Coordinated Construction by a Distributed Multi-Robot Sys. (Ph.D. of Seung-hook Yun, MIT)

- Groups of robots that complete a complex assembly task using the maximal amount of parallelism, in a way that can adapt to the amount of construction material.

- 2 types of robots: part delivery robots that locate & deliver parts, and assembly robots that join the parts into desired objects.

- The target structure is given by a blueprint, a material-density function that encodes the object geometry and is known to all robots.

- The assembly robots partition the structure adaptively into subassemblies, and each robot is responsible for completion of a partition region.
  - Robots locally compute a Voronoi partition, weighted by the mass of all parts in the partition, and perform a gradient descent algorithm to balance the masses of the regions.

- The delivery robots locate parts in a cache and bring them to the assembly robots.
  - Want the assembly robots to work at approx. the same pace.
  - Delivery occurs according to the demand of mass for each subassembly, the amount of remaining work (measured in the # of components that have to be added).
Proposed solution using Voronoi cells:

- Each robot travels the entire blueprint (slow)
- Requires knowledge of exact construction plan & placement of each part before execution
- Each assembly robot builds within its own Voronoi cell.

- Decentralized controllers are proposed for (1) partitioning, (2) part delivery, and (3) assembly.

- Construction involves discrete components. 
  - ex) Truss structure: 
    - Can be modeled as a graph

Problem Formulation

\[ N_a = \# \text{ of } D \quad N_e = \# \text{ of } A \]

- Robots can communicate locally with other robots within their communication range.
- The domain is $Q \subset \mathbb{R}^N$ (N=2 or 3) or a graph $G = (V, E)$.

(a) **Domain is $Q \subset \mathbb{R}^N$:***

- Robots are given a target density fn. $\phi_t : Q \rightarrow \mathbb{R}$
  
  Density of construction material

- If components can be built independently + an assembling robot is capable of assembling all of them, then:

  $$\phi_t = \sum_{u=1}^{z} \beta_u \psi_u$$

  ($\beta_u$ weights the importance of $u$th component, the piece, time until piece is needed in the assembly, etc.)

- To represent truss structures, $\phi_t$ is defined point-wise on the grid that corresponds to the truss. Point density $\propto \#$ of possible truss connections at the point.

(b) **Domain is $G = (V, E)$:**

- $p_i \in V$ is the position node of robot $i$

- $d(\cdot, \cdot) : E \rightarrow \mathbb{R}^+ =$ shortest distance measure between 2 vertices [$d(s, t) = \infty$ when $(s, t) \notin E$]

- $\phi_t(v) =$ vertex weight denoting the importance of a task at vertex $v$ (target density fn.)

- Divide $G$ into graph Voronoi partitions:

  $$V_i = \{ v \in V | d(v, p_i) < d(v, p_j), \forall j \neq i \}$$
The nearest robot to \( v \) will execute the task at \( v \). Each robot is allocated the task that includes its Voronoi partition \( V_i \) in \( G \).

- Need to clarify assignment of a vertex with same distance to multiple robots; give priority to robot with the minimum ID: 
  \[ v \in V_i \Rightarrow i = \min \{ j \mid d(v, p_i) = d(v, p_j) \} \]
- Let \( w_i \) be a weight; larger \( w_i \) ⇒ larger region

Generalized Voronoi partition:

\[ V_i = \{ v \in V \mid (d(v, p_i) - w_i) < (d(v, p_j) - w_j), \forall j \neq i \} \]

Assumptions:
1. \( G \), \( \phi_t(v) \) is given to each robot
2. \( \phi_t(v) \) is fixed
3. robots do not know locations of other robots \( \Rightarrow \) can't precompute the optimal config.
4. robots precompute the distance matrix \( D \) of \( G \) as a \( |V| \times |V| \) symmetric matrix where

\[ D_{ij} = d(v_i, v_j) \].

Construction Algorithm:

1. Deploy \( A \) in \( Q \) or \( V \)
2. Place \( A \) at optimal task locations in \( Q \) or \( V \) (uses a distributed controller)

Repeat until task completed or out of parts:
3. carry source parts to \( A \) with max. demand, \( w_i \) mass
4. assemble delivered parts after determining the optimal placement of the part in the assembly
Equal-mass partitioning

\[ \phi_t : Q \rightarrow R \]

\( p_i = \) position of \( A \)

Mass of robot \( i \) = size of total shaded region in its Voronoi region

\( \rightarrow_v \) = direction of motion of robot;
component of normal to edge of Voronoi region
(combine to get resultant direction of robot's motion)

\[ \Delta M_{V_i}^+ = 4 \quad \Delta M_{V_2}^- = 0 \]

\[ \Delta M_{V_3}^+ = 2 \quad \Delta M_{V_4}^+ = 1 \]

\[ \blacksquare = \) completed assembly
\[ \Delta M_{V_i}^- = \) demanding mass of region \( i \)
\[ = \) area of \( \blacksquare \) in region \( i \)
\[ - \) area of \( \) in region \( i \)

\( \bullet \) is in \( V_4 \): among the neighboring regions \( (V_2, V_3) \),
\( V_3 \) has a higher demanding mass

\[ \Rightarrow \bullet \) moves to \( V_3 \).

\( \bullet \) is in \( V_3 \): among the neigh. regions \( (V_1, V_4) \),
\( V_1 \) has a higher demanding mass

\[ \Rightarrow \bullet \) moves to \( V_1 \).
Decentralized equal-mass partitioning controller

- Given \( q \in Q \), the nearest robot to \( q \) will execute the assembly task at \( q \).
- Each robot is allocated the assembly task in its Voronoi partition:
  \[
  V_i = \{ q \in Q | \| q - p_i \| \leq \| q - p_j \|, \forall j \neq i \} \]
  \( \phi_t = \) density of parts
  \[
  M_{Vi} = \int_{V_i} \phi_t(q) \, dq
  \]

  \[
  \text{Cost function: } H_0 = \sum_{i=1}^{n} \int_{V_i} \frac{1}{2} \| q - p_i \|^2 \phi(q) \, dq
  \]
  [for max sensor coverage]

- Want each robot to have the same amount of assembly work (same # of truss elements)
  \[
  H_0 = \left( \frac{1}{n} \sum_{i=1}^{n} M_{Vi} \right)^n \quad H = H_0 - \sum_{i=1}^{n} M_{Vi}
  \]

- \( H \) is continuously differentiable
- Minimizing \( H \) leads to equal-mass partitioning:
  \[
  \left( \text{arithmetic mean} \right) \quad \frac{1}{n} \sum_{i=1}^{n} M_{Vi} \geq \left( \prod_{i=1}^{n} M_{Vi} \right)^{\frac{1}{n}} \quad \left( \text{geometric mean} \right)
  \]
  \( \geq \) is only if all \( M_{Vi} \) are the same (then \( H = 0 \))

- Can guarantee that \( H \) converges to a local minimum under
  the controller
The robot controller continuously decreases the cost $\mathcal{H}$:

$\mathcal{H} \leq 0, \ t > 0$. 

$\dot{\mathcal{H}} = \sum_{i=1}^{n} \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} \dot{\mathbf{p}}_i \quad \mathbf{p}_i = \text{position of robot } i$

$\mathcal{N}_i = \text{set of neighbors of robot } i$

Can derive $\dot{\mathcal{H}}$ as:

$\dot{\mathcal{H}} = -\sum_{i=1}^{n} \prod_{l \not\in \mathcal{E}_i, \mathcal{N}_i} M_{vl} \sum_{j=1, \mathcal{N}_i} \frac{\partial M_{vj}}{\partial \mathbf{p}_i} \prod_{k \in \mathcal{E}_i, \mathcal{N}_i, k \neq j} M_{vk} \dot{\mathbf{p}}_i$

- Decentralized controller:

$\dot{\mathbf{p}}_i = k \frac{\mathbf{J}_i}{\|\mathbf{J}_i\|^2 + \lambda^2}$

$k = \text{positive control gain}$

$\lambda = \text{constant to stabilize controller even around singularities where } \|\mathbf{J}_i\|^2 = 0$.

- Only depends on variables of neighboring robots.

The controller guarantees that $\dot{\mathcal{H}}$ converges to either a local maximum or a global maximum:

$\dot{\mathcal{H}} = -k \sum_{i=1}^{n} \frac{\|\mathbf{J}_i\|^2}{\|\mathbf{J}_i\|^2 + \lambda^2} \prod_{l \not\in \mathcal{E}_i, \mathcal{N}_i} M_{vl}$

$k > 0, \ M_{vl} > 0 \Rightarrow \text{each term of } \dot{\mathcal{H}} \text{ is negative}$
Also, $H$ is differentiable, and robot trajectories are bounded in $Q$.

Controller keeps $H$ decreasing until all $J_i = 0$ (relocating the robots does not change $H$).

**Equal-mass partitioning with locational optimization**

Although the controller above leads robots to regions with equal masses, the region shapes may not be desirable in terms of robot travel time and communication range.

- **Add locational optimization property to the controller.**
  - A solution of loc. optim. is to locate robots at the centroids:
    
    $$CV_i = \frac{1}{Mv_i} \int_{V_i} q \phi_t(q) dq$$
    
    - Redefine $H_0$ as:
      $$H_0 = \sum_{i=1}^{n} Mv_i \| CV_i - \hat{p}_i \|^2$$
    - New cost function:
      $$\hat{H} = H + \gamma H_0 \quad (\gamma > 0 \text{ can be tuned})$$
    - Can redefine $\hat{p}_i$ to produce same convergence prop. for $\hat{H}$.

**Locational optimization:** Optimization problems in operations research for placing facilities to minimize costs in terms of distance (transportation costs).