] S. Hong, A. Bangunharcana, J.-M. Park, M. Choi, and H.-S. Shin, "Visual slam-based robotic mapping method for planetary construction," *Sensors*, vol. 21, no. 22, p. 7715, 2021.

[7] M. Trojnacki, P. Brzeczkowski, and D. Kleszczyński, "Experimental research of veles planetary rover performing simple construction tasks," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 15, no. 4, pp. 30–36, 2021.

[8] A. R. Cheraghi, S. Shahzad, and K. Graffi, "Past, present, and future of swarm robotics," in *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3*. Springer, 2022, pp. 190–233.

[9] 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.

[10] D. Shachaf, O. Inbar, and D. Zarrouk, "Rsaw, a highly reconfigurable wave robot: Analysis, design, and experiments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4475–4482, 2019.

[11] 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.

[12] K. H. Petersen, N. Napp, R. Stuart-Smith, D. Rus, and M. Kovac, "A review of collective robotic construction," *Science Robotics*, vol. 4, no. 28, p. eaau8479, 2019.

[13] N. Napp and R. Nagpal, "Distributed amorphous ramp construction in unstructured environments," *Robotica*, vol. 32, no. 2, pp. 279–290, 2014.

[14] M. Saboia, V. Thangavelu, and N. Napp, "Autonomous multi-material construction with a heterogeneous robot team," *Robotics and Autonomous Systems*, vol. 121, p. 103239, 2019.

[15] V. Thangavelu, M. S. da Silva, J. Choi, and N. Napp, "Autonomous modification of unstructured environments with found material," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7798–7804.
[16] J. Ondras, D. Ni, X. Deng, Z. Gu, H. Zheng, and T. Bhattacharjee, "Robotic dough shaping," in *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2022, pp. 300–307.
[17] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.
[18] 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.
[19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.
[20] 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.
[21] 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.
[22] 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.
[23] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013.
[24] A. M. Oberman and Y. Ruan, "An efficient linear programming method for optimal transportation," *arXiv preprint arXiv:1509.03668*, 2015.