Aggregation and Coverage with Many Robots and Uniform Inputs

Aaron Becker

***Abstract*— Medical interventions with large numbers of micro-scale robots has two primary control challenges: (1) aggregate the robots at a desired location and (2) ensure robots have covered the region of interest. Both challenges might be trivial with perfect feedback and independent actuation for each robot, however current technology provides low time-resolution feedback and robots are typically actuated by a shared, global field. To make progress, this paper analyzes each task in a 2D grid world, where all obstacles and robots are unit squares, and for each actuation, robots move maximally until they collide with an obstacle or another robot**

I. Introduction

This paper builds on the techniques for controlling many simple robots with uniform control inputs presented in [1]–[3], and outlines new research problems.

A. Aggregation

For many therapeutic treatments it is important to concentrate a drug at a particular site. The old adage *toxicity is a function of concentration* explains that often we can flow a diluted drug through the body without ill-effect, and then kill cells at a targeted location by collecting drug particles. Targeted drug therapy is a goal for treating cancers, delivering pain-killers, and stopping internal bleeding.

B. Coverage

For biomedical imaging, we want to ensure our sensors pass over the entire scan region. Similarly, for a drug treatment, we want to ensure our drug is released everywhere in our treatment region. These are questions of *coverage*.

Traditional systemic drug delivery is wasteful because a uniform dosage is applied throughout the body, often with deleterious side effects on healthy tissue. Case study: metrologists (not weather - nanoscale measuring) can measuring blood vessels in real time (Think 4D MRT - instead of just 3D.) For this it is important to get magnetic particles everywhere - and in a controlled fashion.

C. Problem Definition

We consider the following scenario, which we call GlobalControl-ManyRobots:

1) Initially, the planar square grid is filled with some unit-square robots (each occupying one cell of the grid) and some fixed unit-square blocks.
2) All robots are commanded in unison: the valid commands are "Go Up" (u), "Go Right" (r), "Go Down" (d), or "Go Left" (l). The robots all move in the commanded direction until they hit an obstacle or another robot. A representative command sequence is $\langle u, r, d, l, d, r, u, \ldots \rangle$. We call these global commands *force-field moves*. We assume we know the maximum dimension of the workspace and issue each command long enough for the robots to reach their maximum extent.
3) The goal is to get each robot to its specified position.

The algorithmic decision problem GlobalControl-ManyRobots is to decide whether a given configuration is solvable. This problem is computationally difficult: we prove PSPACE-completeness in [2]. While this result shows the richness of our model (despite the limited control over the individual parts), it also constitutes a major impediment for constructive algorithmic work.

This makes developing algorithmic tools that enable global control by uniform commands important. In Sections II and III, we develop several positive results. The underlying idea is to construct artificial obstacles (such as walls) that allow arbitrary rearrangements of a given two-dimensional robot swarm.

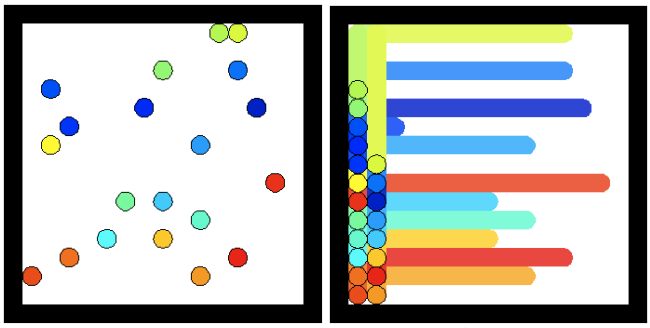

Fig. 1. In an obstacle-free workspace, aggregating particles to a corner can be accomplished in just two moves. The movement sequence $\langle l, d \rangle$ is robust to all initial particle configurations. In a workspace with obstacles, this sequence is insufficient.

II. Aggregation

In *aggregation*, all robots are gathered to the same area. In our grid world, we consider a collection of robots successfully aggregated if they form a single 4-neighbor connected component. For *specific aggregation*, the aggregation must contain one or more desired goal cells. The number of goal cells must be no more than the number of robots n.

Aggregating particles can be as simple as collecting all particles into a single connected component in one corner of the freespace, as shown in Fig. 1. In an obstacle-free rectangular workspace, moving particles from an arbitrary starting configuration to a corner requires only two moves.

A. Becker is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA { atbecker}@uh.edu

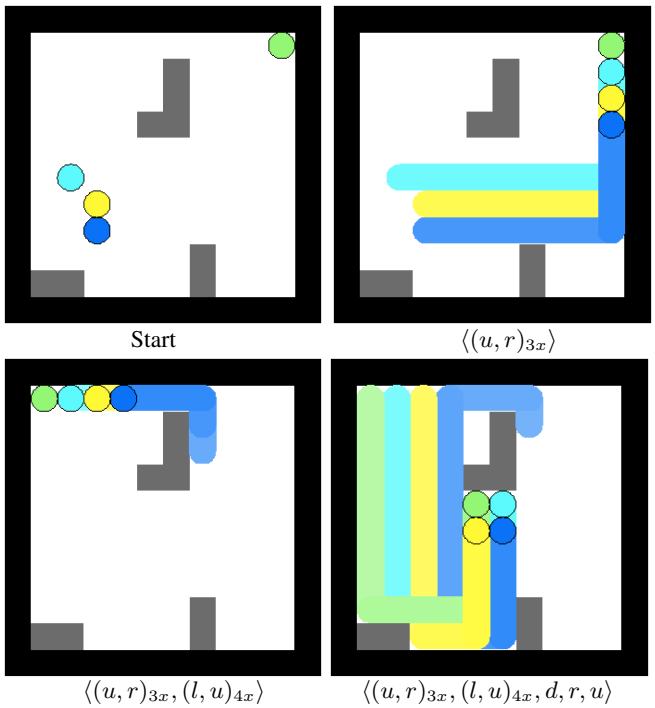

Fig. 2. Aggregating particles to a location other than a corner requires obstacles and additional moves. The sequence $\langle u, r, u, r, u, r, l, u, l, u, l, u, l, u, d, r, u \rangle$, and the eight obstacles (grey boxes) shown will move any starting configuration of four robots to end in a square shape at the center of the free space. This arrangement was hand designed – how can we automatically find these obstacle arrangements? How can we design solutions optimal in terms of minimum movements and/or obstacles?

A more challenging problem is shown in Fig. 2. Here the particles must be end in a square arrangement in the center of the freespace. One solution is shown, and requires many obstacles and a long movement sequence. This solution was hand designed—how can we automatically find these obstacle arrangements? How can we design solutions optimal in terms of minimum movements and/or obstacles? Brute-force solutions are challenging, since for an $n \times n$ workspace with k obstacles and m particles, there are $\binom{n^2-k}{m}$ starting configurations to check. How can this search be simplified?

III. Coverage

A region is *covered* if a robot passes within a fixed distance of every point in the free space. In our our grid world, the region is covered if at least one robot visits every free cell. A more challenging version of coverage, often found in the literature for painting robots, is *even coverage* that minimizes overlaps [4].

check terminology

In our case, ideally every cell is visited the same number of times.

A necessary condition is that every connected component of the area to be covered must contain at least one robot.

We consider a representative problem: Given a rectangular $n \times n$ workspace and m robots, arranged initially in a horizontal line in an input port at the top right of the freespace, what is the fewest moves required to cover a region?

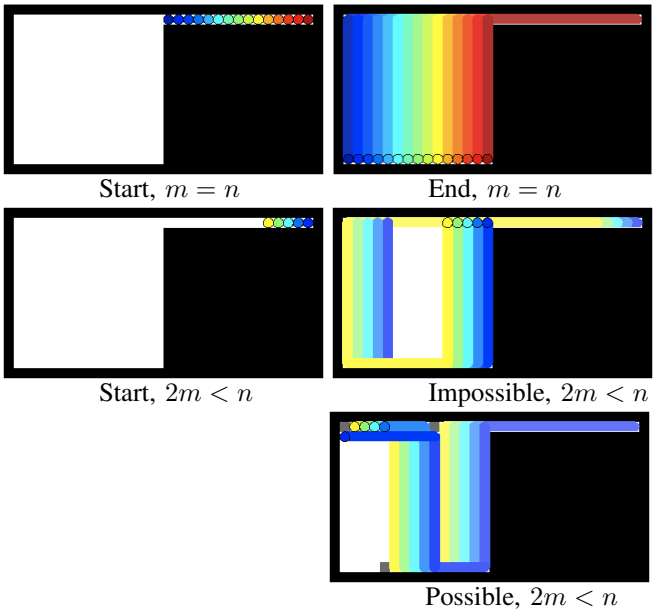

Fig. 3. Coverage can be trivial. If there are sufficient particles to span the region, and the particles are aligned correctly, coverage can require only one move. Here, two moves are used $\langle l, d \rangle$. With too few particles ($2m < n$) complete coverage requires obstacles.

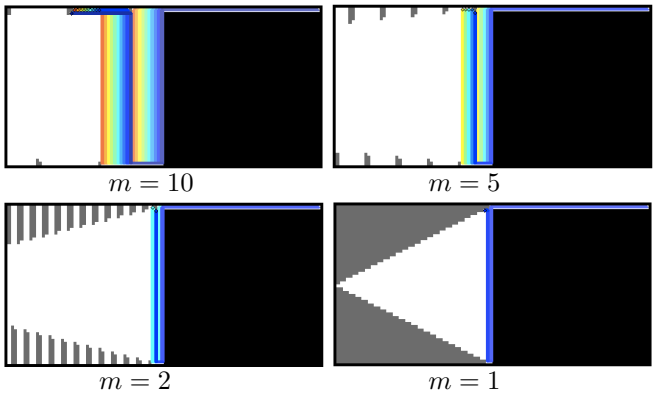

Fig. 4. Boustrophedon coverage $\langle l, d, l, u, \ldots \rangle$ is simple to implement, but requires $\frac{\lceil n/m \rceil \lceil n/m-1 \rceil}{2}$ obstacles. In the worst case when $m = 1$, almost half the workspace is obstacles.

This problem can be trivial if given sufficient robots to span the workspace. Then it requires only a single pass, as shown in Fig. 3. With $2m < n$ robots, obstacles are needed. One possible coverage method uses obstacles placed to enable *boustrophedon* coverage, following parallel paths in alternating directions. This method, shown in Fig. 4, is efficient in the number of moves $2\lceil \frac{n}{m} \rceil$, but requires $O\left(\lceil \frac{n}{m} \rceil^2\right)$ obstacles. This results in large numbers of obstacles when m is small. In the worst case when $m = 1$, almost half the workspace is obstacles.

A competing method passes the robots in a spiral pattern, see Fig. 5. The obstacles required to turn the particles forces the particles to skip one row or column on each pass, and to miss corners. These missed areas could be filled in with obstacles, but this solution requires $O\left(\lceil \frac{n}{m} \rceil^2\right)$ obstacles.

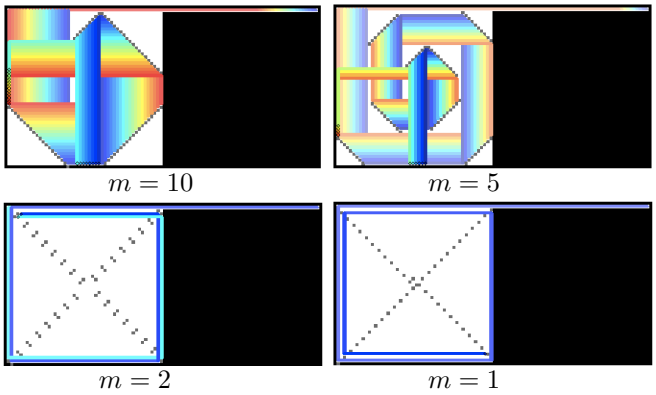

Fig. 5. Arranging obstacles for spiral coverage $\langle l, d, r, u, \ldots \rangle$ requires fewer obstacles, but produces incomplete coverage, leaving a empty path for every spiral path, and misses the corners.

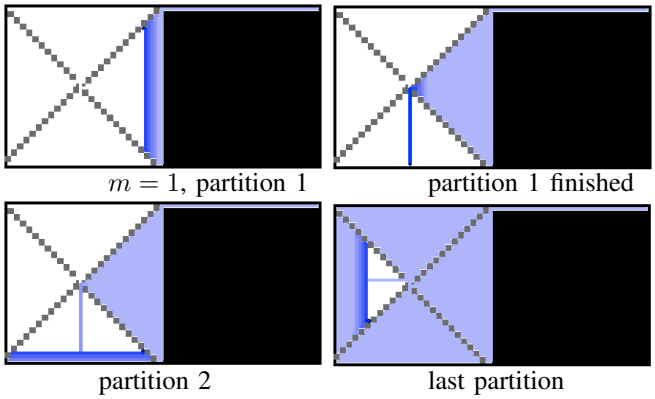

Fig. 6. For $m = 1$ robots, arranging obstacles in an 'X' pattern divides the workspace into four partitions. Each partition can be covered by a boustrophedon path, and the particle can switch between partitions at the center of the X. This movement requires about twice the moves of the boustrophedon path, but only $4n$ obstacles. Is this the most efficient way to cover a rectangular area with one robot?

It may be possible to combine the spiral and boustrophedon approaches to generate worlds that can be covered with far fewer obstacles. For $m = 1$ robots, arranging obstacles in an X pattern divides the workspace into four partitions, see Fig. 6. Each partition can be covered by a boustrophedon path, and the particle can switch between partitions at the center of the X. This movement requires ≈twice the moves of the boustrophedon path, but only $4n$ obstacles. Is this the most efficient way to cover a rectangular area with one robot?

IV. Future Work

As defined in this paper, *coverage* requires only that the robots pass over every square. For a time-sensitive operation such as drug delivery or measurements, it may be necessary to guarantee a robot remains in certain cells for minimum amount of time. This is possible if there exists a sequence of moves such that a robot can be stopped in these cells. Cell i is *reachable* if from any arbitrary starting configuration, there exists a sequence of moves such that the robot ends in i.

A 2×2 bounded free space is fully reachable with 1 robot. A 2×3 array is only partially reachable (66%) with 1 robot, and only fully reachable with 2 robots if the robots start arranged with the long axis of the free space.

Questions to consider:

- Is it possible to build a workspace containing N reachable cells with 1 moving robot?
- Is is possible to build a workspace with 100% reachability with 1 moving robot and N empty cells?
- how to design workspaces with maximum reachable cells?
- how do the above algorithms change by adding robots?

References

[1] A. Becker, E. Demaine, S. Fekete, G. Habibi, and J. McLurkin, "Reconfiguring massive particle swarms with limited, global control," in *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, Sep. 2013.

[2] A. Becker, E. Demaine, S. Fekete, and J. McLurkin, "Particle computation: Designing worlds to control robot swarms with only global signals," in *IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, May 2014, pp. 6751–6756.

[3] A. Becker, E. D. Demaine, S. P. Fekete, G. Habibi, and J. McLurkin, "Reconfiguring massive particle swarms with limited, global control," in *Algorithms for Sensor Systems*, ser. Lecture Notes in Computer Science, P. Flocchini, J. Gao, E. Kranakis, and F. Meyer auf der Heide, Eds. Springer Berlin Heidelberg, 2014, pp. 51–66. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-45346-5_5

[4] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, October 2001.