Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2009-027 June 17, 2009

Keeping Mobile Robots Connected

Alejandro Cornejo, Fabian Kuhn, Ruy Ley-Wild,
and Nancy Lynch

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Keeping Mobile Robot Swarms Connected
Technical Report

Alejandro Cornejo
acornejo@csail.mit.edu

Fabian Kuhn
fkuhn@csail.mit.edu

Ruy Ley-Wild
rleywild@cs.cmu.edu

Nancy Lynch
lynch@csail.mit.edu

Abstract

Designing robust algorithms for mobile agents with reliable communication is difficult due to the distributed nature of computation, in mobile ad hoc networks (MANETs) the matter is exacerbated by the need to ensure connectivity. Existing distributed algorithms provide coordination but typically assume connectivity is ensured by other means. We present a connectivity service that encapsulates an arbitrary motion planner and can refine any plan to preserve connectivity (the graph of agents remains connected) and ensure progress (the agents advance towards their goal). The service is realized by a distributed algorithm that is *modular* in that it makes no assumptions of the motion-planning mechanism except the ability for an agent to query its position and intended goal position, *local* in that it uses 1-hop broadcast to communicate with nearby agents but doesn't need any network routing infrastructure, and *oblivious* in that it does not depend on previous computations.

We prove the progress of the algorithm in one round is at least $\Omega(\min(d, r))$, where d is the minimum distance between an agent and its target and r is the communication radius. We characterize the worst case configuration and show that when $d \geq r$ this bound is tight and the algorithm is optimal, since no algorithm can guarantee greater progress. Finally, we show all agents get ε-close to their targets within $O(D_0/r + n^2/\varepsilon)$ rounds where n is the number of agents and D_0 is the sum of the initial distances to the targets.

1 Introduction

Motivation and Related Work Designing robust algorithms for mobile agents with reliable communication is difficult due to the distributed nature of computation. If the agents form a mobile ad hoc network (MANET) there is an additional tension because communication is necessary for motion-planning, but agent movement may destabilize the communication infrastructure. As connectivity is the core property of a communication graph that makes distributed computation possible, algorithms for MANETs must reconcile the interaction between communication and motion planning in order to preserve connectivity.

Existing distributed algorithms for MANETs provide coordination but typically sidestep the issue of connectivity by assuming it is ensured by other means. For example, algorithms on routing [7, 10], leader election [9], and mutual exclusion [14] for MANETs assume they run on top of a mobility layer that controls the trajectories of the agents. Those algorithms deal with connectivity by assuming the mobility layer guarantees that every pair of nodes that need to exchange a message are connected at some instant or transitively through time, otherwise they work on each independent connected cluster. On the other hand, work on flocking [11, 6], pattern formation [4], and leader following [2] provides a mobility layer for a MANET that determines how agents will move. Again connectivity is sidestepped by assuming coordination runs atop a network layer that ensures it is always possible to exchange information between every pair of agents. The service we present would thus enable to execute the flocking algorithm of [11] using the routing algorithm of [10], or running the leader follower algorithm of [2] using the leader election service of [9], with the formal guarantee that connectivity is maintained and progress is made. The connectivity service allows an algorithm designer to focus on the problems which are specific to the application (*i.e.*, search and rescue, demining fields, space exploration, etc.) without having to deal with the additional issues that arise when there is no fixed communication infrastructure. We expect that algorithms designed on top of this service will be easier to prove correct because the safety and progress properties are maintained orthogonally by the guarantees of the service.

Some algorithms developed in the control theory community are preoccupied with preserving connectivity, though they have limited applicability because they make restrictive assumptions about the goal and computation model. For example, a centralized method preserves connectivity by solving a constrained optimization problem [15] but doesn't exploit the locality of distributed computation, an algorithm for second-order agents [12] is centralized and conservatively preserves all edges, and another algorithm maintains connectivity but only works for agents converging to a single target position [1]. In this paper we focus on providing formal termination and progress guarantees, a preliminary version of the algorithm appeared in a previous paper [3] with no guarantees.

Communication Model We assume each agent is equipped with a communication device that permits reliable broadcasting to all other agents within some communication radius r. Without loss of generality we suppose $r = 1$ throughout. The service operates in synchronous rounds, it assumes access to a positioning device; relative position between neighboring agents is sufficient, but for ease of exposition we assume absolute position is available. Finally the service assumes the existence of a motion planner which is queried at each round for the desired target position, the service produces a trajectory which preserves connectedness and, when possible, gets closer to the target.

Contributions We present a distributed connectivity service that modifies an existing motion plan to ensure connectivity using only local information and without making any assumptions of the current and goal configurations. In particular, even if the goal configuration is disconnected, the service guarantees connectivity while trying to get each agent as close as possible to its target. Furthermore, the connectivity service only requires the immediate intended trajectory and the current position, but it is stateless, and hence *oblivious*. The service is also *robust* to the motion of each agent in that the refined plan preserves connectivity irrespective of the agents' speed changes. Therefore agents remain connected throughout their motion even if they only travel a fraction (possibly none) of their trajectory.

Connectivity is a global property, so determining whether an edge can be removed without disconnecting the graph may require traversing the whole graph. However, exploiting the distributed nature of a team of agents requires allowing each agent to perform tasks with a certain degree of independence, so communicating with every agent in the graph before performing each motion is prohibitive. To solve this we parametrize the service with a filtering method that determines which edges must be preserved and which can be removed, we also suggest several local algorithms which can be used to implement this filtering step.

We define *progress* as the quantification of how much closer each agent gets to its target in a single round. Our algorithm guarantees that the total progress is at least $\min(d, r)$ in configurations where every agent wants to move at least some distance d and the communication radius is r. Furthermore, we exhibit a class of configurations where no local algorithm can do better than this bound, hence under these conditions the bound is tight and the algorithm is asymptotically optimal. In the last section we prove all agents get ε-close to their target within $O(D_0/r + n^2/\varepsilon)$ rounds where D_0 is the total initial distance to the targets and n is the number of agents. Since the motion of the agents occurs in a geometric space and the service deals directly with motion planning, most progress arguments rely on geometrical reasoning.

We introduce some notation and definitions in §2. In §3 we present the intersecting disks connectivity service and discuss its parametrization in a filtering function. We prove the algorithm preserves connectivity and produces robust trajectories (§4). In §5 we prove that any lower-bound on progress for chains also applies for general graphs. We start §6 by giving a lower bound on progress of a very restricted class of chains with only two nodes, and in the rest of the section we show how to extend this lower bound to arbitrary chains. We give the termination bound in §7 and conclude in §8.

2 Preliminary Definitions

The *open disk* centered at p with radius r is the set of points at distance less than r from p: $\mathsf{disk}_r(p) := \{q : \|p - q\| < r\}$. The *circle* centered at p with radius r is the set of points at distance r from p: $\mathsf{circle}_r(p) := \{q : \|p - q\| = r\}$. The *closed disk* centered at p with radius r is the set of points at distance at most r from p: $\overline{\mathsf{disk}}_r(p) := \mathsf{circle}_r(p) \cup \mathsf{disk}_r(p) = \{q : \|p - q\| \leq r\}$. We abbreviate $\mathsf{disk}(p, q) := \mathsf{disk}_{\|p-q\|}(p)$, $\mathsf{circle}(p, q) := \mathsf{circle}_{\|p-q\|}(p)$, $\overline{\mathsf{disk}}(p, q) := \overline{\mathsf{disk}}_{\|p-q\|}(p)$. The *unit disk* of point p is $\mathsf{disk}_1(p)$.

The *lens* of two points p and q is the intersection of their unit disks: $\mathsf{lens}(p, q) := \mathsf{disk}_1(p) \cap \mathsf{disk}_1(q)$. The *cone* of two points p and q is defined as the locus of all the rays with origin in p that pass through $\mathsf{lens}(p, q)$ (the apex is p and the base is $\mathsf{lens}(p, q)$): $\mathsf{cone}(p, q) := \{r : \exists s \in \mathsf{lens}(p, q).r \in \mathsf{ray}(p, s)\}$, where $\mathsf{ray}(p, q) := \{p + \gamma(q - p) : \gamma \geq 0\}$.

A *configuration* $C = \langle I, F \rangle$ is an undirected graph where an agent $i \in I$ has a *source* coordinate $s_i \in \mathbb{R}^2$, a *target* coordinate $t_i \in \mathbb{R}^2$ at distance $d_i = \|s_i - t_i\|$, and every pair of neighboring agents $(i, j) \in F$ are *source-connected* (*i.e.*, $\|s_i - s_j\| \leq r$) where r is the communication radius. We say a configuration C is a *chain* (resp. *cycle*) if the graph is a simple path (resp. cycle).

3 Distributed Connectivity Service

In this section we present a distributed algorithm for refining an arbitrary motion plan into a plan that moves towards the intended goal and preserves global connectivity. No assumptions are made about trajectories generated by the motion planner, the connectivity service only needs to know the current and target positions and produces a straight line trajectory at each round; the composed trajectory observed over a series of rounds need not be linear. The trajectories output by the service are such that connectivity is preserved even if an adversary is allowed to stop or control the speed of each agent independently.

The algorithm is parametrized by a filtering function that determines a sufficient subset of neighbors such that maintaining 1-hop connectivity between those neighbors preserves global connectivity. The algorithm is *oblivious* because it is stateless and only needs access to the current plan, hence it is resilient to changes in the plan over time.

3.1 The Filtering Function

Assuming the communication graph is connected, we are interested in a FILTER subroutine that determines which edges can be removed while preserving connectivity. Let s be the position of an agent with a set N of 1-hop neighbors, we require a function $\text{FILTER}(N, s)$ that returns a subset of neighbors $N' \subseteq N$ such that preserving connectivity with the agents in the subset N' is sufficient to guarantee connectivity.

We say i and j are *symmetric neighbors* if FILTER determines i should preserve j ($s_j \in N_i'$) and vice versa ($s_i \in N_j'$). A FILTER function is *valid* if preserving connectivity of all symmetric edges is sufficient to preserve global connectivity. Observe that FILTER need not be symmetric in the sense that it may deem it necessary for i to preserve j as a neighbor, but not the other way around.

The identity function $\text{FILTER}(N, s) := N$ is trivially valid because connectivity is preserved if no edges are removed. However we ideally want a FILTER function that in some way "minimizes" the number of edges kept. A natural choice is to compute the minimum spanning tree (MST) of the graph, and return for every agent the set of neighbors which are its one hop neighbors in the MST. Although in some sense this would be the ideal filtering function, it cannot be computed locally and thus it is not suited for the connectivity service.

Nevertheless, there are well known local algorithms that compute sparse connected spanning subgraphs, amongst them is the Gabriel graph (GG) [5], the relative neighbor graph (RNG) [13], and the local minimum spanning tree ($LMST$) [8]. All these structures are connected and can be computed using local algorithms. Since we are looking to remove as many neighbors as possible and $MST \subseteq LMST \subseteq RNG \subseteq GG$, from the above $LMST$ is best suited.

Remark The connected subgraph represented by symmetric filtered neighbors depends on the positions of the agents, which can vary from one round to the next. Hence the use of a filtering function enables preserving connectivity without preserving a fixed set of edges (topology) throughout the execution; in fact, it is possible that no edge present in the original graph appears in the final graph.

3.2 The Algorithm

We present a three-phase service (fig. 1) that consists of a collection phase, a proposal phase, and an adjustment phase. In the collection phase, each agent queries the motion planner and the location

Collection phase:
$s_i \leftarrow \mathit{query\_positioning\_device}()$
$t_i \leftarrow \mathit{query\_motion\_planner}()$
broadcast s_i to all neighbors
$N_i \leftarrow \{s_j \mid \text{for each } s_j \text{ received}\}$

Proposal Phase:
$N'_i \leftarrow \textsc{Filter}\ (N_i, s_i)$
$R_i \leftarrow \bigcap_{s_j \in N'_i} \mathsf{disk}_1(s_j)$
$p_i \leftarrow \operatorname{argmin}_{p \in R_i} \|p - t_i\|$
broadcast p_i to all neighbors
$P_i \leftarrow \{p_j \mid \text{for each } p_j \text{ received}\}$

Adjustment Phase:
if $\forall s_j \in N'_i . \|p_j - p_i\| \leq r$ **then**
  return trajectory from s_i to p_i
else return trajectory from s_i to $s_i + \frac{1}{2}(p_i - s_i)$

Figure 1: ConnServ algorithm run by agent i.

service to obtain its current and target positions (s_i and t_i respectively). Each agent broadcasts its position and records the position of neighboring agents discovered within its communication radius.

In the proposal phase, the service queries the Filter function to determine which neighboring agents are sufficient to preserve connectivity. Using the neighbors returned by Filter the agent optimistically chooses a target p_i. The target is optimistic in the sense that if none of its neighboring agents move, then moving from source s_i to the target p_i would not disconnect the network. The proposed target p_i is broadcast and the proposals of other agents are collected.

Finally in the adjustment phase, each agent checks whether neighbors kept by the Filter function will be reachable after each agent moves to their proposed target. If every neighbor will be reachable, then the agent moves from the current position to its proposed target, otherwise it moves halfway to its proposed target, which ensures connectivity is preserved (proved in the next section).

4 Preserving Connectivity

In this section we prove the algorithm preserves network connectivity with any valid Filter function. Observe that since R_i is the intersection of a set of disks that contain s_i, it follows that R_i is convex and contains s_i. By construction $p_i \in R_i$ and thus by convexity the linear trajectory between s_i and p_i is contained in R_i, so the graph would remain connected if agent i were to move from s_i to p_i and every other agent would remain in place. The following theorems prove a stronger property, namely that the trajectories output guarantee symmetric agents will remain connected, even if they slow down or stop abruptly at any point of their trajectory.

Lemma 4.1 (Adjustment)**.** *The adjusted proposals of symmetric neighbors are connected.*

Proof. The adjusted proposals of symmetric agents i and j are $p'_i = s_i + \frac{1}{2}(p_i - s_i)$ and $p'_j = s_j + \frac{1}{2}(p_j - s_j)$. By construction $\|s_i - p_j\| \leq r$ and $\|s_j - p_i\| \leq r$, so the adjusted proposals are connected:

$$\|p_i' - p_j'\| = \|s_i - s_j + \frac{1}{2}(p_i - p_j + s_j - s_i)\| = \|\frac{1}{2}(s_i - s_j + p_i - p_j)\| \le \frac{1}{2}(\|s_i - p_j\| + \|s_j - p_i\|) \le r$$

□

Safety Theorem. *If* Filter *is valid, the service preserves connectivity of the graph.*

Proof. Assuming Filter is valid, it suffices to prove that symmetric neighbors remain connected after one round of the algorithm. Fix symmetric neighbors i and j. If $\|p_i - p_j\| > r$, both adjust their proposals and they remain connected by the Adjustment lemma. If $\|p_i - p_j\| \le r$ and neither adjust, they trivially remain connected. If $\|p_i - p_j\| \le r$ but (without loss of generality) i adjusts but j doesn't adjust, then $s_i, p_i \in \mathsf{disk}_1(p_j)$, and by convexity $p_i' \in \mathsf{disk}_1(p_j)$, whence $\|p_i' - p_j\| \le r$. □

Even if two agents are connected and propose connected targets, they might disconnect while following their trajectory to the target. Moreover, agents could stop or slow down unexpectedly (perhaps due to an obstacle) while executing the trajectories. We prove the linear trajectories prescribed by the algorithm for symmetric neighbors are *robust* in that any number of agents can stop or slow down during the execution and connectivity is preserved.

Robustness Theorem. *The linear trajectories followed by symmetric neighbors are robust.*

Proof. Fix symmetric neighbors i and j. Since each trajectory to the proposal is linear, we need to prove that all intermediate points on the trajectories remain connected. Fix points $q_i := s_i + \gamma_i(p_i - s_i)$ and $q_j := s_j + \gamma_j(p_j - s_j)$ ($\gamma_i, \gamma_j \in [0, 1]$) on the trajectory from each source to its proposal. Since the neighbors are symmetric, $s_i, t_i \in \mathsf{disk}_1(s_j) \cap \mathsf{disk}_1(t_j)$ and by convexity $q_i \in \mathsf{disk}_1(s_j) \cap \mathsf{disk}_1(t_j)$. Similarly $s_j, t_j \in \mathsf{disk}_1(q_i)$ and by convexity $q_j \in \mathsf{disk}_1(q_i)$, whence $\|q_i - q_j\| \le r$. □

5 Ensuring Progress for Graphs

For the algorithm to be useful, in addition to preserving connectivity (proved in §4) it should also guarantee that the agents make progress and eventually reach their intended destination. However, before proving any progress guarantee we first identify several subtle conditions without which no local algorithm can both preserve connectivity and guarantee progress.

Cycles Consider a configuration where nodes are in a cycle, two neighboring nodes want to move apart and break the cycle and every other node wants to remain in place. Clearly no local algorithm can make progress because, without global information, nodes cannot distinguish between being in a cycle or a chain, and in the latter case any movement would violate connectivity. As long as the longest cycle of the graph is bounded by a known constant, say k, using local LMST filtering over $\lfloor k/2 \rfloor$-hops will break all cycles. A way to deal with graphs with arbitrary long cycles without completely sacrificing locality would be to use the algorithm proposed in this paper and switch to a global filtering function to break all cycles when nodes detect no progress has been made for some number of rounds. For proving progress, in the rest of the paper we assume there are no cycles in the filtered graph.

Target-connectedness If the proposed targets are disconnected, clearly progress cannot be achieved without violating connectivity, hence it's necessary to assume that the target graph is connected. For simplicity, in the rest of the paper we assume that the current graph is a subgraph of the target graph, this avoids reasoning about filtering when proving progress and one can check that as a side effect the adjustment phase is never required.

5.1 Dependency graphs

Fix some node in an execution of the $\mathsf{ConnServ}$ algorithm, on how many other nodes does its trajectory depend? Let $\mathsf{region}(S) := \bigcap_{s \in S} \mathsf{disk}_1(s)$ and let $\mathsf{proposal}(S,t) := \text{argmin}_{p \in \mathsf{region}(S)} \|p - t\|$, then a node with filtered neighbor set N' and target t depends on k neighbors (has dependency k) if there exists a subset $S \subseteq N'$ of size $|S| = k$ such that $\mathsf{proposal}(S,t) = \mathsf{proposal}(N',t)$ but $\mathsf{proposal}(S',t) \neq \mathsf{proposal}(N',t)$ for any subset $S' \subseteq N'$ of smaller size $|S'| < k$.

The dependency of a node can be bounded by the size of its filtered neighborhood. If the filtering function is LMST then the number of neighbors is at most 6 or 5 depending on whether the distances to neighbors are unique (*i.e.*, breaking ties using unique identifiers). The following lemma gives a tighter upper bound on the dependency of a neighbor which is independent of the filtering function.

Lemma 5.1. *Every agent depends on at most two neighbors.*

Proof. Fix agent i with filtered neighbors N' and target t, let $R = \mathsf{region}(N')$. If $t \in R$ then $\mathsf{proposal}(N',t) = \mathsf{proposal}(\varnothing,t) = t$ and agent i depends on no neighbors. If $t \notin R$ then $\mathsf{proposal}(N',t)$ returns a point p in the boundary of region R. Since R is the intersection of a finite set of disks it follows that p is either in the boundary of a single disk, in which case i depends on a single neighbor, or the intersection of two disks, in which case i depends on at most two neighbors. $\square$

Given the above, for any configuration $C = \langle I, F\rangle$ we can consider its dependency graph $D = \langle I, E\rangle$ where there exists a directed edge $(u,v) \in E$ iff node u depends on node v. Hence D is a directed subgraph of C with maximum out-degree 2. Moreover since graphs with cycles cannot, in general, be handled by any local connectivity service, then for the purpose of proving progress we assume C has no undirected cycles. This implies that the only directed cycles in D are simple cycles of length 2, we refer to such dependency graphs as *nice* graphs.

A *prechain* H is a sequence of vertices $\langle v_i\rangle_{i \in 1..n}$ such that there is a simple cycle between v_i, v_{i+1} ($i \in 1..n-1$), observe that a vertex v is a singleton prechain. Below we prove that any nice dependency graph D contains a nonempty prechain H with no out-edges.

Theorem 5.2. *Every finite nice graph $G = \langle V, E\rangle$ contains a nonempty prechain $H \subseteq V$ with no out-edges.*

Proof. Fix a graph $G = \langle V, E\rangle$ and consider the graph G' that results from iteratively contracting the vertices $u, v \in V$ if $(u,v) \in E$ and $(v,u) \in E$. Clearly G' is also a finite nice graph and any vertex v' in G' is a prechain of G, however G' does not contain any directed cycles.

We follow a directed path in G' starting at an arbitrary vertex u', since the graph is finite and contains no cycles, we must eventually reach some vertex v' with no outgoing edges, such a vertex is a prechain and has no outgoing edges, which implies the lemma. $\square$

Therefore by theorem 5.2 any lower bound on progress for chains also holds for general configurations. In particular the lower bound of $\Omega(\min(d,r))$ for chains proved in the next section applies for general graphs as well.

6 Ensuring Progress for Chains

In this section we restrict our attention to chain configurations and show that, if agents execute the connectivity service's refined plan, the total progress of the configuration is at least $\min(d,r)$, where d is the minimum distance between any agent and its target and r is the communication radius. We introduce some terminology to classify chains according to their geometric attributes, then we prove the progress bound for a very restricted class of chains. Finally, we establish the result for all chains by showing that the progress of an arbitrary chain is bounded below by the progress of a restricted chain.

Terminology Each agent has a local coordinate system where the source is the origin ($s_i = \langle 0, 0\rangle$) and the target is directly above it ($t_i = \langle 0, d_i\rangle$). The left side of agent i is defined as $L_i := \{\langle x, y\rangle : x \le 0\}$ and the right side as $R_i := \{\langle x, y\rangle : x > 0\}$ where points are relative to the local coordinate system. An agent in a chain is *balanced* if it has one neighbor on its left side, and the other on its right side; a configuration is balanced if every agent is balanced.

A configuration is *d-uniform* if every agent is at distance d from its target ($d_i = d$ for every agent i). Given a pair of agents i and j, they are *source-separated* if $\|s_i - s_j\| = 1$; they are *target-separated* if $\|s_i - s_j\| = 1$; and they are *target-parallel* if the rays $\mathsf{ray}(s_i, t_i)$ and $\mathsf{ray}(s_j, t_j)$ are parallel. An agent i with neighbors j and k is *straight* if s_i, s_j, and s_k are collinear; a chain configuration is straight if all agents are straight.

Given an agent with source s, target t and a (possibly empty) subset of neighbors $S \subseteq N$, its proposed target w.r.t. S is defined as $t^* = \mathsf{proposal}(S, t)$. The *progress* of the agent would be $\delta(s, t; S) := \|s - t\| - \|t^* - t\|$, which we abbreviate as δ_i for agent i when the s_i, t_i and S_i are clear from context. Observe that since $\mathsf{region}(S \cup S') \subseteq \mathsf{region}(S)$, $\delta(s, t; S \cup S') \le \delta(s, t; S)$. The *progress* of a configuration C is the sum of the progress of the individual agents: $\mathsf{prog}(C) := \sum_i \delta_i$.

Proof Overview We first characterize the progress of agents in a balanced and source-separated chain and show the progress bound specifically for chains that are d-uniform, source- and target-separated, balanced, and straight (§6.2). Then we show how to remove each the requirements of a chain being straight, balanced, source- and target-separated, and d-uniform (§6.3). Ultimately, this means that an *arbitrary* target-connected chain configuration $C = \langle I, F\rangle$ can be transformed into a d-uniform, source- and target-separated, balanced, straight chain configuration C' such that $\mathsf{prog}(C) \ge \mathsf{prog}(C') \ge \min(d, r)$, where d is the minimum distance between each source and its target in the original configuration C ($d := \min_{i \in I} \|s_i - t_i\|$) and the communication radius is r. At each removal step we show that imposing a particular constraint on a more relaxed configuration does not increase progress, so that the lower bound for the final (most constrained) configuration is also a lower bound for the original (unconstrained) configuration. The bound shows that straight chains (the most constrained configurations) are the worst-case configurations since their progress is a lower bound for all chains. We show the lower bound is tight for d-uniform configurations by exhibiting a chain with progress exactly $\min(d, r)$ (§6.3).

6.1 Progress Function for Balanced and Separated Chains

We explicitly characterize the progress of an agent in a balanced, source-separated chain. In such a configuration, if an agent has source s with target t, the source-target distance is $d := \|s - t\|$ and the position of its neighbors s_{-1}, s_{+1} (if any) can be uniquely determined by the angles of the left ($\lambda := \angle t, s, s_{-1}$) and right neighbor ($\rho := \angle t, s, s_{+1}$). Since an agent's progress is determined by it's neighbors, its progress can be defined as a function $\delta^{\angle}(d, \lambda, \rho)$.

If the agent doesn't depend on either neighbor, it can immediately move to its target and its progress is d. If it (partially) depends on a single (left or right) neighbor at angle θ, then progress is $\delta^{\mathsf{single}}(d, \theta) := d + 1 - \sqrt{1 + d^2 - 2d\cos\theta}$. If it (partially) depends on both neighbors at angles ρ and λ, then progress is $\delta^{\mathsf{both}}(d, \lambda, \rho) := d - \sqrt{2 + d^2 - 2d\cos\rho + 2\cos(\rho + \lambda) - 2d\cos\lambda}$. If it is completely immobilized by one or both of its neighbors, its progress is 0. Therefore the progress of an agent can be fully described by the following piecewise function, parametrized by the source-target distance d and the angle to its neighbors ρ and λ. Observe that agent i's progress function is monotonically decreasing in ρ and λ.

$$\delta^{\angle}(d,\lambda,\rho) = \begin{cases} d & \text{depends on neither: } \rho \leq \cos^{-1}\frac{d}{2} \text{ and } \lambda \leq \cos^{-1}\frac{d}{2} \\ \delta^{\mathsf{single}}(d,\rho) & \text{depends on right: } \rho > \cos^{-1}\frac{d}{2} \text{ and } \sin(\rho+\lambda) \geq d\sin\lambda \\ \delta^{\mathsf{single}}(d,\lambda) & \text{depends on left: } \lambda > \cos^{-1}\frac{d}{2} \text{ and } \sin(\rho+\lambda) \geq d\sin\rho \\ \delta^{\mathsf{both}}(d,\lambda,\rho) & \text{depends on both: } \rho+\lambda < \pi \text{ and } \sin(\rho+\lambda) < d\sin\rho, d\sin\lambda \\ 0 & \text{immobilized by either or both: } \rho+\lambda \geq \pi \end{cases}$$

6.2 Progress for Restricted Chains

We prove a lower bound on progress of $\min(d,r)$ for d-uniform, source- and target-separated, balanced, straight chains with communication radius r. Let $C_k(d,\theta)$ represent a d-uniform, source- and target-separated, straight chain of k nodes, where $\angle t_i, s_i, s_{i+1} = \theta$ for $i \in 1..n-1$. We first establish the progress bound for chains of two nodes and then extend it to more than two nodes.

Progress Theorem for Restricted 2-Chains. *For any $\theta \in [0,\pi]$, the chain $C_2(d,\theta)$ makes progress at least $\min(d,r)$ ($\mathsf{prog}(C_2(d,\theta)) \geq \min(d,r)$).*

Proof. Suppose $\theta \leq \arccos\frac{d}{2}$, then if $d \leq r$ agent 1 makes progress d, if $d > r$ then agent 1 makes progress at least r. Similarly if $\theta \geq \pi - \arccos\frac{d}{2}$ and $d \leq r$ agent 2 makes progress d, if $d > r$ then agent 2 makes progress at least r. Otherwise $\theta \in (\arccos\frac{d}{2}, \pi - \arccos\frac{d}{2})$ and the progress function from §6.1 yields

$$\delta^{\mathsf{single}}(d,\theta) + \delta^{\mathsf{single}}(d,\pi-\theta) = 2 + 2d - \sqrt{1+d^2-2d\cos\theta} - \sqrt{1+d^2+2d\cos\theta}$$

The partial derivative is $\partial_\theta(\delta_1+\delta_2) = d\sin\theta(1/\sqrt{1+d^2+2d\cos\theta} - 1/\sqrt{1+d^2-2d\cos\theta})$, whose only root in $(0,\pi)$ is $\theta = \frac{\pi}{2}$, which is a local minimum. By using the first order Taylor approximation as an upper bound of $\sqrt{1+d^2}$ and since $d^2 < d$:

$$\mathsf{prog}(C_2(d,\theta)) \geq \mathsf{prog}(C_2(d,\frac{\pi}{2})) \geq 2\delta^{\mathsf{single}}(d,\frac{\pi}{2}) \geq 2+2d-2\sqrt{1+d^2} \geq 2+2d-2-d^2 \geq d$$

□

Progress Theorem for Restricted n-Chains. *Configurations $C_n(d,\theta)$ ($n > 2$) and $C_2(d,\theta)$ have the same progress ($\mathsf{prog}(C_n(d,\theta)) = \mathsf{prog}(C_2(d,\theta))$).*

Proof. Since C_n is straight and separated, internal nodes make no progress ($\delta_i = 0$ for $i \in 2..n-1$). The first node in C_n (and C_2) has a single neighbor at angle θ, so $\delta_1^n = \delta_1^2$. Similarly the last node in C_n (and C_2) has a single neighbor at angle $\pi-\theta$, so $\delta_n^n = \delta_2^2$. Therefore $\mathsf{prog}(C_n(d,\theta)) = \mathsf{prog}(C_2(d,\theta))$. □

6.3 Progress for Arbitrary Chains

We prove that the progress of an *arbitrary* chain is bounded below by the progress of a restricted chain, hence the progress bound proved in the previous section for restricted chains extends to all chains. Furthermore, we show the bound is tight for d-uniform configurations by exhibiting a class of chains for which progress is exactly $\min(d,r)$.

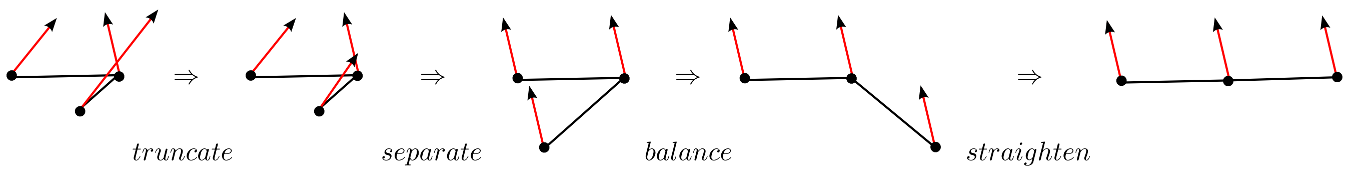

Figure 2: Transformation overview from arbitrary to restricted chains.

To extend the progress result from restricted to arbitrary chains, we exhibit a sequence of transformations (cf. fig. 2) that show how to transform an arbitrary chain to be d-uniform, source-separated, target-separated, balanced, and finally also straight. Each transformation doesn't increase progress and preserves the configuration's properties.

To warm up, observe that if an agent i and its immediate neighbors are subjected to the same rigid transformation (rotations and translations), then its progress δ_i is unchanged. Therefore if all agents $i \leq k$ are subjected to the same rigid transformation and all agents $i > k$ remain fixed, every progress value δ_i ($i \neq k, k+1$) will be unaffected and the only progress values that may change are δ_k, δ_{k+1}.

Proposition 6.1 (Reflected distance)**.** *Suppose p, q are on the same side of a line L and let q' be the reflection of q in L. Then $\|q' - p\| \geq \|q - p\|$.*

Proof. Consider the coordinate system with L as the y axis. Then $p = \langle p_x, p_y \rangle, q = p + \Delta$ for some $\Delta = \langle \Delta_x, \Delta_y \rangle$, and $q' = \langle -p_x - \Delta_x, p_y + \Delta_y \rangle$. Therefore $\|q' - p\| = \sqrt{4p_x^2 + \|\Delta\|^2} \geq \|\Delta\| = \|q - p\|$. □

Proposition 6.2 (Unrestricted movement)**.** *Suppose source s with target t has neighbors $S = \{s_i\}_{i \in 1..n}, s_{n+1}, s'_{n+1}$ and doesn't depend on neighbor s_{n+1}. Then $\delta(s_i, t_i; S \cup \{s'_{n+1}\}) \leq \delta(s_i, t_i; S \cup \{s_{n+1}\})$.*

Proof. Since s doesn't depend on s_{n+1}, $\delta(s_i, t_i; S \cup \{s'_{n+1}\}) \leq \delta(s_i, t_i; S) = \delta(s_i, t_i; S \cup \{s_{n+1}\})$. □

Proposition 6.3 (Restricted movement)**.** *Suppose source s with target t has neighbors $S = \{s_i\}_{i \in 1..n}, s_{n+1}, s'_{n+1}$ and only depends on agent s_{n+1}. If $\|s'_{n+1} - t\| \geq \|s_{n+1} - t\|$, then $\delta(s, t; S \cup \{s'_{n+1}\}) \leq \delta(s, t; S \cup \{s_{n+1}\})$.*

Proof. Since s only depends on s_{n+1}, its proposal is the point $t^* \in \mathsf{disk}_1(s_{n+1})$ closest to t. Since $\|s'_{n+1} - t\| \geq \|s_{n+1} - t\|$, the point $t'^* \in \mathsf{disk}_1(s'_{n+1})$ closest to t has distance $\|t'^* - t\| \geq \|t^* - t\|$, so $\delta(s, t; S \cup \{s'_{n+1}\}) \leq \delta(s, t; S \cup \{s_{n+1}\})$. □

6.3.1 Truncating

For an agent with source s and target t consider a *truncated* target $t^T = s + \gamma(t - s)$ where $\gamma \in [0, 1]$. We prove prove that truncating preserves target-connectedness and doesn't increase progress. Therefore a configuration can be assumed to be uniform by truncating to ensure the source-target distance is $\min(d, r)$ where d is the minimum source-target distance ($d := \min_{i \in I} \|s_i - t_i\|$).

Truncation Theorem. *Suppose a source s with target t and neighbors S. Let $t^T = s + \gamma(t - s)$ with $\gamma \in [0, 1]$ be its truncated target, then $\delta(s, t; S) \geq \delta(s, t^T; S)$.*

Proof. Let a be the proposal of source s and target t, and let a^T be the proposal of source s using the truncated target t^T. By the definition of proposal, a is the point in $\mathsf{region}(S)$ that minimizes

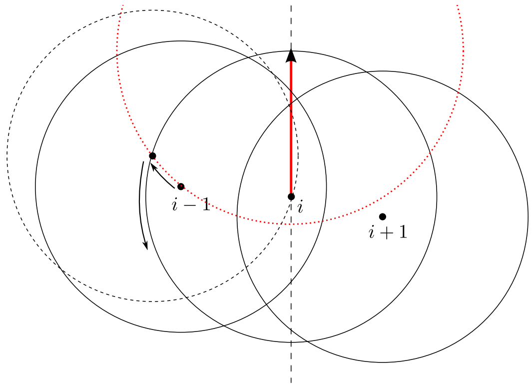

Figure 3: Separating s_{i-1} and s_i by the independence lemma.

the distance to t, so $\|a-t\| \leq \|a^T-t\|$. By the triangle inequality $\|a^T-t\| \leq \|a^T-t^T\|+\|t^T-t\|$ and $\|t-t^T\| = (1-\gamma)\|t-s\|$,

$$\begin{aligned}\delta(s,t;S) &= \|s-t\|-\|a-t\| \geq \|s-t\|-\|a^T-t\| \geq \|s-t\|-\|a^T-t^T\|-\|t^T-t\| \\ &= \gamma\|s-t\|+(1-\gamma)\|s-t\|-\|a^T-t^T\|-(1-\gamma)\|s-t\| \\ &= \|\gamma(s-t)\|-\|a^T-t^T\| = \|s-(s+(\gamma(t-s)))\|-\|a^T-t^T\| = \delta(s,t^T;S)\end{aligned}$$

□

Target-connectedness Theorem. *Let neighboring agents i and j be truncated to targets t_i^T and t_j^T with the same source-target distance $d \in [0, \min(\|s_i-t_i\|, \|s_j-t_j\|)]$. If the non-truncated agents are target-connected ($\|t_i-t_j\| \leq 1$), then the truncated agents are also connected ($\|t_i^T-t_j^T\| \leq 1$).*

Proof. Let $\gamma_i = \frac{d}{\|s_i-t_i\|}$ and $\gamma_j = \frac{d}{\|s_j-t_j\|}$, then the truncated targets are $t_i^T = s_i+\gamma_i(t_i-s_i)$ and $t_j^T = s_j+\gamma_i(t_j-s_j)$. Observe that since $d \in [0, \min(\|s_i-t_i\|, \|s_j-t_j\|)]$, $\gamma_i, \gamma_j \leq 1$. We can choose $\gamma' \in \{\gamma_i, \gamma_j\}$ so that $\gamma' \leq 1$ and,

$$\begin{aligned}\|t_i^T-t_j^T\| &= \|s_i-s_j+\gamma_i(t_i-s_i)+\gamma_j(t_j-s_j)\| \leq \|s_i-s_j+\gamma'(t_i-s_i)+\gamma'(t_j-s_j)\| \\ &= \|(1-\gamma')(s_i-s_j)+\gamma'(t_i-t_j)\| \leq (1-\gamma')\|s_i-s_j\|+\gamma'\|t_i-t_j\| \leq (1-\gamma')+\gamma' = \gamma' \leq 1\end{aligned}$$

□

6.3.2 Separating

Here we further remove the requirement for chains to be source- and target-separated. We show that neighboring agents can be source-separated without increasing progress, as long as the separated source (*i.e.*, s_{i-1}) doesn't get closer to the other agent's target (*i.e.*, t_i). Furthermore, d-uniform neighbors can be source- and target-separated while satisfying the former proviso. Therefore a d-uniform configuration can be source- and target-separated without increasing progress.

Consider agent i in a chain configuration, with neighboring agents $i-1$ and $i+1$. We will describe a transformation that separates the sources and targets of the three agents, and decreases the progress of all of them. As a first step, we first prove there exists a transformation that separates each pair (agent i and $i-1$, and agent i and $i+1$) independently, decreases the progress of that pair, while leaving the progress of the other pair unchanged.

Lemma 6.4 (Independence)**.** *Suppose source* s_i *has target* t_i *and neighbors* $s_{i-1}, s'_{i-1} \in L_i$ *and* $s_{i+1} \in R_i$ *such that* $\|s'_{i-1} - s_i\| = 1$ *and* $\|s'_{i-1} - t_i\| \geq \|s_{i-1} - t_i\|$*. Let* $\delta_i := \delta(s_i, t_i; \{s_{i-1}, s_{i+1}\})$ *and* $\delta'_i := \delta(s_i, t_i; \{s'_{i-1}, s_{i+1}\})$*. Then* $\delta'_i \leq \delta_i$*.*

Proof. If $t_i \in \mathsf{cone}(s_{i+1}, s_{i-1})$ then progress decreases by proposition 6.2. If $t_i \in \mathsf{cone}(s_{i-1}, s_{i+1}) \setminus \mathsf{cone}(s_{i+1}, s_{i-1})$ progress decreases by proposition 6.3. Otherwise $t_i \notin \mathsf{cone}(s_{i+1}, s_{i-1}) \cup \mathsf{cone}(s_{i-1}, s_{i+1})$ and agent i depends on both neighbors, so s's proposal t^* is the upper corner of $\mathsf{lens}(s_{i-1}, s_{i+1})$ (cf. fig. 4). In what follows we show that progress of does not increase via the rotations in fig. 3. At a high level, progress doesn't increase by moving s_{i-1} to s^+_{i-1} by rotating counterclockwise around t_i until reaching $\mathsf{disk}_1(s_i)$, and then moving s^+_{i-1} to s'_{i-1} by rotating clockwise around $\mathsf{disk}_1(s_i)$.

Let s^*_{i-1} (resp. s^+_{i-1}) be the corner of $\mathsf{lens}(t^*, s_i)$ (resp. $\mathsf{lens}(t_i, s_i)$) at the intersection of $\mathsf{disk}_1(s_i)$ and $\mathsf{disk}_1(t^*)$ (resp. $\mathsf{disk}_1(t_i)$) closest to s_{i-1} and farthest from s_{i+1}, which can be expressed as $s^*_{i-1} = 1\angle\theta^*$ (resp. $s^+_{i-1} = 1\angle\theta^+$) for some $\theta^* \in [\frac{\pi}{2}, \frac{3\pi}{2}]$ (resp. $\theta^+ \in [\frac{\pi}{2}, \frac{3\pi}{2}]$) in i's polar coordinate system (cf. fig. 5). Define $s'(\theta) := 1\angle\theta$ and $\delta'(\theta) := \delta(s_i, t_i; \{s'(\theta), s_{i+1}\})$, and observe that $s^*_{i-1} = s'(\theta^*)$.

($s'_{i-1} = s'(\theta')$ for some $\theta' \in [\theta^*, \frac{3\pi}{2}]$) Let A^+ be the line segment between s_{i-1} and s^+_{i-1} and let B^+ be the plane bisector perpendicular to A^+, and observe that B^+ passes through t_i. Let A^* be the line segment between s_{i-1} and s^*_{i-1} and let B^* be the plane bisector perpendicular to A^*, and observe that B^* passes through t^* (cf. fig. 6). Since $t_i \notin \mathsf{cone}(s_{i+1}, s_{i-1}) \cup \mathsf{cone}(s_{i-1}, s_{i+1}) \ni t^*$, $\partial_x B^+ > \partial_x B^*$, so $\partial_x A^+ < \partial_x A^*$ and $\theta^+ \in [\theta^*, \frac{3\pi}{2}]$. Since $s'_{i-1} \in L_i$, $\|s'_{i-1} - s_i\| = 1$, and $\|s'_{i+1} - t\| \geq \|s_{i+1} - t\|$, there exists $\theta' \in [\theta^+, \frac{3\pi}{2}] \subseteq [\theta^*, \frac{3\pi}{2}]$ such that $s'_{i+1} = 1\angle\theta'$ in i's coordinate system (cf. fig. 7).

It suffices to show that $\delta_i = \delta'(\theta^*)$ and $\delta'(\theta)$ is monotonically decreasing on the interval $\theta \in [\theta^*, \pi]$, whence $\delta'_i = \delta'(\theta') \leq \delta'(\theta^*) = \delta_i$ because $\theta' \geq \theta^*$.

($\delta_i = \delta'(\theta^*)$) Since $t^* \in \mathsf{circle}_1(s_{i-1}) \cap \mathsf{circle}_1(s_{i+1}) \subseteq \mathsf{circle}_1(s_{i+1})$ and $s^*_{i-1} \in \mathsf{circle}_1(t^*) \cap \mathsf{circle}_1(s_i) \subseteq \mathsf{circle}_1(t^*)$, $t^* \in \mathsf{circle}_1(s^*_{i-1})$ and $t^* \in \mathsf{circle}_1(s^*_{i-1}) \cap \mathsf{circle}_1(s_{i+1})$. By the choice of s^*_{i-1} farthest from from s_{i+1}, t^* is the closest point in $\mathsf{lens}(s^*_{i-1}, s_{i+1})$ to t and s^*_{i-1}'s proposal, whence $\delta_i = \delta'(\theta^*)$.

($\delta'(\theta)$ is monotonically decreasing in θ) Let $t'(\theta)$ be the corner of the $\mathsf{lens}(s'(\theta), s_{i+1})$ closest to t_i and let $d'(\theta)$ be the point inside $\mathsf{disk}_1(s'(\theta))$ closest to t_i. By definition, $\|t'(\theta) - t_i\|$ and $\|d'(\theta) - t_i\|$ are monotonically increasing for $\theta \in [\theta^*, \frac{3\pi}{2}]$. Observe that $t^* = t'(\theta^*)$ and there exists $\theta^= \in [\theta^*, \frac{3\pi}{2}]$ such that $t'(\theta^=) = d'(\theta^=)$. Therefore

$$\delta'(\theta) = \begin{cases} 1 - t'(\theta) & \text{if } \theta \in [\theta^*, \theta^=] \\ 1 - d'(\theta) & \text{if } \theta \in]\theta^=, \frac{3\pi}{2}] \end{cases}$$

and by the monotonicity of t' and d' it follows that $\delta'(\theta) \leq \delta'(\theta^*)$ in the interval $\theta \in [\theta^*, \frac{3\pi}{2}]$. $\square$

If we want to source-separate agents i and $i+1$ without increasing the total progress, the Independence lemma says we can do so without worrying about the positions of $i-1$ or $i+2$, as long as the separation does not bring one agent's source closer to the other agent's target. Formally, if the initial configuration is $\langle s_i, t_i\rangle, \langle s_{i+1}, t_{i+1}\rangle$ and a transformation (*e.g.*, separation) moves $\langle s_i, t_i\rangle$ to $\langle s'_i, t'_i\rangle$, we need to ensure i's source doesn't get closer to $i+1$'s target ($\|s'_i - t_{i+1}\| \geq \|s_i - t_{i+1}\|$) and vice versa ($\|t'_i - s_{i+1}\| \geq \|t_i - s_{i+1}\|$), these two conditions are henceforth abbreviated as $\langle s'_i, t'_i\rangle \succeq_{\langle s_{i+1} t_{i+1}\rangle} \langle s_i, t_i\rangle$.

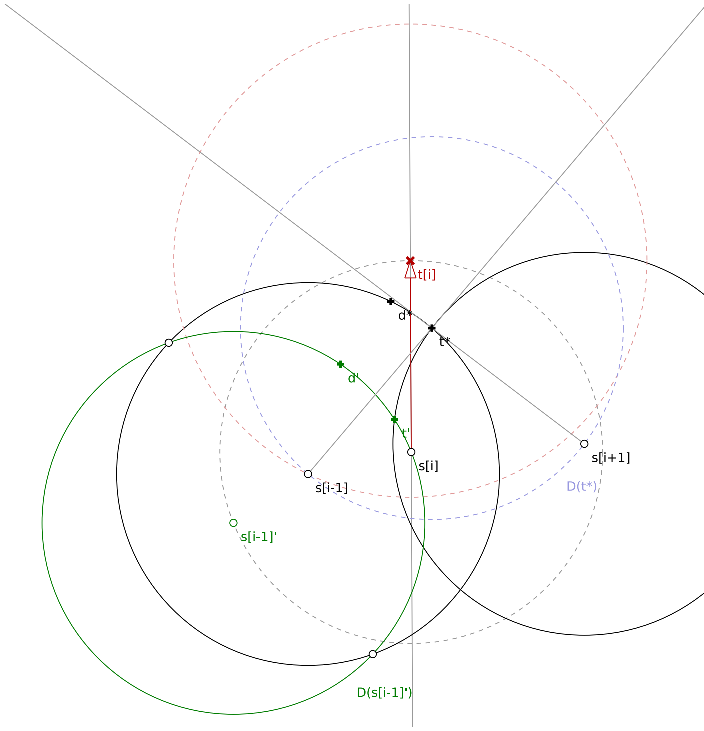

Figure 4: Independence: Initial source s'_i and final source s'_{i-1}.

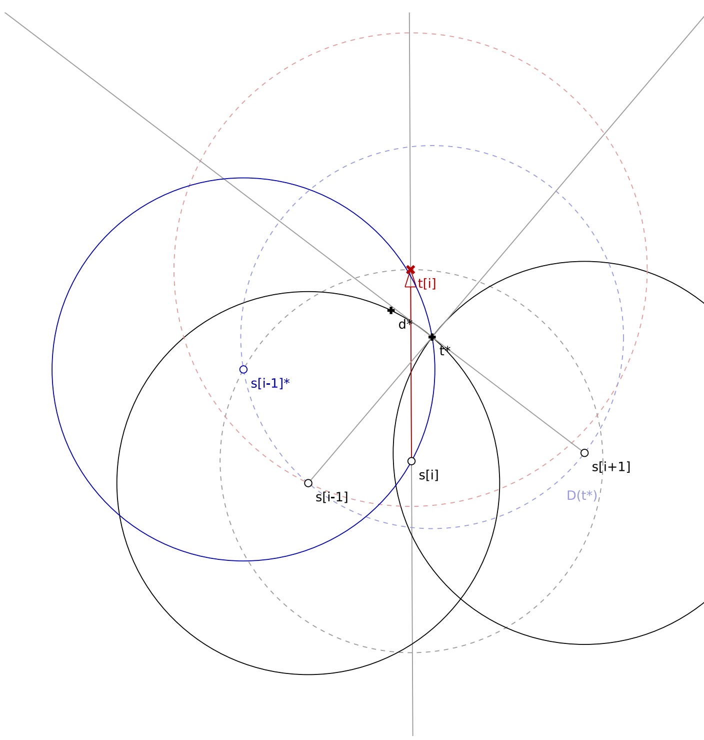

Figure 5: Independence: Initial source s'_i and intermediate source s^*_{i-1}.

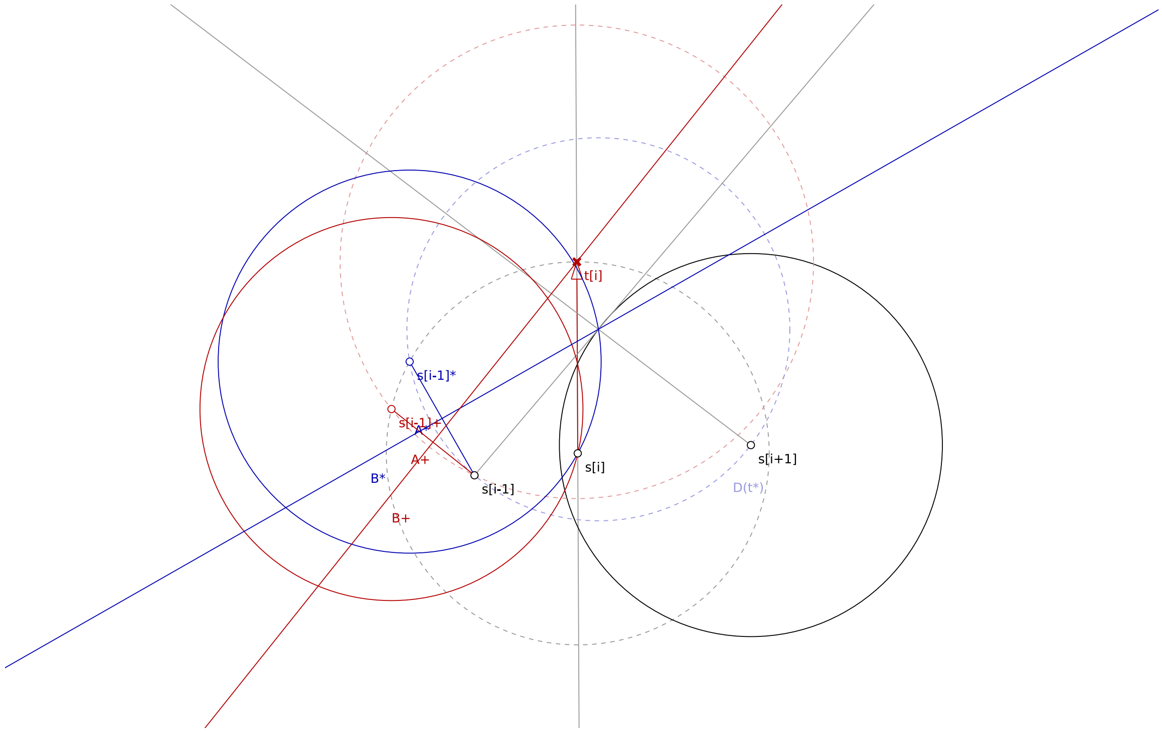

Figure 6: Independence: Intermediate sources s_{i-1}^{*} and s_{i-1}^{+}.

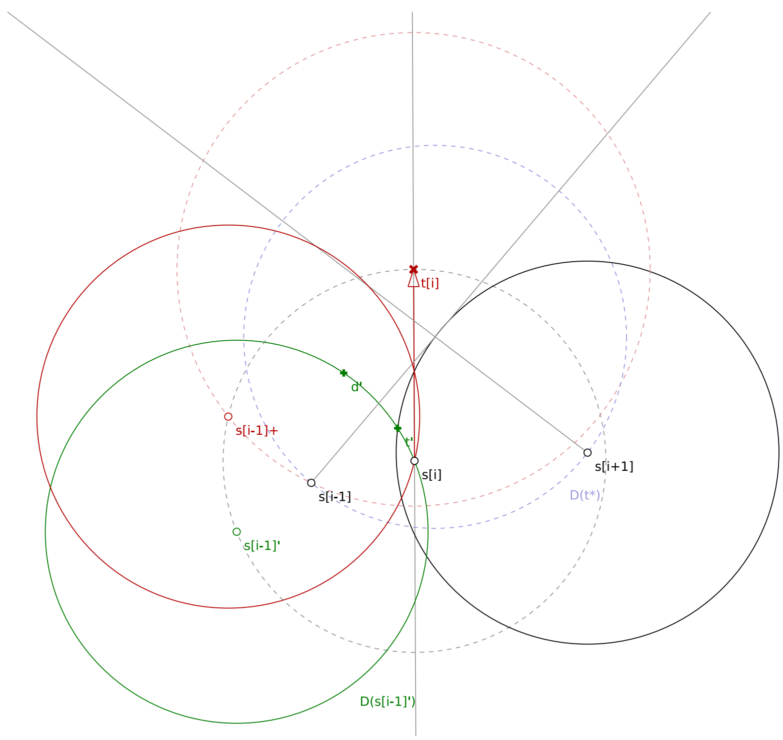

Figure 7: Independence: Intermediate source s_{i-1}^{+} and final source s_{i-1}'.

The following lemma proves that under some restrictions on the placements of the sources and targets, there exists a rigid transformation that source- and target-separates a pair of agents while ensuring that the source of the first agent is not closer to the target of the second, and vice versa.

Lemma 6.5. *Let $q^* := p + (q_0 - p_0), m_0 := \frac{1}{2}(q_0 - p_0), m := \frac{1}{2}(q - p), m^* := \frac{1}{2}(q^* - p)$ and suppose $\|p_0 - q_0\| = \|p - q\| = \|p - q^*\| = d \leq d'$, $p \in \overline{\mathsf{disk}}_{d'}(p_0)$, $q \in \overline{\mathsf{disk}}_{d'}(q_0)$, and $\|m_0 - m^*\| \geq \|m_0 - m\|$. Then there exist $p' \in \mathsf{circle}_{d'}(p_0), q' \in \mathsf{circle}_{d'}(q_0)$ such that $\|p' - q'\| = d$ and $\langle p', q'\rangle \succeq_{\langle p_0, q_0\rangle} \langle p, q\rangle$.*

Proof. Let A be the line passing through p and m_0. If q, q^* are on opposite sides of A, then let $q^\sharp$ be the reflection of q in A, otherwise let $q^\sharp := q$. By proposition 6.1, $\langle p^\sharp, q^\sharp\rangle \succeq_{\langle p_0, q_0\rangle} \langle p, q\rangle$. Define the midpoint $m^\sharp := \frac{1}{2}(q^\sharp - p)$ and observe $\|m_0 - m^\sharp\| = \|m_0 - m\| \leq \|m_0 - m^*\|$. In particular q^* results from the rotation of $q^\sharp$ about p away from p_0, so $\langle p^*, q^*\rangle \succeq_{\langle p_0, q_0\rangle} \langle p^\sharp, q^\sharp\rangle$. Let $\langle p', q'\rangle$ be the translation of $\langle p^*, q^*\rangle$ along the line from m_0 to m^* until $\|p' - p_0\| = d' = \|p' - p_0\|$. Then $p' \in \mathsf{circle}_{d'}(p_0), q' \in \mathsf{circle}_{d'}(q_0)$, $\|p' - q'\| = \|p^* - q^*\| = d$, and $\langle p', q'\rangle \succeq_{\langle p_0, q_0\rangle} \langle p^*, q^*\rangle \succeq_{\langle p_0, q_0\rangle} \langle p^\sharp, q^\sharp\rangle \succeq_{\langle p_0, q_0\rangle} \langle p, q\rangle$. ☐

Proposition 6.6. *Suppose $p = \langle x, y\rangle, q^+ = \langle d, 0\rangle, q^- = \langle -d, 0\rangle$, then $\|p - q^+\| \geq \|p\|$ or $\|p - q^-\| \geq \|p\|$.*

Proof. Either $|x+d|$ or $|x-d|$ is greater than $|x|$, so either $\|p - q^+\| = \sqrt{(x+d)^2 + y^2} \geq \sqrt{x^2 + y^2} = \|p\|$ or $\|p - q^-\| = \sqrt{(x-d)^2 + y^2} \geq \sqrt{x^2 + y^2} = \|p\|$. ☐

Lemma 6.7 (Local separation)**.** *Suppose $i-1, i$ are neighbors in a d-uniform configuration, then there is a rigid transformation of $\langle s_{i-1}, t_{i-1}\rangle$ into $\langle s'_{i-1}, t'_{i-1}\rangle$ that separates neighboring sources and targets ($\|s'_{i-1} - s_i\| = 1 = \|t'_{i-1} - t_i\|$) and doesn't decrease the distance between neighboring source and target ($\langle s'_{i-1}, t'_{i-1}\rangle \succeq_{\langle s_i, t_i\rangle} \langle s_{i-1}, t_{i-1}\rangle$).*

Proof. Define the intermediate agents $\langle s^+, t^+\rangle := \langle s_{i-1}, s_{i-1} + (t_i - s_i)\rangle$ and $\langle s^-, t^-\rangle := \langle t_{i-1} + (s_i - t_i), t_{i-1}\rangle$ that result from rotating agent $i-1$ about s_{i-1} and t_{i-1} until parallel with agent i. Let $m_i := \frac{1}{2}(t_i - s_i), m_{i-1} := \frac{1}{2}(t_{i-1} - s_{i-1}), m^+ := \frac{1}{2}(t^+ - s^+), m^- := \frac{1}{2}(t^- - s^-)$ be the respective midpoints of the agents. By proposition 6.6, there is $m^* \in \{m^+, m^-\}$ farther from m_i than m_{i-1} and by lemma 6.5 there is a rigid transformation of $\langle s^*, t^*\rangle$ to a $\langle s'_{i-1}, t'_{i-1}\rangle$ satisfying the conditions. ☐

Separation Theorem. *A d-uniform configuration C can be transformed into a d-uniform, source- and target-separated configuration C' such that $\mathsf{prog}(C') \leq \mathsf{prog}(C)$.*

Proof. By the Local Separation lemma, each pair of agents $i, i+1$ can be separated to be source- and target-separated. This satisfies the conditions of the Independence lemma for both agents, so the transformation doesn't increase the progress of each agent ($\delta'_i \leq \delta_i$ and $\delta'_{i+1} \leq \delta_{i+1}$) or the total progress ($\mathsf{prog}(C') \leq \mathsf{prog}(C)$). The Independence lemma can be applied to the endpoints (if $i = 1$ or $i + 1 = n$) by adding a dummy (left or right) neighbor with the same source and target ($i-1$ identical to i or $i+2$ identical to $i+1$). ☐

Remark Observe that in the resulting configuration the source-target vectors are parallel ($t_i - s_i = t_{i+1} - s_{i+1}$) and the configuration is source- and target-separated ($\|s_i - s_{i+1}\| = d_r = \|t_i - t_{i+1}\|$), so there exists a unique $\theta_i \in [0, \pi]$ such that $\theta_i := \angle s_i, s_{i+1}, t_{i+1}$ and this uniquely determines agent $i+1$ ($s_{i+1} = d_r \angle \frac{\pi}{2} - \theta_i$ and $t_{i+1} = s_{i+1} + t_i$) in i's coordinate system. In particular, a d-uniform configuration resulting from the previous lemma has $\|t_i - s_i\| = d$ and $d_r = 1$, so it is uniquely determined by the $n-1$ angles $\{\theta_i \in [0, \pi]\}_{i \in 1..n-1}$.

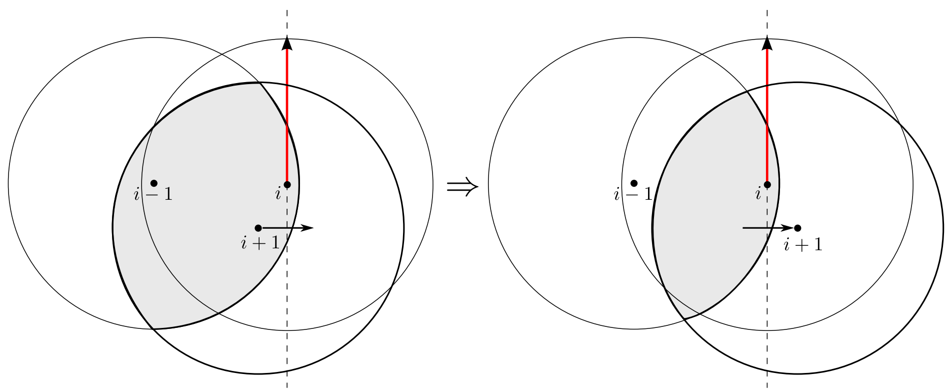

Figure 8: Balancing agent i by reflecting agent $i+1$.

6.3.3 Balancing

Now we show that agents can be balanced without increasing progress, hence the assumption that chains are balanced can be made without loss of generality. Observe that the transformation described preserves the properties of being d-uniform and source- and target-separated.

Balancing Theorem. *Consider a configuration C where agent i has neighbors $i-1$ and $i+1$ on the same side. Let C' be the configuration obtained by reflecting every s_j and t_j for $j > i$ (or $j < i$) around agent i's y-axis. Then $\mathsf{prog}(C') \leq \mathsf{prog}(C)$.*

Proof. Without loss of generality the neighbors $i-1$ and $i+1$ are on agent i's left side ($s_{i-1}, s_{i+1} \in L_i$) and C' is obtained by reflecting s_j, t_j for $j > i$ around i's y-axis (cf. Fig. 8). This transformation only affects the relative position of agent i's neighbors, so it suffices to show $\delta'_i \leq \delta_i$. Let t^* (resp. t'^*) be i's proposal in C (resp. C').

If $t'^* \notin \mathsf{circle}_1(s'_{i+1})$, then $t'^* = t^* \in \overline{\mathsf{disk}}_1(s_{i+1}) \cap \overline{\mathsf{disk}}_1(s'_{i+1})$ and $\delta'_i = \delta_i$.

If $t'^* \in \mathsf{circle}_1(s'_{i+1})$, then we consider whether t'^* is on the left or right of i. If $t'^* \in \mathsf{circle}_1(s'_{i+1}) \cap R_i$, then both $t^*, t'^* \in \overline{\mathsf{disk}}_1(s_{i-1})$ and t'^* is the reflection of t^* in i's y-axis, so $\delta'_i = \delta_i$. Otherwise if $t'^* \in \mathsf{circle}_1(s'_{i+1}) \cap L_i$, then t^* is the closest point in $\overline{\mathsf{disk}}_1(s_{i-1}) \cap \overline{\mathsf{disk}}_1(s_{i+1}) \cap L_i$ to t_i and t'^* is the closest point in $\overline{\mathsf{disk}}_1(s_{i-1}) \cap \overline{\mathsf{disk}}_1(s'_{i+1}) \cap L_i$ to t_i. Since $\overline{\mathsf{disk}}_1(s'_{i+1})$ is the reflection of $\overline{\mathsf{disk}}_1(s_{i+1})$ in i's y-axis, $\overline{\mathsf{disk}}_1(s_{i-1}) \cap \overline{\mathsf{disk}}_1(s'_{i+1}) \cap L_i \subseteq \overline{\mathsf{disk}}_1(s_{i-1}) \cap \overline{\mathsf{disk}}_1(s_{i+1}) \cap L_i$ whence $\delta'_i \leq \delta_i$.

□

6.3.4 Straightening

Figure 9 shows how a chain C^k whose first k agents are collinear can be further straightened by aligning the first $k+1$ agents to obtain chain C^{k+1}. The formal result of this section is this transformation can be iteratively used to straighten a d-uniform, source- and target-separated, balanced configuration without increasing progress.

Lemma 6.8 (Single Straightening)**.** *For $k \in 1..n-2$, let configuration C^k be described by $\{\theta^k_i\}_{i \in 1..n-1}$ where the first k angles are identical ($\theta^k_i = \theta^k_k$ for $i \leq k$), and configuration C^{k+1} be described by $\{\theta^{k+1}_i\}_{i \in 1..n-1}$ where the first $k+1$ angles are identical to the $k+1$ angle of C^k ($\theta^{k+1}_i := \theta^k_{k+1}$ for*

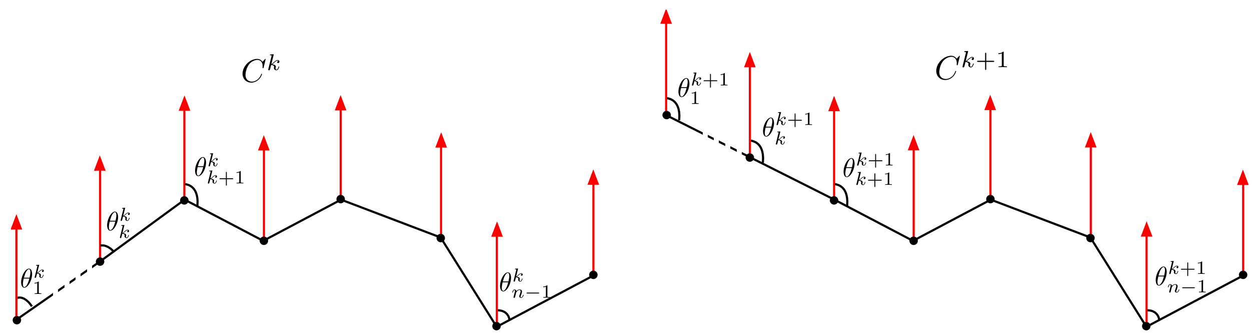

Figure 9: Chain C^k and C^{k+1}.

$i \leq k+1$) and the remaining angles are identical to the corresponding angles from C^k ($\theta_i^{k+1} := \theta_i^k$ for $i > k+1$). Then $\mathsf{prog}(C^{k+1}) \leq \mathsf{prog}(C^k)$.

Proof. For $i \in 2..k$ observe agent i in chain C^k is straight since $\theta_{i-1}^k = \theta_i^k$, hence $\delta_i^k = 0$. Similarly for $i \in 2..k{+}1$ agent i in chain C^{k+1} is straight since $\theta_{i-1}^{k+1} = \theta_i^{k+1}$, hence $\delta_i^{k+1} = 0$. For $i \in k{+}2..n{-}1$ observe that $\theta_{i-1}^k = \theta_{i-1}^k$ and $\theta_i^k = \theta_i^{k+1}$ hence it follows that $\delta_i^k = \delta_i^{k+1}$ and $\delta_{n-1}^k = \delta_{n-1}^{k+1}$. Therefore only for $i \in \{1, k+1\}$ we have $\delta_i^k \neq \delta_i^{k+1}$, so to prove $\mathsf{prog}(C^{k+1}) \leq \mathsf{prog}(C^k)$ it suffices to show $\delta_1^{k+1} + \delta_{k+1}^{k+1} \leq \delta_1^k + \delta_{k+1}^k$, but since $\delta_{k+1}^{k+1} = 0$ this reduces to proving $\delta_1^{k+1} \leq \delta_1^k + \delta_{k+1}^k$.

Straight ($\theta_k^k = \theta_{k+1}^k$). Then $C^{k+1} = C^k$ so $\mathsf{prog}(C^{k+1}) \leq \mathsf{prog}(C^k)$.

Convex ($\theta_k^k < \theta_{k+1}^k$). In this case s_i is on the upper corner of $\mathsf{lens}(s_{i-1}, s_{i+1})$, which is also its proposal, so $\delta_i(\theta_{i-1}, \theta_i) = 0$. Observe $\theta_1^{k+1} = \theta_{k+1}^k > \theta_k^k = \theta_1^k$, and δ_1 is monotonically decreasing in θ_1 it follows that $\delta_1^{k+1} \leq \delta_1^k$, and hence $\mathsf{prog}(C^{k+1}) \leq \mathsf{prog}(C^k)$.

Concave ($\theta_k^k > \theta_{k+1}^k$). In this case s_i is on the bottom corner of $\mathsf{lens}(s_{i-1}, s_{i+1})$ and its proposal is elsewhere on the lens, so $\delta_i(\theta_{i-1}, \theta_i) > 0$. We split into cases according to how agent $k+1$ depends on its neighbors in chain C^k.

Subcase agent $k+1$ doesn't depend on $k+2$. Then agent 1 and $k+1$ execute as if they were on a d-uniform, straight and separated subchain of length $k+1$, and such chains have progress at least d by the Progress lemmas for straight n- and 2-chains (§6.2). Therefore $\delta_1^k + \delta_k^k \geq d$ and $\mathsf{prog}(C^{k+1}) \leq \mathsf{prog}(C^k)$.

Subcase agent $k+1$ only depends on $k+2$. Since $\theta_1^{k+1} = \theta_{k+1}^k$ and agent $k+1$ only depends on $k+2$, $\delta_1^{k+1} = \delta_{k+1}^k \leq \delta_1^k + \delta_{k+1}^k$.

Subcase agent $k+1$ depends on k and $k+2$. This implies that $\theta_k + \theta_{k+1} \leq \pi$, using the progress definition of §6.1 we have $\delta_{k+1}^k := \delta^\angle(d, \pi - \theta_k, \theta_{k+1})$, $\delta_1^k := \delta^\angle(d, \theta_k, 0)$ and $\delta_1^{k+1} := \delta^\angle(d, \theta_{k+1}, 0)$.

Recall that $\delta^\angle(d, \theta, \phi)$ is monotonically decreasing w.r.t. θ and ϕ, hence $\theta_k < \theta_{k+1}$ implies $\delta_1^k - \delta_1^{k+1} \geq 0$ and thus $\delta_1^{k+1} \leq \delta_1^k + \delta_{k+1}^k$ would hold. Therefore assume $\theta_k \geq \theta_{k+1}$, this together with $\theta_k + \theta_{k+1} \leq \pi$ implies that $\theta_{k+1} \leq \frac{\pi}{2}$ and $\theta_k \geq \frac{\pi}{2}$.

Finally observe that analytically evaluating the minimum of $\delta^\angle(d, \pi - \theta_k, \theta_{k+1}) + \delta^\angle(d, \theta_k, 0) - \delta^\angle(d, \theta_{k+1}, 0)$ in that interval yields two minima. One at $\theta_k = \theta_{k+1} = \frac{\pi}{2}$ and another at $\theta_k = \pi$, and at both the function is 0, hence $\mathsf{prog}(C^{k+1}) \leq \mathsf{prog}(C^k)$.

□

Straightening Theorem. *Fix a configuration C is described by $\{\theta_i\}_{i\in 1..n-1}$ and a straight configuration C' described by $\{\theta'_i\}_{i\in 1..n-1}$ where every angle is θ_{n-1} ($\theta'_i := \theta_{n-1}$ for $i \in 1..n$). Then $\mathsf{prog}(C') \leq \mathsf{prog}(C)$.*

Proof. For $j \in 1..n-1$, let configuration C^j be described by $\{\theta^j_i\}_{i\in 1..n-1}$ where $\theta^j_i := \theta_j$ $(i < j)$ and $\theta^j_i := \theta_i$ $(i \geq j)$. Observe $C = C^1$ and $C' = C^{n-1}$. For $j \in 1..n-2$, $\mathsf{prog}(C^{j+1}) \leq \mathsf{prog}(C^j)$ by the Single Straightening lemma, whence $\mathsf{prog}(C') = \mathsf{prog}(C^{n-1}) \leq \mathsf{prog}(C^1) = \mathsf{prog}(C)$. □

6.3.5 Applying transformations

Using the transformations described in the previous subsections we can extend the progress lower bound of $\min(d, r)$ for balanced, d-uniform, separated, straight chains described in §6.2, to a lower bound of $\min(\min_{i\in I} d_i, r)$ for arbitrary chains.

Progress Theorem for Chains. *The progress of a chain $C = \langle I, F\rangle$ is $\mathsf{prog}(C) \geq \min(\min_{i\in I} d_i, r)$.*

Proof. By the Truncation lemma we can set all the source-target distances to $d = \min(\min_{i\in I} d_i, r)$ to obtain a d-uniform chain. Using the Separation, Balancing, and Straightening lemmas there exists an angle $\theta \in [0, \pi]$ such that the straight chain $C_n(d, \theta)$ has less progress than C ($\mathsf{prog}(C) \geq \mathsf{prog}(C_n(d, \theta))$)).

Finally, by the Progress theorem for straight n-chains we have $\mathsf{prog}(C_n(d, \theta)) = \mathsf{prog}(C_2(d, \theta))$, and by the lemma of progress of 2-chains we have $\mathsf{prog}(C_2(d, \theta)) \geq d$ for any θ. Hence this proves that $\mathsf{prog}(C) \geq \mathsf{prog}(C_n(d, \theta)) = \mathsf{prog}(C_2(d, \theta)) \geq d$. □

Moreover, the bound is tight for d-uniform configurations for any service (local or global) that guarantees robust trajectories.

Optimality Theorem. *There are chains that cannot make more than* $\min(d, r)$ *progress under any service that produces robust trajectories.*

Proof. For any n, we exhibit a chain of n agents with progress exactly $\min(d, r)$. Fix n and consider the straight chain $C_n(d, 0)$, the first agent has progress $\min(d, r)$ ($\delta^n_1 = \min(d, r)$) while every other agent has no progress ($\delta^n_i = 0$ for $i > 1$), therefore $\mathsf{prog}(C_n(0, d)) = \min(d, r)$.

If a service guaranteed some progress $q > 0$ to any other agent i, then if this agent advances q units and all other agents remain still, the graph will disconnect (thus the trajectory is not robust). □

7 Termination

Consider an arbitrary chain of agents running the connectivity service. How many rounds does it take the agents to get (arbitrarily close) to their target? Let $d_i[k]$ be the source-target distance of agent i after round k, we say an agent is ε-close to its target at round k iff $d_i[k] \leq \varepsilon$. Given the initial source-target distance $d_i[0]$ of each agent, we will give an upper bound on k to guarantee every agent is ε-close.

So far we proved that while the target of every agent is outside its communication radius r, the collective distance traveled is r; moreover this is tight up to a constant factor. However, once an agent has its target within its communication radius, we can only argue that collective progress is proportional to the smallest source-target distance (since we truncate to the smallest distance). Unfortunately this is not enough to give an upper bound on k.

Let $D_k = \sum_i d_i[k]$ and $d_{\min}[k] = \min_i d_i[k]$, then $D_{k+1} \leq D_k - \min(d_{\min}[k], r)$. However if $d_{\min}[k] = 0$ this yields $D_{k+1} \leq D_k$ and we cannot prove termination. The following lemma allows us to sidestep this limitation. We call a chain *almost* d-uniform if all the inner nodes are d-uniform and the outermost nodes have source-target distance 0.

Progress Theorem for Almost-Uniform Chains. *An* almost *d-uniform chain C_n of size $n \geq 3$ has progress* $\mathsf{prog}(C_n) \geq \delta^{\angle}(d, \frac{\pi}{2}, \arccos\frac{d}{2}) \geq \gamma_0 d$ *where* $\gamma_0 := 1 - \sqrt{2-\sqrt{3}} \approx 0.48.$

Proof. Observe that the Balancing and Separation theorems still apply. Moreover, by the independence lemma and the monotonicity of the progress function we can assume the endpoints are at an angle of $\arccos\frac{d}{2}$ to their neighboring source-target vector.

Hence for $n = 3$ we need to consider one configuration, and by the target-connectedness assumption it's clear that the inner node makes full progress and hence $\mathsf{prog}(C_3) \geq d$. For $n > 3$ there is a family of possible chains determined by the angles between the inner nodes, we proceed by a complete induction on n. Observe that we can assume the progress of the internal nodes depends on both of its neighbors, since otherwise we could argue about a smaller subchain.

Base case. Let $n = 4$, clearly only the two internal nodes make progress, therefore we have $\mathsf{prog}(C_4) = \delta^{\mathsf{both}}(d, \arccos\frac{d}{2}, \alpha) + \delta^{\mathsf{both}}(d, \pi-\alpha, \arccos\frac{d}{2})$ where α is the angle between the two internal nodes. If $\alpha \leq \arccos\frac{d}{2}$ or $\pi - \alpha \leq \arccos\frac{d}{2}$, then $\mathsf{prog}(C_4) \geq d$. For $\arccos\frac{d}{2} \leq \alpha \leq \pi - \arccos\frac{d}{2}$ we define the restricted minimization problem $\alpha^* = \operatorname{argmin}_\alpha \mathsf{prog}(C_4)$. There is a unique minimum at $\alpha^* = \frac{\pi}{2}$ and hence $\mathsf{prog}(C_4) \geq 2\delta^{\angle}(d, \frac{\pi}{2}, \arccos\frac{d}{2}) \geq \gamma_0 d$.

Inductive step. Consider a chain of length $n > 4$ with $n-2$ interior nodes. Let S be the set of angles between the first $n-3$ interior nodes and let α be the angle between the last interior nodes. The progress of the chain is $\mathsf{prog}(C_n) = p(S,\alpha) + \delta^{\angle}(d, \alpha, \arccos\frac{d}{2})$, where $p(S,\alpha)$ represents the progress of the first $n-3$ interior nodes. Similarly for a chain of length $n+1$ there are $n-1$ interior nodes, and its progress is $\mathsf{prog}(C_{n+1}) = p(S,\alpha) + \delta^{\angle}(d,\alpha,\beta) + \delta^{\angle}(d, \pi-\beta, \arccos\frac{d}{2})$.

We prove the bound by cases on α. If $\alpha \leq \frac{\pi}{2}$, we can minimize the last two terms of $\mathsf{prog}(C_{n+1})$ by solving $\min_{\alpha,\beta} \delta^{\angle}(d,\alpha,\beta) + \delta^{\angle}(d,\pi-\beta,\arccos\frac{d}{2})$, which has a single minimum at $\alpha = \beta = \frac{\pi}{2}$, and thus $\mathsf{prog}(C_{n+1}) = p(S,\alpha) + \delta^{\angle}(d,\alpha,\beta) + \delta^{\angle}(d,\pi-\beta,\arccos\frac{d}{2}) \geq \delta^{\angle}(d,\frac{\pi}{2},\frac{\pi}{2}) + \delta^{\angle}(d,\frac{\pi}{2},\arccos\frac{d}{2}) \geq \gamma_0 d$.

If $\alpha > \frac{\pi}{2}$, by the inductive hypothesis we have $\mathsf{prog}(C_n) \geq \gamma_0 d$ and it suffices to show $\mathsf{prog}(C_{n+1}) \geq \mathsf{prog}(C_n)$. This is equivalent to showing $\delta^{\angle}(d,\alpha,\beta) + \delta^{\angle}(d,\pi-\beta,\arccos\frac{d}{2}) - \delta^{\angle}(a,\alpha,\arccos\frac{d}{2}) \geq 0$ for $\alpha > \frac{\pi}{2}$ and any β, which also holds. □

The progress theorem for almost-uniform chains proves that once a subset of the agents reach their target, the rest of the agents make almost the same progress as before. It seems reasonable to expect that if a subset of the agents get ε-close to their target (for small enough ε) a similar result holds. This is at the core of the termination theorem which proves an upper bound on the number of rounds needed for nodes to be ε-close to their targets.

We say the targets of two nodes are ℓ-connected if they are at distance ℓ of each other. So far we have assumed neighboring nodes have connected targets, that is they are r-connected. To prove the next theorem we require a stronger assumption, namely, that targets are $(r-2\varepsilon)$-connected for any $\varepsilon > 0$.

Termination Theorem. *If targets are $(r-2\varepsilon)$-connected, nodes get ε-close within $O(D_0/r + n^2/\varepsilon)$ rounds.*

Proof. Since targets are $(r-2\varepsilon)$-connected, we can assume each node stops at the first round when they are ε-close to their target and the resulting configuration is $(r-2\varepsilon)$-connected. Therefore we can consider the source-target distance of a node to be either greater than ε when it is not ε-close, or zero once it is ε-close.

If initially every node i is at distance $d_i \geq r$ from its target, it takes at most D_0/r rounds before there exists some node i with $d_i < r$. If there is a node i with source-target distance $d_i < r$ it follows that $D_k < r\frac{n^2}{2}$, we argue that from this point on we can assume a progress of at least $\gamma_0\varepsilon$ per round until every node reaches its target, therefore the total number of rounds is $O(D_0/r + n^2/\varepsilon)$.

Consider a chain $C = \langle I, F\rangle$ and let the subset $S_k \subseteq I$ represent the set of agents which are already at their target at round k ($i \in S_k$ iff $d_i[k] = 0$). If $S_k = I$ then we are done, otherwise there exists a subchain $C' \subseteq C$ where all agents except possibly the endpoints have $d_i[k] > \varepsilon$. Hence by the progress theorem for almost-uniform chains the progress is at least $\gamma_0\varepsilon$, which concludes the proof. □

8 Conclusion

In this paper we present a local, oblivious connectivity service (§3) that encapsulates an arbitrary motion planner and can refine any plan to preserve connectivity (the graph of agents remains connected) and ensure progress (the agents advance towards their goal). We prove the algorithm not only preserves connectivity, but also produces robust trajectories so if an arbitrary number of agents stop or slow down along their trajectories the graph will remain connected (§4).

We also prove a tight lower bound of $\min(d, r)$ on progress for d-uniform configurations (§6). The truncation lemma allows this lower bound to apply to general configurations using the minimum distance between any agent and its goal. Thus, when each agent's target is within a constant multiple of the communication radius, the lower bound implies the configuration will move at a constant speed towards the desired configuration.

As the agents get closer to their goal, this bound no longer implies constant speed convergence. We prove a bound of $O(D_0/r + n^2/\varepsilon)$ on the number of rounds until nodes are ε-close. This bound requires assuming targets are $(r - 2\varepsilon)$-connected, though we conjecture that it is possible to remove this assumption. The D_0/r term in the bound is necessary because when the initial source-target distance is large enough, clearly no service can guarantee robust, connected trajectories if agents advance faster than one communication radius per round.

It would be tempting to prove agents advance at a rate proportional to the mean (instead of the minimum) source-target distance, which would imply a termination bound of $O(D_0/r + n\log\frac{n}{\varepsilon})$. However, it is possible to construct an example which shows that the progress is less than $\gamma \cdot mean$, for any constant $\gamma > 0$. An alternative approach we intend to pursue in future work is to directly argue about the number of rounds it takes the agents to reach their target. This may give a tighter bound on the rate of convergence over quantifying the distance traveled by the agents in a single round, which necessarily assumes a worst case configuration at every step.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobilerobots with limited visibility. *Robotics and Automation, IEEE Transactions on*, 15(5):818–828, 1999.

[2] S. Carpin and L. E. Parker. Cooperative Leader Following in a Distributed Multi-Robot System. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[3] A. Cornejo and N. Lynch. Connectivity Service for Mobile Ad-Hoc Networks. *Spatial Computing Workshop*, 2008.

[4] R. Fierro and A.K. Das. A modular architecture for formation control. *Robot Motion and Control, 2002. RoMoCo '02. Proceedings of the Third International Workshop on*, pages 285–290, Nov. 2002.

[5] K.R. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 1969.

[6] A.T. Hayes and P. Dormiani-Tabatabaei. Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, 2002.

[7] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks. *Computer Communications Review - SIGCOMM*, 1996.

[8] N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. *INFOCOM*, 3:1702–1712, 2003.

[9] N. Malpani, J. Welch, and N. Vaidya. Leader election algorithms for mobile ad-hoc networks. *DIAL-M: Workshop in Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000.

[10] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. *Workshop on Mobile Computing Systems and Applications*, 1999.

[11] A. Regmi, R. Sandoval, R. Byrne, H. Tanner, and CT Abdallah. Experimental implementation of flocking algorithms in wheeled mobile robots. In *American Control Conference, 2005. Proceedings of the 2005*, pages 4917–4922, 2005.

[12] K. Savla, G. Notarstefano, and F. Bullo. Maintaining limited-range connectivity among second-order agents. *SIAM Journal on Control and Optimization*, 2007.

[13] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

[14] J. Walter, J. Welch, and N. Vaidya. A mutual exclusion algorithm for ad-hoc mobile networks. *Wireless Networks*, 2001.

[15] M. M. Zavlanos and G. J. Pappas. Controlling Connectivity of Dynamic Graphs. *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 6388–6393, 2005.