Superdiffusive Dispersion and Mixing of Swarms

JACOB BEAL, Raytheon BBN Technologies

A common swarm task is to disperse evenly through an environment from an initial tightly packed formation. Due to communication and sensing limitations, it is often necessary to execute this task with little or no communication between swarm members. Unfortunately, prior approaches based on repulsive forces or uniform random walks can often converge quite slowly. With an appropriate choice of random distribution, however, it is possible to generate optimal or near-optimal dispersion and mixing in swarms with zero communication. In particular, we discuss three extremely simple algorithms: reactive Levy walk, reactive ball dispersion, and purely reactive dispersion. All three algorithms vastly outperform prior approaches in both constrained and unconstrained environments, providing a range of options for trading off between aggressiveness and evenness in dispersion.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Spatial computing, swarm, dispersion, deployment, coverage, mixing, Levy flight, Levy walk, reactive, anomalous diffusion

ACM Reference Format:
Jacob Beal. 2015. Superdiffusive dispersion and mixing of swarms. ACM Trans. Autonom. Adapt. Syst. 10, 2, Article 10 (June 2015), 24 pages.
DOI: http://dx.doi.org/10.1145/2700322

10

1. INTRODUCTION

Dispersion is one of the basic maneuvers needed for a wide variety of swarm applications: beginning from an initially tightly packed formation, the individuals composing a swarm spread out evenly through their environment. Moreover, due to communication and sensing limitations, it is often necessary to execute this task with little or no communication between swarm members.

To be effective in many applications, swarm dispersion must balance between two contradictory goals: on the one hand, members of a swarm need to spread outward quickly in order to disperse well (aggressive dispersion). At the same time, however, it is often the case that the swarm needs to simultaneously maintain a continuous coverage over its initial deployment area (even dispersion). For example, if the swarm is being used for information gathering or for communication, then the swarm needs to start accomplishing its task quickly, yet the process of deploying the swarm should not leave any large transient gaps in covering the immediate area around the people deploying it.

Prior methods of low-communication dispersion have generally been based on simple physics models, such as repulsive forces or diffusion through uniform random walks.

Author's address: J. Beal, Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA; email: jakebeal@bbn.com.

DOI: http://dx.doi.org/10.1145/2700322

These methods, however, are unable to spread the members of a swarm quickly unless carefully tuned for the expected scale of the swarm and its environment.

One possible issue that may have contributed to this problem is a lack of sufficient definition of the problem of dispersion. In particular, there has been no general metric for comparing the efficacy of dispersion algorithms, nor analysis of upper bounds for "ideal dispersion" as a benchmark to compare against. Prior work has instead generally examined only particular outcomes of dispersion, such as the time to reach some "final" configuration (e.g., McLurkin and Smith [2004], Poduri and Sukhatme [2007], and Howard et al. [2002]), the position of swarm members at an arbitrary "snapshot" in time (e.g., Ludwig and Gini [2006] and Howard et al. [2002]), or how well an application goal is achieved (e.g., Mullins et al. [2012] and Oyekan et al. [2010]).

After a review of related work in Section 2, the article gives a mathematical definition of the problem of dispersion of swarms (Section 3). This formulation reveals that dispersion algorithms must make tradeoffs between competing goals of aggressiveness and evenness, which cannot in general both be simultaneously optimal. From this analysis, in Section 4, we derive two extreme algorithms, reactive ball dispersion and purely reactive dispersion, the first of which is maximally even and the second maximally aggressive in unconstrained environments. A third algorithm, reactive Levy walk, is derived from physical models of superdiffusion and offers tunability across a spectrum of dispersion behaviors, including both maximal evenness and maximal aggressiveness. Through a range of experiments in simulation, presented in Sections 5, 6, and 7, we find that all three algorithms vastly outperform prior approaches in both constrained and unconstrained environments while requiring minimal sensing and no communication between members of the swarm and providing a range of options for trading off between aggressiveness and evenness in dispersion. Finally, Section 8 summarizes our findings and presents directions for further investigation.

2. RELATED WORK

Much of the prior work on swarm dispersion (or coverage, which is closely related) has considered systems out of the scope of the investigation in this article due to reliance on high levels of communication, centralized planning, or the ability to leave markers such as pheromones in their shared (real or virtual) environment. The vast majority of such methods rely on some form of shared coordinate system and map (a good introductory survey may be found in Choset [2001]). If map-based coordination is centralized, then every member of the swarm must be able to communicate frequently with the central coordinator, which may not be possible for any number of reasons, including physical constraints (e.g., operation in confined spaces), lack of infrastructure (e.g., disaster response), or size of the swarm (e.g., UAVs sharing a satellite uplink). If map-based coordination is decentralized, then swarm members must exchange potentially high-resolution map data in order to maintain aligned models of the environment and the dispersal of swarm members therein. Thus, these methods often have high communication requirements, which may be problematic for a number of reasons, including difficult communication environment (e.g., operation in buildings or urban areas) or lack of available bandwidth (e.g., acoustic communication underwater). Marker-based approaches eliminate these problems by placing information directly into the actual physical environment. At present, however, it is rarely practical to make frequent marks on a physical environment: for example, marks may be too durable or too fragile, require time to place, expend a limited payload of marking agent, be unreliable to read, and so on. Virtual markers do not have these problems, but reduce back to the same problem of map alignment and exchange.

More local and self-organizing approaches typically fall into two categories of nature-inspired models. One set is primarily based on uniform random walks, either unbiased or biased (e.g., Oyekan et al. [2010], Poduri and Sukhatme [2007], and Ludwig and Gini [2006]). The other, more common strategy uses repulsive forces in a variety of combinations and models such as flocking (e.g., Mathews et al. [2012]), potential fields (e.g., Howard et al. [2002]), and gradient descent (e.g., McLurkin and Smith [2004]). There are also a number of modified or hybrid strategies, such as combining biased random walks and diffusion-limited aggregation [Mullins et al. 2012] or springs and random walks [Eckert et al. 2012]. Asymptotic analyses, however, reveal that both uniform random walks [Beal et al. 2009] and force-driven dispersion (viewed as a process of distributed consensus [Elhage and Beal 2010]) will generally perform poorly for large swarms.

The use of reactive Levy walks for swarm dispersion was introduced in Beal [2013] (a prior, less developed version of this article). Purely reactive dispersion was introduced in Beal [2013] as well, as a control comparison for reactive Levy walks, and discovered to provide highly effective dispersion in constrained environments. Levy walks and/or flights have previously been used for generating search patterns for robots or other agents a number of times [Nurzaman et al. 2010; Calitoiu 2009; Keeter et al. 2012], as well as for routing [Shin et al. 2008], though these systems considered only small numbers of agents in sparse environments, a problem with significantly different constraints and requirements than dispersion.

More generally, Levy flights and Levy walks are scale-free particle motion processes originally formulated as models in the study of chaotic physical phenomena [Mandelbrot 1983] (precise definitions will be given in Section 4.3). Although these two random processes are distinct, they are often frequently used and referred to interchangeably in the literature (this sort of imprecise thinking can lead to significant problems, as in the recent challenges to some analyses of Levy motions by animals [Edwards 2011; Benhamou 2007; Plank and Codling 2009]). Levy motions have been applied to modeling a number of physical processes, including diffusion under turbulence [Shlesinger et al. 1987], the passage of photons through hot gases [Mercadier et al. 2009], and plasma physics [AV et al. 2002]. More recently, it has been proposed that animals use Levy walks in their foraging patterns [Viswanathan et al. 1999; Schuster and Levandowsky 1996], and evidence of such behavior has been reported for an extremely wide range of organisms, from amoebas [Schuster and Levandowsky 1996] to bumblebees, deer, and albatross [Viswanathan et al. 1999], and from predatory fish, turtles, and penguins [Sims et al. 2008; Humphries et al. 2010] to spider monkeys [Ramos-Fernndez et al. 2004]. Even humans appear to evidence Levy statistics in our movements [Brockmann et al. 2006; Rhee et al. 2011].

3. FORMALIZING THE PROBLEM OF DISPERSION

Let us begin our investigation of swarm dispersion with a careful examination of the nature of the problem. Speaking informally, there are two qualitative goals that any swarm dispersion algorithm is generally expected to fulfill. First, the members of the swarm should rapidly spread out through space. Second, as the members of the swarm spread through space, they should do so in a manner that does not lead to any large gaps between swarm members. We will term these goals aggressiveness and evenness, respectively.

Despite their simplicity, it is often impossible to simultaneously maximize both aggressiveness and evenness, particularly in highly constrained environments. Intuitively, this can be understood as aggressiveness, requiring members of the swarm to move outward as quickly as possible, while evenness requires some to lag behind to fill gaps (we will demonstrate this more formally later). Depending on the particulars of the applications, different tradeoffs between aggressiveness and evenness may be desirable. For example, a swarm of UAVs used to deploy a wireless broadband network over a large wilderness area might maximize aggressiveness and be able to tolerate a large transient gap during its initial deployment. A swarm of ground robots used for tactical building clearance, on the other hand, may need to sacrifice some aggressiveness to maximize evenness in order to ensure that there are no dangerous gaps in observation.

There are many additional challenges and constraints that can also play into the dispersion problem: the maneuvering capabilities of swarm members, imprecision of movement, environmental factors such as wind or rough terrain, maintenance requirements, energy constraints, and so forth. For the purposes of this article, we will abstract these away, dealing only with a swarm of idealized particles.[1]

3.1. Formal Problem Definition

Under the assumption of idealized particles, the problem of dispersion may be formalized as follows:

—A swarm consists of a collection of n particles.
—Particles can move freely through a k-dimensional area of space A with a velocity of up to v meters per second.
—The particles of the swarm are initially arbitrarily dispersed through some initial region of space $D_0 \subseteq A$.
—Particles can coordinate by broadcasting information to neighboring particles (under some communication model) at a rate of $O(1)$ bits per second.
—Particles have a proximity sensor that can reliably detect the close presence of other particles or obstacles (e.g., via bump sensors, sonar ranging, LIDAR, positioning beacons, etc.), including the boundary of A.[2]
—Particles have no information about A other than its dimensionality k and whatever can be gathered from their motions, proximity sensor, and communication with other particles. For example, particles do not have any a priori information about the position of the boundaries of A.
—Following t seconds of movement under the control of some dispersion algorithm, the space through which the swarm has dispersed is a region $D_{t|f} \subseteq A$, where f is a fraction between zero and one and $D_{t|f}$ is the convex hull of the closest $\lceil fn \rceil$ particles to D_0.

Figure 1 shows a graphical illustration of this definition.

The reason for the somewhat complex definition of the dispersed region $D_{t|f}$ is that we need some way of mapping back from discrete positions to a continuous region of space that can cohere with our intuitive understanding of "where the swarm has dispersed through." A convex hull is a simple way of capturing such a notion, yet with many methods it can be significantly distorted by a few "lucky" particles that move a long distance, while the bulk of the particles are generally much less well dispersed. By setting f slightly lower than 1, we can exclude such outliers and focus on the bulk of the swarm. Except where otherwise noted, we will use $f = 0.95$, that is, restricting to particles at or below the 95th percentile of displacement from their initial positions.

Note also that the communication model is left extremely flexible and open to further definition. The communication model is left so open ended in order to cover as broad a

[1]It is worth noting, however, that many such concerns can be dealt with as modulations of space, per the discussion in Beal [2012], and the algorithms we discuss in the article all should adapt well to such an approach.
[2]Note that the boundary of A might be either physical (e.g., walls of a building) or virtual (e.g., an "invisible fence" on permitted operating space).

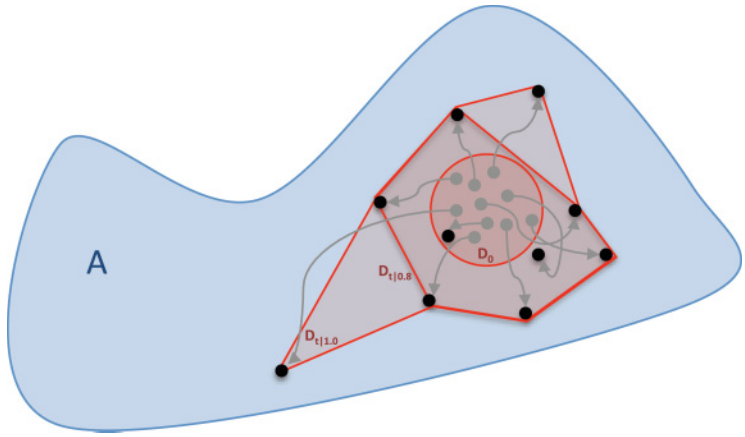


Fig. 1. Graphical illustration of the dispersion problem: a collection of n particles (here, 10) move freely through some space A at a speed of up to v meters per second. Particles are initially positioned (gray dots) in some space D_0. After t seconds of dispersion, the new positions (black dots) define the effective region of dispersion $D_{t|f}$ as the convex hull around the fraction f of particles closest to D_0. Here, two examples are shown: $D_{t|1.0}$ includes all particles, while $D_{t|0.8}$ excludes the furthest two as likely outliers. The goal is to expand $D_{t|f}$ aggressively while maintaining an even distribution of particles within $D_{t|f}$.

class of swarm algorithms as possible.[3] The main goal of this definition is simply to rule out "heavyweight" algorithms that do not scale well, such as joint planning. In fact, we shall see that with an appropriate choice of random distribution, highly effective dispersion can be achieved with no communication at all (except for whatever might be used in implementing the proximity sensor).

3.2. Aggressiveness and Evenness

Under this formalization of dispersion, the goal of aggressive dispersion is to maximize the expected area of $D_{t|f}$ at every t, and the goal of even dispersion is to ensure a uniform expected distribution of particles through $D_{t|f}$.

The first is easy to quantify: to quantify aggressiveness, we will simply measure the rate of growth of $D_{t|f}$ with respect to time. The optimally aggressive behavior for a dispersion algorithm is for $D_{t|F}$ to expand its boundaries outward at rate v. We shall thus consider as near optimal any algorithm where the diameter of $D_{t|f}$ with respect to rising t is $O(t)$.

The second is a bit more tricky: we will quantify evenness at any given point in time by examining the probability density function ρ of particles distributed over some region of space X (typically $D_{t|0.95}$ in this article). To be precise, the probability density function ρ may be defined for a specific dispersion algorithm and set of experimental conditions as a function that, when sampled, produces the same statistics of particle positions as execution of the algorithm is expected to produce under those conditions. When studying the behavior of algorithms empirically, of course, we cannot measure probability density directly, but must estimate it in various ways from the observed samples of various trial runs.

Given an estimate of such a probability density function, we will calculate a "worst case" measure of the evenness $E(X)$ of the distribution by the ratio between the lowest

[3] In fact, in many swarm scenarios, it would be entirely reasonable to allow somewhat more, for example, $O(\log n)$.

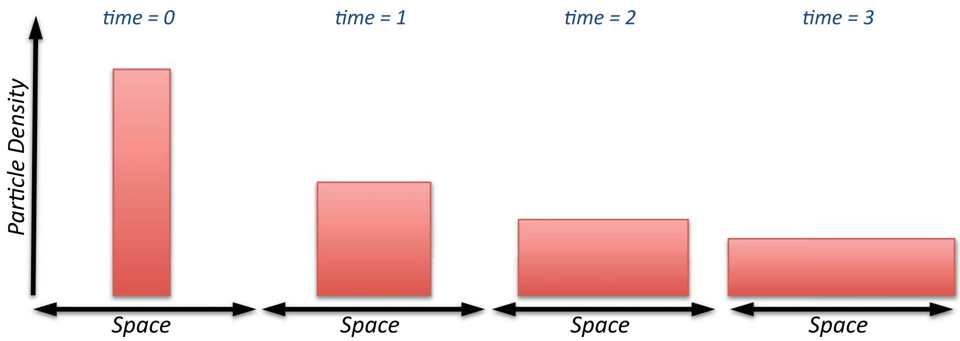


Fig. 2. Example of optimally aggressive and even dispersion in one dimension at $v = 1$ meter/second from an initial uniform distribution D_0 on a 2-meter interval. At each second, the distribution has expanded outward by 1 meter in each direction, with the density of particles decreasing proportionally but remaining uniform.

and highest density regions:

$$E(X) = \frac{\min\{\rho(x)|x \in X\}}{\max\{\rho(x)|x \in X\}}.$$

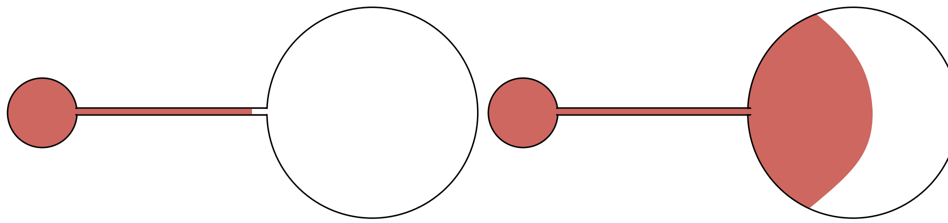
A perfectly even dispersion will thus obtain the maximum measure of $E(X) = 1$, and dispersion with a consistent failure to cover some portion of space will obtain the minimum measure of $E(X) = 0$. Using the ratio effectively normalizes the measure, so that a distribution with the same structure will receive the same rating no matter the scale of region size. We shall thus consider as near optimal any algorithm where the evenness for $E(D_{t|f})$ with respect to rising t is $O(1)$ and significantly above zero.

Note that $E(X)$ is, of course, dependent on our choice of region. For example, with a dispersion algorithm that produces high densities in the center and lower toward the edges, such as a uniform random walk, $E(D_{t|0.95})$ will always be lower than $E(D_{t|0.8})$. This is not a problem, however, so long as we choose the fraction consistently and focus on the bulk of the swarm rather than outlier particles.

3.3. Tradeoffs Between Aggressiveness and Evenness

In an unconstrained environment, it is possible for dispersion to be both optimally aggressive and optimally even. Figure 2 shows an example of such a dispersion in one dimension. Beginning with an initial uniform distribution D_0 on a 2-meter interval, the distribution expands outward in both directions at $v = 1$ meter per second. As the covered interval expands, the density of particles decreases proportionally but can remain uniform, since the required rate of mass flow is never greater than v. An equivalent construction can be made with an expanding disc in two dimensions, an expanding ball in three dimensions, and so forth.

In a constrained, environment, however, it is not generally possible for dispersion to simultaneously optimize both aggression and evenness. Instead, we must make some tradeoff between them. Figure 3 shows a simple counterexample, using a dumbbell-shaped space of two open areas connected by a very thin "pipe." If the swarm starts in one open area and fills it, then the bulk of its particles must be in that area; once it begins to spread into the other area, a large mass of particles must pass through the thin connecting space. Both the narrowness and length of the pipe pose obstacles: to keep optimal evenness, only a small fraction of particles can be passing through at any time, and even if this were violated, the particles cannot arrive until they have had time to pass along the length of the pipe. To be aggressive, particles must disperse at a



(a) (b)

Fig. 3. In constrained dispersion, it is not generally possible for a dispersion to be simultaneously optimally aggressive and optimally even. For example, (a) shows a snapshot of dispersion starting from the left side of an "uneven dumbbell" comprising two open areas connected by a very thin "pipe," with the region filled by the swarm marked in red. In this state, optimal evenness requires that the vast majority of the swarm particles are in the left area. A short time later, in (b), optimal evenness requires that the majority of the particles are in the right area, but this cannot be accomplished at an optimally aggressive rate because too many particles were too far away in the left side at time (a).

lower density in the new area; to be even, the new area must fill more slowly. Moreover, by varying the sizes of the open areas and the pipe, we can force an arbitrarily bad tradeoff. Problematic proportioning such as this is likely to be quite common in built environments: consider, for example, dispersing a swarm through a pair of buildings connected by a skywalk or through a city center where tightly built-up areas connect to plazas or parks.

Thus, we see that there can be no "one size fits all" solution to the problem of dispersion in general. Different applications will need to make different tradeoffs in aggressiveness versus evenness. Thus, in the next section, we introduce a set of algorithms that allow such a tradeoff to be made.

4. REACTIVE FAST DISPERSION ALGORITHMS

Prior methods of dispersion are nowhere near the theoretical bounds established in the prior section. Both uniform random walks and force-driven dispersion are expected to perform well on evenness, as driven by an expected normal distribution of particles. On aggressiveness, however, they are far from optimal with the diameter of the dispersed region expected to grow at a rate of no more than $O(\sqrt{t})$.

We thus introduce a set of three new algorithms for fast dispersion, covering a spectrum of tradeoffs between maximal aggressiveness and maximal evenness. The first, reactive ball dispersion, is maximally even but does not take full advantage of particle mobility. The second, purely reactive dispersion, is maximally aggressive but suffers from high transient unevenness. Finally, reactive Levy walk is a physics-based model that offers a tunable tradeoff between aggressiveness, evenness, and centrality of distribution, reducing to purely reactive dispersion at one extreme and uniform random walks at the other.

4.1. Maximizing Evenness: Reactive Ball Dispersion

If we vary the speed at which swarm members move so that some may move much more slowly than others, then an optimally even and aggressive dispersion of a swarm in unconstrained space can be produced by an extremely simple mechanism. Pseudocode for this algorithm, which we will term *reactive ball dispersion*, is given in Algorithm 1.

With this algorithm, each particle essentially selects a polar coordinate position uniformly distributed over a sphere of dimension k by choosing a random direction $\hat{i}$ and a fractional radius α. Particles then move at rate αv in their chosen direction,

ALGORITHM 1: Reactive Ball Dispersion

repeat
  // Choose a Random Motion
  $\hat{i}$:= **random-direction()**
  $\alpha := U[0,1]^{\frac{1}{k}}$ // kth root of unit uniform random number
  // Move until proximity sensor triggered
  repeat every Δt **seconds**
    move$(\alpha v \hat{i})$
  until proximity-sensor()

thereby uniformly expanding the sphere outward at a collective rate v (quantized in steps of Δt in this implementation).[4]

This algorithm also includes a reactive term, which aborts and picks a new random motion whenever the proximity sensor is triggered. This is important because it allows the algorithm to cope with constrained environments—including that of the interior of a packed initial distribution of particles. When a swarm is dispersing from an initially tight distribution, most of its particles are initially operating in a highly constrained environment, where each member's movements are obstructed by the other nearby members of the swarm. Without a reactivity term, particles moving counter to one another may enter a mutually blocking configuration and become stuck, and likewise when a particle encounters an obstacle in the environment.

With the inclusion of reactivity, we may predict that this algorithm (and the other reactive fast dispersion algorithms we present) will produce different behaviors on the edges and in the interior of a swarm (Figure 4). On the edges, particles are not significantly constrained by proximity and are able to move superdiffusively, rapidly dispersing the swarm. In the interior, where particles are constantly in close proximity to one another, the reactive Levy walk reduces to a constrained random walk. Particles in this region move at best diffusively, effectively marking time while they wait for the dispersion of the edges to allow them to move more freely.

This algorithm is guaranteed to produce an optimally even dispersion that expands with optimal aggressiveness through unconstrained space. Because some particles move arbitrarily slowly, however, this algorithm is not taking advantage of the full mobility of the particles and can be expected to perform less well in constrained spaces.

4.2. Maximizing Aggressiveness: Purely Reactive Dispersion

If reactive ball dispersion does not take full advantage of the mobility of swarm particles, that can be remedied easily enough. The *purely reactive dispersion* pseudocode in Algorithm 2 is identical except that the rate term α modulating particle velocity is dropped.

This algorithm is thus expected to produce a maximally aggressive dispersion in open space, in which every particle moves away from its initial location at maximum speed v (once it is no longer constrained by other swarm particles). The disadvantage is that the distribution is also maximally uneven, leaving an empty region with no particles in the center of the dispersed region. In a constrained space, reactivity will eventually randomize the paths of the particles such that this gap is filled, and on highly constrained spaces where reactivity dominates, this algorithm performs the best of the three that we present.

[4]Note that the infinitesimal chance of a particle choosing $\alpha = 0$ and therefore not moving is acceptable: a particle's current location is just as valid a position to cover as anywhere else.

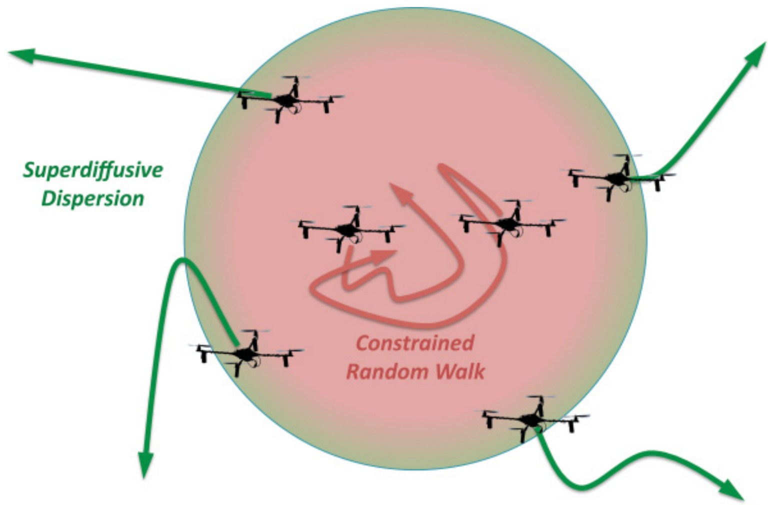


Fig. 4. Particles in a swarm that begins tightly packed will transition between two distinct phases in executing a reactive fast dispersion algorithm. Particles in the interior of the tightly packed portion of the swarm are highly constrained by one another and will move slowly (or possibly not at all) following a constrained uniform random walk. Particles at the "surface" of the packed region, on the other hand, are largely unconstrained and will disperse rapidly away from the packed region, enabling the particles they had constrained to begin rapidly dispersing as well.

ALGORITHM 2: Purely Reactive Dispersion

repeat
  // Choose a Random Motion
  $\hat{i}$:= **random-direction()**
  // Move until proximity sensor triggered
  repeat every Δt **seconds**
    move$(v\hat{i})$
  until proximity-sensor()

4.3. Trading Off Aggression and Evenness: Reactive Levy Walk

Reactive ball dispersion and purely reactive dispersion both have shortcomings: reactive ball dispersion does not take full advantage of particle mobility, and purely reactive dispersion is highly uneven in its initial distribution. Our third algorithm, *reactive Levy walk*, addresses both of these shortcomings and provides a tunable tradeoff between aggressiveness, evenness, and centrality of distribution.

A Levy flight [Mandelbrot 1983] is a random movement process similar to a random walk: a particle makes a sequence of moves, where each move is in a random direction. Unlike a random walk, however, where the lengths of the moves are identical, Levy flight moves have a random length generated from a (usually heavy-tailed) probability distribution, such that the probability of moving an integral distance of d is:

$$p(d = l) \propto l^{-k}. \tag{1}$$

The exponent k gives the fractal dimension of the area visited by the process [Shlesinger et al. 1993]. When $k \geq 1$, each move of a Levy flight moves an unbounded expected distance, modeling a scale-free *superdiffusive* motion of particles—that is, where particles have an expected displacement over time much further than predicted by uniform random-walk models of diffusion.

Levy walks are Levy flights where the particle moves at a constant velocity. This maintains the scale-free property of the distribution while restricting to more physically realizable motions. This model (and its generalization to coupled continuous-time random walks [Becker-Kern et al. 2004; Kotulski 1995]) is used to model a number of natural phenomena, as described in Section 2.

A reactive Levy walk is then identical to an ordinary Levy walk, except that it adds a reactivity term for handling constrained environments, much the same as the prior two algorithms: whenever the proximity sensor is triggered, the current move is aborted and a new random move is started. Pseudocode for a reactive Levy walk is given in Algorithm 3.

ALGORITHM 3: Reactive Levy Walk

```
repeat
    // Choose a Random Motion
    î := random-direction()
    α := U[0, 1]^k // kth power of unit uniform random number
    accumulator := 0
    // Move until threshold is reached or proximity sensor triggered
    repeat every Δt seconds
        move(vî)
        accumulator := accumulator + α · Δt
    until accumulator ≥ T or proximity-sensor()
```

In addition to adding the reactivity term, this formulation also specifies the scale-free distribution using a variable-rate accumulator rather than the standard probability computation given in Equation (1). A particle executing a reactive Levy walk begins by selecting a random direction $\hat{\imath}$, selecting a random rate of accumulation α as the kth power of a uniformly random number between 0 and 1, and setting its accumulator to zero. While the accumulator value rises at a rate of α per second (quantized in steps of Δt in this implementation), the particle moves in direction $\hat{\imath}$ at velocity v. The particle resets its accumulator and selects a new direction and rate whenever either its accumulator reaches some fixed threshold T (taking α^{-1} seconds) or its proximity sensor is triggered.[5]

Neglecting the reactivity term, this inverted formulation of Levy walking has equivalent statistics to a standard Levy walk but can be implemented with a simple parallel circuit, as shown in Figure 5, that can be easily realized using either digital or analog electronics or a neural or biochemical network.

As with standard Levy walks, the behavior of a reactive Levy walk can be modulated by adjusting the k term, which corresponds to the fractal dimension expected to be covered by each particle's trajectory.[6] Applied to swarm dispersion, this allows us to tune the algorithm for different tradeoffs between aggressiveness and evenness. Choosing k equal to the dimension of the overall space A is expected to produce the best

[5]Note that in the infinitesimal chance that a particle chooses $\alpha = 0$, it will never change direction except reactively. This is, however, an acceptable behavior, indistinguishable from the much greater likelihood of choosing an α that happens to have a timeout longer than the dispersion process will be run for.

[6]Changing the other parameters affects behavior only by a constant factor, so long as the algorithm parameters are nontrivial ($v > 0$ and $T > 0$).

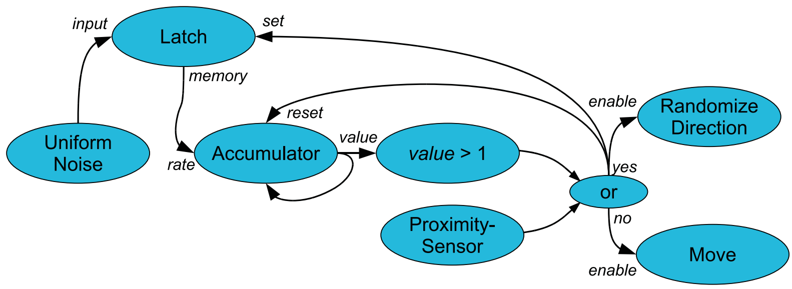


Fig. 5. Reactive Levy walk implemented as a parallel circuit that can be easily implemented as an electronic, neural, or biochemical network.

evenness that this algorithm can offer. Increasing k will increase the aggressiveness of the algorithm, decreasing the proportion of particles in the center of the distribution, until at the limit of $k = \infty$, reactive Levy walk reduces to purely reactive dispersion. In the opposite direction, decreasing k will produce more centrally weighted distributions, until at the limit of $k = 0$, reactive Levy walk reduces to a uniform random walk.

5. ASYMPTOTIC (UNCONSTRAINED) DISPERSION

In comparing the new reactive fast dispersion algorithms against prior methods of dispersion, let us begin by considering the asymptotic case, where the swarm is dispersing through a very large environment with little constraint. In this case, the reactivity term plays little role, and we can approximate behavior with fast dispersion in which the proximity sensor is never triggered, since the likelihood of particles coming in proximity of one another drops rapidly as they disperse.

5.1. Comparison of Dispersion Algorithms

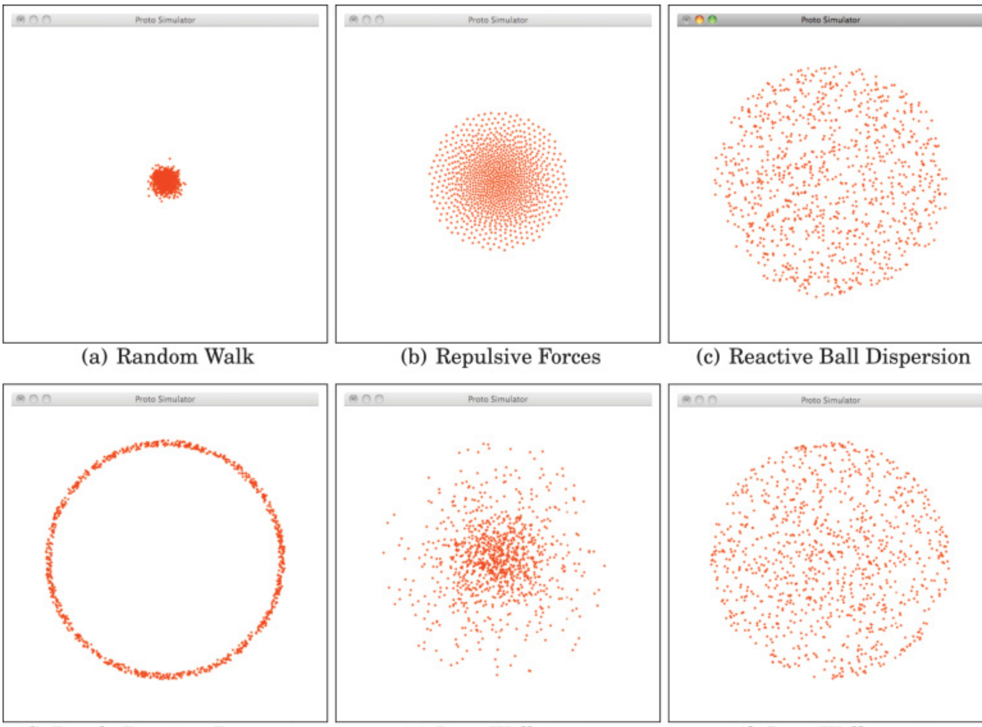
In particular, we investigate dispersion in two dimensions, though the results in two dimensions are expected to generalize to one-dimensional, three-dimensional, or even higher-dimensional spaces. Thus, for reactive ball dispersion and reactive Levy walk, we choose the appropriate dimensionality parameter $k = 2$. For comparison, we also test an underdimensioned reactive Levy walk with $k = 1$ (we investigate the effect of k on reactive Levy walks more thoroughly in Section 5.2).

For comparison with prior methods, we use two representative algorithms, uniform random walk and repulsive forces. We implement these as follows:

—Random walk moves at v meters per second in a random direction, selecting a new random direction every Δt seconds.
—Repulsive forces applies a force inversely proportional to the distance d_{ij} from each particle i to its neighbors $j \in nbrs(i)$, where the neighbors are the set of particles up to r meters away:

$$v(i) = F \sum_{j \in nbrs(i)} d_{ij}^{-1}. \tag{2}$$

Note that although the combination of forces and exponents for various repulsive force approaches in the literature varies, the asymptotic behavior will typically be similar, as the convergence time is typically driven more by the delays and decay of influence from neighborhood to neighborhood, rather than by the structure of interactions within an individual neighborhood.



(a) Random Walk (b) Repulsive Forces (c) Reactive Ball Dispersion

(d) Purely Reactive Dispersion (e) Levy Walk $k = 1$ (f) Levy Walk $k = 2$

Fig. 6. Snapshots of swarms of 1,000 particles after $t = 200$ seconds of dispersion from the center of a 500×500-meter region via random walk (a), repulsive forces (b), reactive ball dispersion (c), purely reactive dispersion (d), and reactive Levy walk for $k = 1$ (e) and $k = 2$ (f).

To empirically compare the efficacy of dispersion algorithms with the predictions in Section 4, we consider the trajectories of swarms of 1,000 particles dispersing in open space in two dimensions.

Figure 6 shows examples of these six methods operating with $v = 1$, $\Delta t = 1$, $T = 1$, and repulsive force parameters $F = 0.05$ and $r = 10$, after dispersing for $t = 200$ seconds from an initial uniform random distribution across a 10-meter square. Even after such a short period of time, the contrast between the various approaches is stark: random walk has barely begun to disperse (Figure 6(a)), while purely reactive dispersion (Figure 6(d)) has moved outward the maximum distance possible, leaving the center entirely empty. The properly dimensioned reactive ball dispersion (Figure 6(c)) and reactive Levy walk ($k = 2$, Figure 6(f)) have the same extent but fill space generally evenly—reactive Levy walk has a slightly denser band of particles at its outer boundary. The underdimensioned $k = 1$ Levy walk (Figure 6(e)) has spread particles through the same area, but they are much more densely clustered in the center than at the edges. Finally, at this early moment, repulsive forces (Figure 6(b)) appears to come close to matching the even dispersion of reactive ball dispersion and reactive Levy walk, but in fact it is already rapidly decelerating its dispersion as repulsive forces decrease on the particles at the edge of the swarm.

For a more systematic comparison, we consider the trajectories of 10 trials of each method on a swarm of 1,000 particles dispersing for 10,000 seconds with the same parameters and initial distribution, with the positions of the swarm members recorded every 100 simulated seconds. All other parameters are the same as before.

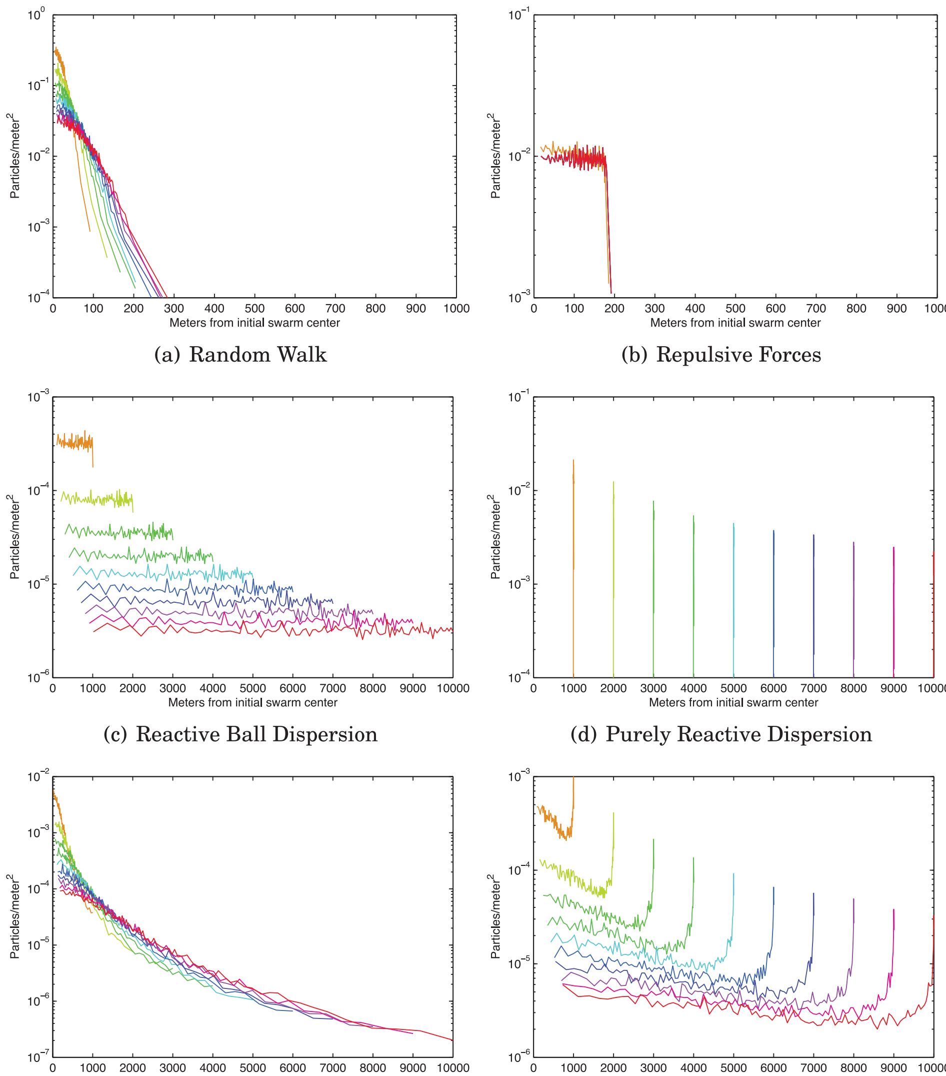
In evaluating the results of these simulations, we exploit the radial symmetry of all of the algorithms and consider only the absolute displacement of each particle from the initial center of the swarm. We then estimate the radius of $D_{t|f}$ for each percentile, that is, for f ranging from 0.01 to 1.00 in steps of 0.01. From this set of radii, we can compute an approximate probability density function by dividing the number of particles in each percentile by the difference in area between $D_{t|f}$ and $D_{t|f-1}$, and compute evenness by comparing the densities of different percentiles. Note that since we are working with random processes, this approximate calculation is affected by our choice of granularity, particularly with regard to computing evenness. As noted with the choice of f in Section 3.2, this does not impair our ability to compare algorithms and evaluate their asymptotic properties. Note also that this approximation yields a very small value rather than the true value of zero for the interior of purely reactive dispersion, which we shall ignore.

Figure 7 illustrates the evolution of distributions for each of the six methods, sampled every $1,000$ seconds. The results are, in general, as expected from the analysis in Section 4 and the snapshots in Figure 6. Random walk has a slowly expanding normal distribution, while repulsive forces comes to an even distribution and stops. Reactive ball dispersion maintains a close approximation of an optimally even distribution that expands at an optimally aggressive rate. Purely reactive dispersion moves an impulse distribution outward at an optimally aggressive rate. Finally, the $k = 2$ reactive Levy walk mixes a nearly even distribution with an impulse at the edge similar to purely reactive, while underdimensioned $k = 1$ reactive Levy walk blends a central peak with aggressive expansion at a slowly decreasing relative density.

These trends are quantified over the whole dataset in Figure 8 and Figure 9. Figure 8 compares the aggressiveness of the six methods, as well as an optimally aggressive and even dispersion, by measuring the displacement of the 95th and 10th percentiles over time. The trends are as expected from the sample distributions in Figure 7: the displacement of the 95th percentile from the center of the initial distribution rises at a near-optimal linear rate for all three fast dispersion algorithms with appropriate dimension. Both purely reactive dispersion and reactive Levy walk with $k = 2$ are slightly "faster than optimal" due to the high number of particles at the outer edge of the distribution shifting the location of the 95th percentile closer to the edge, while reactive ball distribution is almost precisely optimal. Random walk disperses slowly, at a rate proportional to $\sqrt{t}$ (following the well-known mathematical result). The expected bound for repulsive forces may be derived from the same result: repulsive forces may be considered as a process of diffusion of potential energy, and diffusion is the continuous analog of random walk. Thus, we may expect repulsive forces to initially also follow a $\sqrt{t}$ trajectory (which it does); once particles reach distance r from one another, however, they lose contact and will cease dispersing.

Figure 9 shows the evenness metric computed by comparing the density estimates for each percentile. Note that expanding the estimation window to reduce variability causes the evenness metric to rise for all methods but does not affect the conclusions drawn from the comparison. As can be seen, appropriately dimensioned reactive ball dispersion and reactive Levy walk are both near optimal, providing a consistent level of evenness, as do uniform random walk and repulsive forces. The underdimensioned reactive Levy walk with $k = 1$, however, slowly degrades in evenness, and purely reactive dispersion is of course completely uneven.

Thus, we see that for dispersion in large open regions, appropriately dimensioned reactive ball dispersion and reactive Levy walk both operate with near-optimal aggressiveness and near-optimal evenness. By contrast, other methods produce either much slower or highly uneven dispersion.



(a) Random Walk (b) Repulsive Forces

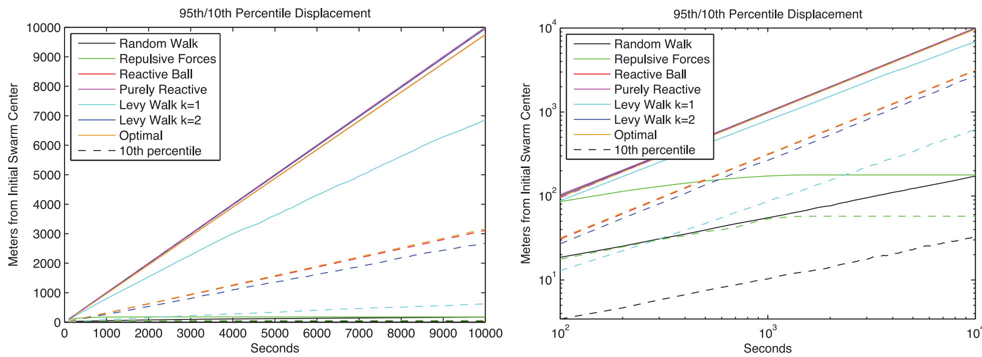
(c) Reactive Ball Dispersion (d) Purely Reactive Dispersion

(e) Levy Walk $k = 1$ (f) Levy Walk $k = 2$

Fig. 7. Evolution of distributions of 1,000 particles dispersing from the center of a region via random walk (a), repulsive forces (b), reactive ball dispersion (c), purely reactive dispersion (d), and Levy walk for $k = 1$ (e) and $k = 2$ (f). Colors indicate time, ranging linearly from $t = 1{,}000$ (orange) to $t = 10{,}000$ seconds (red). Note that random walk and repulsive forces are shown with a different x-axes, because they disperse so slowly relative to the rest. The others all disperse with optimal or near-optimal aggressiveness but vary wildly in evenness.

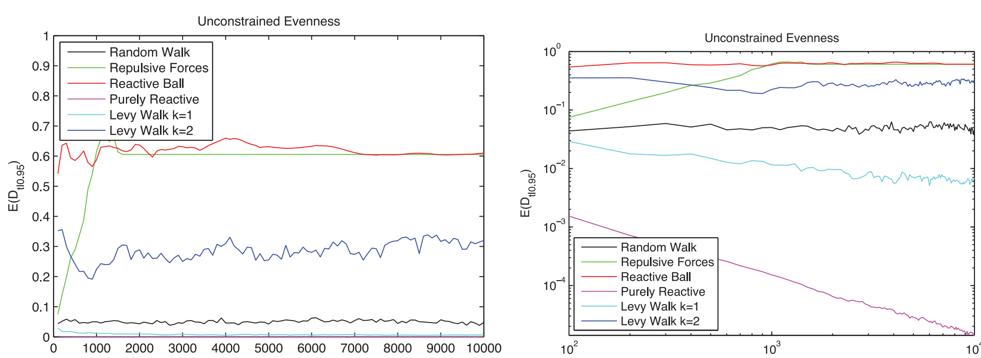
5.2. Tuning Levy Walk Dimensionality

From the analysis in Section 4 and the two examples in the previous section, we know that the dimensionality constant k in reactive Levy walks should allow this algorithm to be incrementally tuned to blend aggressiveness, evenness, and centrality. We now study this blending more closely by running reactive Levy walks varying k linearly from 0.0 to 5.0 in steps of 0.1. For each condition, we consider 10 trials on a swarm of



(a) Linear Scale (b) Log Scale

Fig. 8. The 95th percentile of particle displacement rises at a near-optimal linear rate for all three fast dispersion algorithms when their dimension k matches the dimension of the space. Both purely reactive dispersion and reactive Levy walk with $k = 2$ are "faster than optimal" due to the high number of particles at the outer edge in these distributions, while reactive ball dispersion is almost precisely optimal. The dimension-mismatched Levy walk with $k = 1$ disperses slightly more slowly than linear. Repulsive forces and random walk both disperse at a rate of $O(\sqrt{t})$, though repulsive forces rapidly slows to a stop as neighbor forces decrease. The 10th percentile lines follow a similar trend, except for purely reactive dispersion, which is not visible because it precisely overlaps its 95th percentile line, indicating the empty center of the distribution.

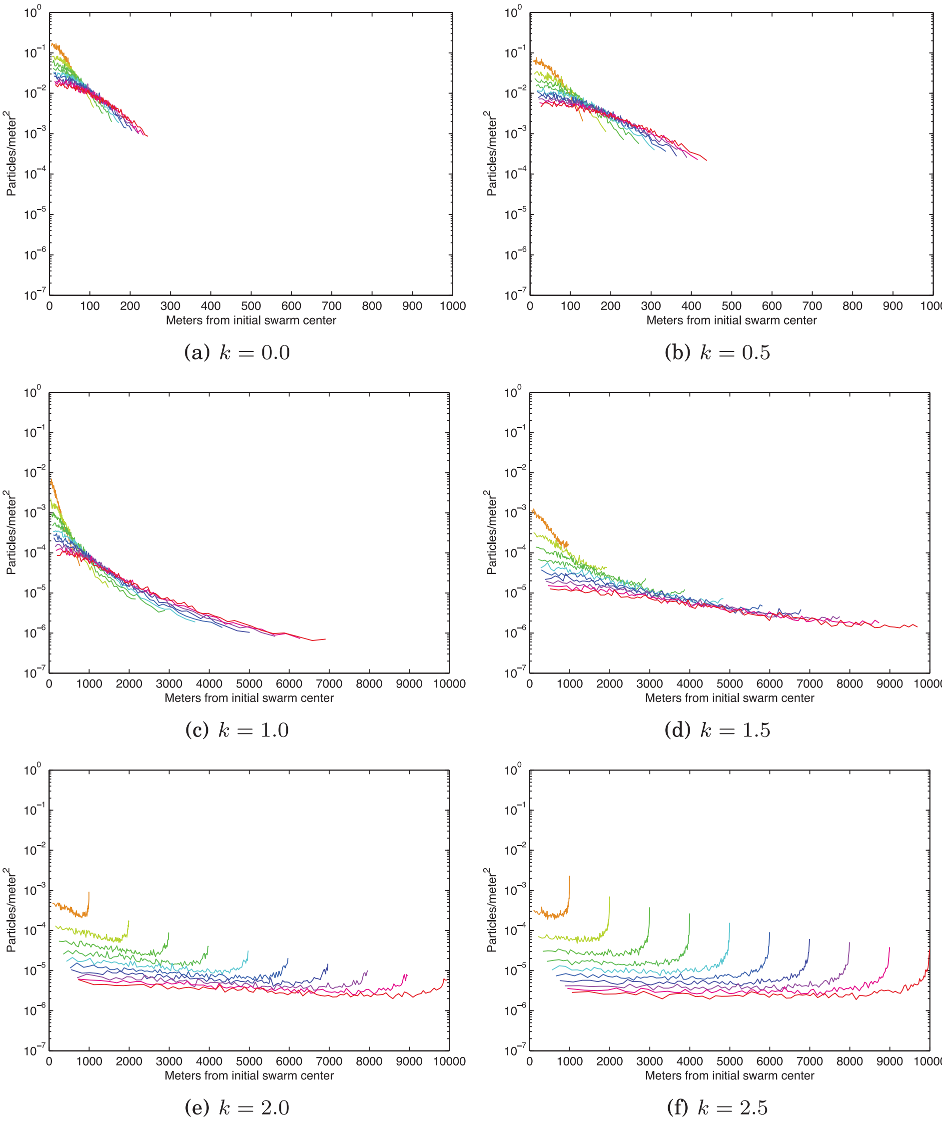


(a) Linear Scale (b) Log Scale

Fig. 9. All dispersion methods provide consistent near-optimal evenness except for the underdimensioned reactive Levy walk with $k = 1$ (which has an increasing relative density at the center) and purely reactive dispersion, which has a completely empty center and is above zero only by virtue of the numerical approximations in our density estimation.

1,000 particles dispersing for 10,000 seconds with the same parameters ($v = 1$, $\Delta t = 1$, and $T = 1$) and initial distribution are before. Also as before, the positions of the swarm members are recorded every 100 simulated seconds, and radial statistics are computed for aggressiveness and evenness.

Figure 10 illustrates the evolution of distributions at various values for k, sampled every 1,000 seconds and excluding the farthest 5% of particles as possible outliers. For two-dimensional dispersion, at $k = 0$, the distribution is identical to uniform random walk. As k rises, the distribution stretches out more aggressively, until it becomes heavy tailed and approaches optimal aggressiveness by around $k = 1$ (though of course we know from the prior experiment that it is not quite optimally aggressive). From these, the distribution continues to flatten out, losing its central bulge, until by $k = 2$



(a) $k = 0.0$ (b) $k = 0.5$

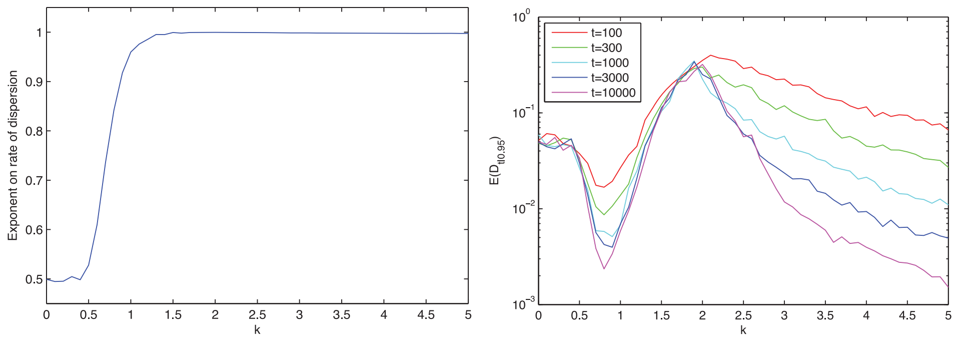
(c) $k = 1.0$ (d) $k = 1.5$

(e) $k = 2.0$ (f) $k = 2.5$

Fig. 10. Reactive Levy walks exhibit radically different behaviors depending on their value of k: at low k (a), they behave as uniform random walks. At k equal to the spatial dimension (e), they provide a near-optimal even dispersion. At high k, they approximate purely reactive dispersion, and in between (b, c, d, f), they provide tunable blends of these behaviors.

the distribution is quite even, with the central moment almost (but not quite) entirely gone and an "impulse" of maximally aggressive particles gathering at the outer edge, similar to those in purely reactive dispersion. As k continues to rise, this impulse continues to grow, with the distribution rapidly coming to approximate purely reactive dispersion.

Figure 11 summarizes the observed aggressiveness and evenness for the experiment as a whole. Evenness is calculated as before, and Figure 11(b) shows several time points to illustrate how it evolves in uneven distributions. In general, evenness is high when



(a) Aggressiveness (b) Evenness

Fig. 11. In two dimensions, reactive Levy walks are near-optimally aggressive for k above around 1.5 but only provide even dispersion when k is near 0 or 2.

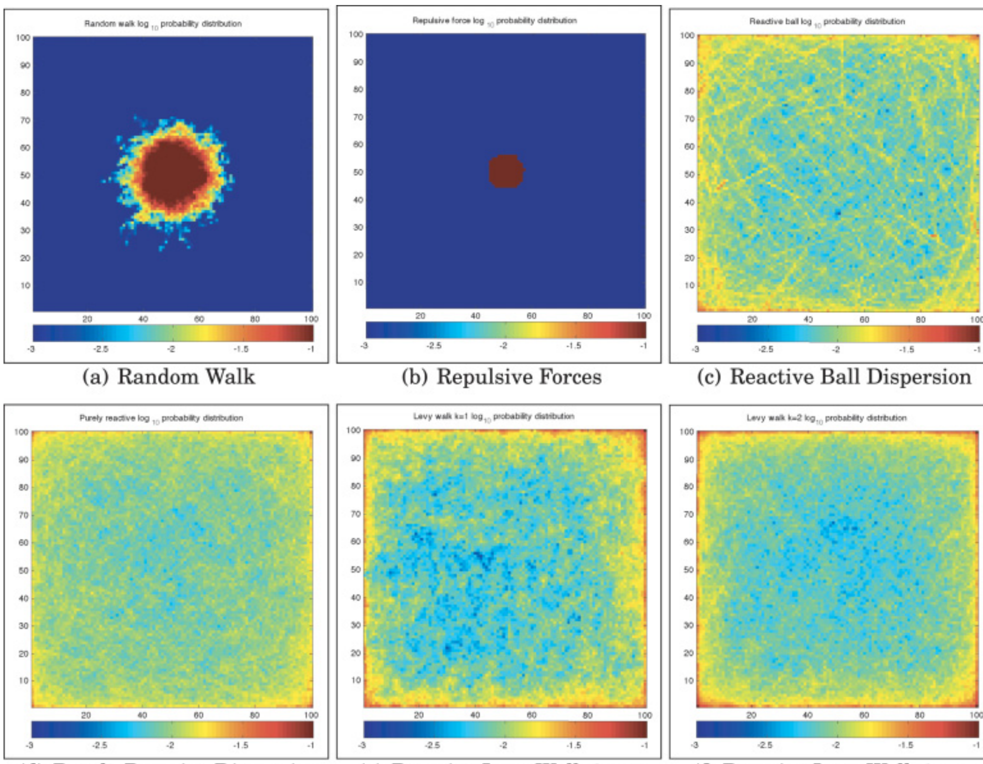
the distribution is similar to uniform random, fails at the middle of the transition to even dispersion, and fails again for high k, when the distribution is similar to purely reactive dispersion. The aggressiveness values in Figure 11(a) give the exponent on the growth of the radius of $D_{t|0.95}$ with respect to t (computed as a linear least squares fit of $\log D_{t|0.95}$ vs. $\log t$). Initially, $O(\sqrt{t})$, as expected for a uniform random walk, rapidly rises starting at around $k = 0.05$ to provide near-optimal aggressiveness in dispersion by around $k = 1.5$.

We thus see that in two dimensions, reactive Levy walks provide both aggressive and even distributions when $k = 2$; this "sweet spot" is likely due to the match between the fractal and real dimension, and so for other dimension spaces, the "preferred" k is expected to shift to match the dimension of the space. From this "preferred" point, k can be adjusted to mix either with a central distribution or increased aggressiveness, providing a range of tradeoffs that can be used to tune the algorithm for different applications. For example, a partially central distribution may be desirable for applications with greater sensitivity near the origin, such as convoy protection, while increased aggressiveness may be useful in highly constrained environments such as urban areas.

6. CONSTRAINED DISPERSION AND REACTIVITY

In a constrained space, the reactivity in reactive fast dispersion comes into play, causing the particles of the swarm to restart their random motions whenever they encounter an obstacle. This has a large effect on the patterns of dispersion produced by an algorithm, particularly with regard to evenness. We should expect that the more constrained the space is, the more that reactivity will dominate, and the less that algorithm elements aimed at creating evenness will matter. Instead, evenness in highly constrained environments will be produced primarily by the frequent restarts of random motion originating in new locations.

To evaluate the performance of our algorithms, we consider two constrained environments, one where the amount of constraint is low, the other where it is high. The low-constraint environment is an open box 1,000 meters on a side. The high-constraint environment is a 100-by-100-meter maze-like environment with a large central "room" and narrow "corridors" leading outward and around it (Figure 14). For both experiments, we use a swarm of 100 particles, each particle in the swarm is a 1-meter cube, and the movements and physical interactions of particles with one another and the maze are simulated in Proto [Beal and Bachrach 2006; MIT Proto 2012] using the ODE [Russel Smith et al. 2010] Newtonian physics engine. The swarm begins packed as tightly as possible, in physical contact in a 10-meter square in the center of the



(a) Random Walk (b) Repulsive Forces (c) Reactive Ball Dispersion

(d) Purely Reactive Dispersion (e) Reactive Levy Walk $k = 1$ (f) Reactive Levy Walk $k = 2$

Fig. 12. Mean density of particles from $t = 5{,}000$ to $t = 10{,}000$ seconds of dispersion of a swarm of 100 particles from the center of a $1{,}000 \times 1{,}000$-meter box via random walk (a), repulsive forces (b), reactive ball dispersion (c), purely reactive dispersion (d), and reactive Levy walk for $k = 1$ (e) and $k = 2$ (f). Warmer colors indicate logarithmically greater density.

environment. Dispersion is then run for 10,000 seconds using the same methods and parameters as before, 10 trials per method, with the positions of the swarm members recorded every 10 simulated seconds.

6.1. Low-Constraint Dispersion

The results of the low-constraint simulation are presented in Figure 12, where each pixel shows the mean density of particles in a 10-by-10-meter square over the second half of the simulation, from $t = 5{,}000$ to $t = 10{,}000$, for all runs. As in the asymptotic case, random walk and repulsive forces perform poorly, never even coming close to reaching the edges of the environment. All of the reactive fast dispersion methods perform fairly well, though in all cases there is a tendency for the random motions of particles to keep them closer to the walls than in the center.

Figure 13 shows the evenness values computed by comparing the densities quantized in 10-by-10-meter squares. In computing evenness, we exclude the set of squares immediately adjacent to the edges to avoid boundary effects and the fact that particles occasionally get "stuck" in a corner in this simulation. Reactivity dramatically transforms the relative ordering of the algorithms: rather than being least even, purely reactive dispersion is now the most even, thanks to the fact that particles have been able to decorrelate their trajectories through random bounces off of the walls. The $k = 2$ reactive Levy walk and reactive ball dispersion perform similarly but somewhat less

	Open Box	Maze
Random Walk	0	0
Repulsive Forces	0	0
Reactive Ball Dispersion	0.062	0.067
Purely Reactive Dispersion	0.176	0.055
Reactive Levy Walk $k = 1$	0.049	0.028
Reactive Levy Walk $k = 2$	0.073	0.042

Fig. 13. Evenness of dispersion in constrained conditions.

well, and the underdimensioned $k = 1$ reactive Levy walk is a nearby but unsurprising last.

The one significant surprise here is the relatively poor performance of reactive ball dispersion, which is, after all, optimally even in unconstrained space. Careful examination of the density plot for reactive ball dispersion gives a hint as to why this is: there are strong individual "streaks" of higher probability, likely caused by individual particles that are moving particularly slowly. Moreover, the proportion of slow-moving particles may be expected to be higher in constrained space than unconstrained, since faster particles encounter walls and rechoose their speed more frequently than slower ones. Random variations in coverage are thus therefore more likely to persist longer with this algorithm than with purely reactive dispersion and reactive Levy walks.

6.2. High-Constraint Dispersion

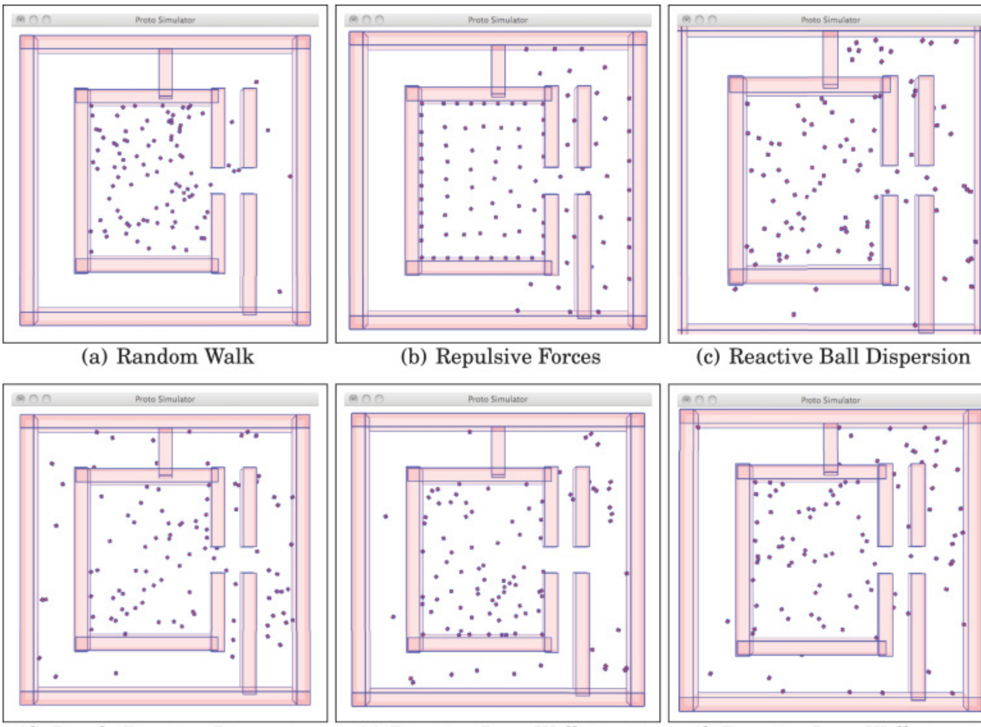
The results of the high-constraint "maze" simulation are presented in Figure 14, which shows example snapshots of swarms after $t = 2{,}000$ seconds of dispersion, and Figure 15, which shows the mean density of particles per square meter over the second half of the simulation, from $t = 5{,}000$ to $t = 10{,}000$. As in the asymptotic case, random walk performs poorly, spreading slowly enough that it never actually reaches all parts of the maze (though it does better than the low-constraint case simply because the space is smaller). Repulsive forces perform even worse, quickly reaching a stable equilibrium where particles on the edges no longer move outward, even though many others in the swarm are under significant stress. Once again, all of the reactive fast dispersion algorithms effectively disperse particles through the entire maze, with turn-on-contact slightly outperforming reactive Levy walks.

Figure 13 shows the evenness values computed by comparing the densities quantized in 1-meter squares. To compute evenness in this environment, we exclude the walls of the maze and the edges. As before, we also exclude the immediate boundary squares to avoid boundary effects and occasional stuck particles. In this highly constrained environment, reactivity clearly dominates and the differences between the reactive fast dispersion algorithms are nearly eliminated, except for the underdimensioned $k = 1$ reactive Levy walk, which is hindered by the centrality of its distribution.

We thus see that, when properly dimensioned, both reactive ball dispersion and reactive Levy walks produce aggressive and even dispersion in both highly constrained and unconstrained environments. Purely reactive dispersion, while extremely uneven in unconstrained environments or during initial deployment in a low-constraint environment, performs well in a highly constrained environment where particle motions become rapidly decorrelated through reactivity.

7. SWARM MIXING

Mixing of a swarm is a similar maneuver to dispersion, but starting with the swarm already distributed over space rather than tightly packed. In our problem formulation, this means that D_0 is on the same scale as A, and the growth of $D_{t|f}$ is not generally



(a) Random Walk (b) Repulsive Forces (c) Reactive Ball Dispersion

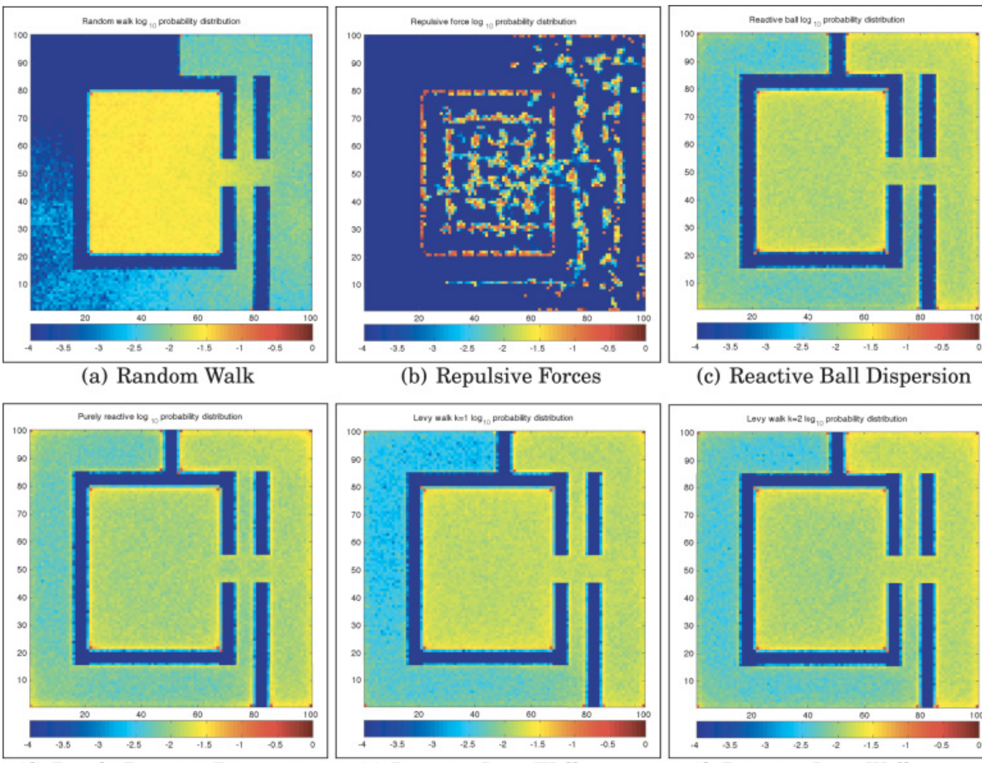
(d) Purely Reactive Dispersion (e) Reactive Levy Walk $k = 1$ (f) Reactive Levy Walk $k = 2$

Fig. 14. Snapshots of swarms of 100 particles after $t = 2{,}000$ seconds of dispersion from the center of a 100×100-meter “maze” of barriers via random walk (a), repulsive forces (b), reactive ball dispersion (c), purely reactive dispersion (d), and reactive Levy walk for $k = 1$ (e) and $k = 2$ (f).

of interest. Uses include data ferrying, blending of mission and maintenance tasks, and enhancing robustness through diversity. Dispersion algorithms should generally work well for mixing also, since both dispersion and mixing require particles to move relatively long distances with low correlation in their movements. Based on the results in prior sections, we should thus expect the efficacy of mixing to be good for the reactive fast dispersion methods but poor for random walk (where particles traverse space more slowly) and for repulsive forces (where there is high correlation in movement).

For an empirical test, we consider the same algorithms and parameters as before, but now consider a swarm of 300 particles, beginning distributed uniformly randomly through a 1,000-by-1,000-meter space with no internal obstacles. Dispersion is then run for 5,000 seconds using the same methods and parameters as before, 10 trials per method, with the positions of the swarm members recorded every 10 simulated seconds.

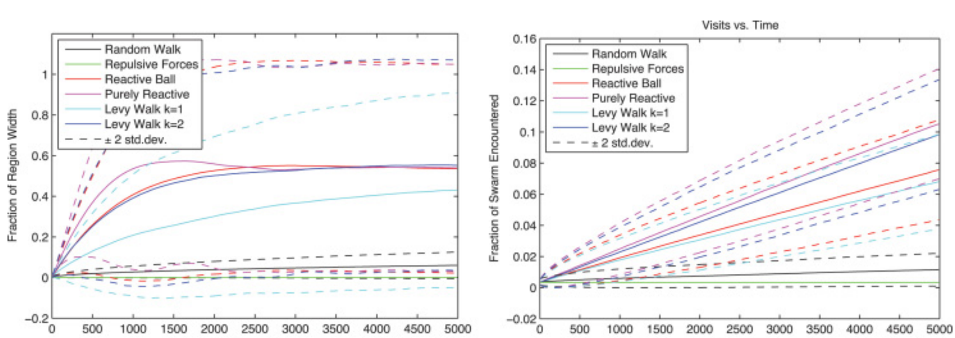
To evaluate the mixing efficacy of the six methods under consideration, we will use two measures: the displacement of a particle from its initial position and the number of other particles that a particle “visits” over time, that is, that it comes within some threshold d meters of. Figure 16 shows the evolution of the displacement and visit metrics over time for the four methods, using a threshold of $d = 10$ meters for visit proximity. As expected, all of the reactive fast dispersion algorithms perform well, while random walk and repulsive forces do not. The underdimension reactive Levy walk with $k = 1$ does perform significantly less well than the other fast dispersion algorithms, as we might expect from the low-constraint dispersion experiment in the last section. Reactive ball dispersion also fares significantly less well in the fraction of the swarm



(a) Random Walk (b) Repulsive Forces (c) Reactive Ball Dispersion

(d) Purely Reactive Dispersion (e) Reactive Levy Walk $k = 1$ (f) Reactive Levy Walk $k = 2$

Fig. 15. Mean density of particles per square meter from $t = 5{,}000$ to $t = 10{,}000$ seconds of dispersion of a swarm of 100 particles from the center of a 100 ×100-meter "maze" of barriers via random walk (a), repulsive forces (b), reactive ball dispersion (c), purely reactive dispersion (d), and reactive Levy walk for $k = 1$ (e) and $k = 2$ (f). Warmer colors indicate logarithmically greater density.



(a) Distance Displaced (b) Swarm Visited

Fig. 16. All of the reactive fast dispersion algorithms effectively mix a swarm, as shown by measurements of distance displaced (a) and fraction of swarm visited (b) in a swarm of 300 sparsely distributed particles.

visited, likely due to the fact that particles are generally moving more slowly and thus have less opportunity to rendezvous.

8. SUMMARY AND CONCLUSIONS

Formalizing and analyzing the problem of swarm dispersion has shown that the goals of aggressive dispersion and even dispersion are compatible in an unconstrained environment but cannot be generally simultaneously optimized in a constrained environment. The three algorithms presented in this article—reactive ball dispersion, purely reactive dispersion, and reactive Levy walks—are all effective methods for dispersion and mixing of swarms. These methods are much faster than prior approaches while requiring only simple computation (readily implementable in electronic or biological hardware), minimal sensing, and no communication between swarm particles. Between them, they offer a range of tradeoffs between aggressiveness and evenness, particularly given the fact that reactive Levy walks can be tuned via their dimensionality parameter to mix properties of aggressiveness, evenness, and centrality.

An important direction for future investigation will be to determine how many various existing applications can be improved by replacing the use of prior less effective dispersion methods. In addition, both analysis and algorithms could be further refined with respect to constrained environments. While the algorithms of this article perform fairly well, there is much room for improvement. For example, a small amount of active coordination between devices might allow faster exploration of newly discovered regions, obtaining some of the benefits of "heavyweight" approaches such as map making without the corresponding communications cost. Finally, for pursuing some applications, it will also be important to ensure that these algorithms can be adapted to cope with environmental constraints such as wind and terrain and maneuverability constraints such as limited turning radius, imprecise movement, and limited energy.

REFERENCES

A. V. Chechkin, V. Y. Gonchar, and M. Szydlowski M. 2002. Fractional kinetics for relaxation and superdiffusion in a magnetic field. *Physics of Plasmas* 9, 78–88.

J. Beal. 2012. A tactical command approach to human control of vehicle swarms. In *AAAI 2012 Fall Symposium on Human Control of Bio-Inspired Swarms*. AAAI Press, 14–20.

J. Beal. 2013. Superdiffusive dispersion and mixing of swarms with reactive levy walks. In *IEEE International Conference on Self-Adaptive and Self-Organizing Systems*. IEEE Press, 141–148.

J. Beal and J. Bachrach. 2006. Infrastructure for engineered emergence in sensor/actuator networks. *IEEE Intelligent Systems* 21, 2, 10–19.

J. Beal, N. Correll, L. Urbina, and J. Bachrach. 2009. Behavior modes for randomized robotic coverage. In *2nd International Conference on Robot Communication and Coordination*. IEEE Press, 1–6.

P. Becker-Kern, M. M. Meerschaert, and H.-P. Scheffler. 2004. Limit theorems for coupled continuous time random walks. *Annals of Probability* 32, 18, 730–756.

S. Benhamou. 2007. How many animals really do the Levy walk? *Ecology* 88, 8, 1962–1969.

D. Brockmann, L. Hufnagel, and T. Geisel. 2006. The scaling laws of human travel. *Nature* 439, 7075, 462–465.

D. Calitoiu. 2009. New search algorithm for randomly located objects: A non-cooperative agent based approach. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA'09)*. IEEE Press, 1–6. DOI:http://dx.doi.org/10.1109/CISDA.2009.5356564

H. Choset. 2001. Coverage for robotics–A survey of recent results. *Annals of Mathematics and Artificial Intelligence* 31, 1–4, 113–126.

J. Eckert, H. Lichte, F. Dressler, and H. Frey. 2012. On the feasibility of mass-spring-relaxation for simple self-deployment. In *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS'12)*. IEEE Press, 203–208. DOI:http://dx.doi.org/10.1109/DCOSS.2012.14

A. M. Edwards. 2011. Overturning conclusions of Lévy flight movement patterns by fishing boats and foraging animals. *Ecology* 92, 6, 1247–1257.

N. Elhage and J. Beal. 2010. Laplacian-based consensus on spatial computers. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'10)*. 907–914.

A. Howard, M. J. Mataric, and G. S. Sukhatme. 2002. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *International Symposium on Distributed Autonomous Robotic Systems*. Springer, 299–308.

N. E. Humphries, N. Queiroz, J. R. M. Dyer, N. G. Pade, M. K. Musyl, K. M. Schaefer, D. W. Fuller, J. M. Brunnschweiler, T. K. Doyle, J. D. R. Houghton, G. C. Hays, C. S. Jones, L. R. Noble, V. J. Wearmouth, E. J Southall, and D. W. Sims. 2010. Environmental context explains Lévy and Brownian movement patterns of marine predators. *Nature* 465, 1066–1069.

M. Keeter, D. Moore, R. Muller, E. Nieters, J. Flenner, S. E. Martonosi, A. L. Bertozzi, A. G. Percus, and R. Levy. 2012. Cooperative search with autonomous vehicles in a 3D aquatic testbed. In *2012 American Control Conference (ACC'12)*. IEEE Press, 3154–3160.

M. Kotulski. 1995. Asymptotic distributions of continuous-time random walks: A probabilistic approach. *Journal of Statistical Physics* 81, 3–4, 777–792. DOI:http://dx.doi.org/10.1007/BF02179257

L. Ludwig and M. Gini. 2006. Robotic swarm dispersion using wireless intensity signals. In *Distributed Autonomous Robotic Systems 7*, M. Gini and R. Voyles (Eds.). Springer, Japan, 135–144. DOI:http://dx.doi.org/10.1007/4-431-35881-1_14

B. B. Mandelbrot. 1983. *The Fractal Geometry of Nature*. W. H. Freeman, New York.

E. Mathews, T. Graf, and K. S. S. B. Kulathunga. 2012. Biologically inspired swarm robotic network ensuring coverage and connectivity. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC'12)*. IEEE Press, 84–90. DOI:http://dx.doi.org/10.1109/ICSMC.2012.6377681

J. McLurkin and J. Smith. 2004. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *7th International Symposium on Distributed Autonomous Robotic Systems (DARS'04)*. Springer, 399–408.

N. Mercadier, W. Guerin, M. Chevrollier, and R. Kaiser. 2009. Lévy flights of photons in hot atomic vapours. *Nature Physics* 5, 8, 602–605.

MIT Proto. 2012. MIT Proto. Software available at http://proto.bbn.com/.

J. Mullins, B. Meyer, and A. P. Hu. 2012. Collective robot navigation using diffusion limited aggregation. In *Parallel Problem Solving from Nature (PPSN XII'12)*, C. A. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone (Eds.). Lecture Notes in Computer Science, Vol. 7492. Springer, Berlin, 266–276. DOI:http://dx.doi.org/10.1007/978-3-642-32964-7_27

S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, K. Shirai, S. Koizumi, and H. Ishiguro. 2010. An adaptive switching behavior between Levy and Brownian random search in a mobile robot based on biological fluctuation. In *Intelligent Robots and Systems (IROS'10)*. IEEE Press, 1927–1934. DOI:http://dx.doi.org/10.1109/IROS.2010.5651671

J. Oyekan, H. Hu, and D. Gu. 2010. Exploiting bacterial swarms for optimal coverage of dynamic pollutant profiles. In *2010 IEEE International Conference on Robotics and Biomimetics (ROBIO'10),* IEEE Press, 1692–1697. DOI:http://dx.doi.org/10.1109/ROBIO.2010.5723586

M. J. Plank and E. A. Codling. 2009. Sampling rate and misidentification of Lévy and non-Lévy movement paths. *Ecology* 90, 12, 3546–3553.

S. Poduri and G. S. Sukhatme. 2007. Latency analysis of coalescence in robot groups. In *IEEE International Conference on Robotics and Automation (ICRA'07)*. IEEE Press, 3295–3300.

G. Ramos-Fernndez, J. L. Mateos, O. Miramontes, G. Cocho, H. Larralde, and B. Ayala-Orozco. 2004. Levy walk patterns in the foraging movements of spider monkeys (Ateles geoffroyi). *Behavioral Ecology and Sociobiology* 55, 3, 223–230.

I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong. 2011. On the Levy-walk nature of human mobility. *IEEE/ACM Transactions on Networks* 19, 3, 630–643. DOI:http://dx.doi.org/10.1109/TNET.2011.2120618

Russel Smith, et al. 2001 to 2010. Open Dynamics Engine (version 0.11). http://www.ode.org.

F. L. Schuster and M. Levandowsky. 1996. Chemosensory responses of Acanthamoeba castellani: Visual analysis of random movement and responses to chemical signals. *Journal of Eukaryotic Microbiology* 43, 150–158.

M. Shin, S. Hong, and I. Rhee. 2008. DTN routing strategies using optimal search patterns. In *3rd ACM Workshop on Challenged Networks (CHANTS'08)*. ACM, New York, NY, 27–32. DOI:http://dx.doi.org/10.1145/1409985.1409992

M. F. Shlesinger, B. J. West, and J. Klafter. 1987. Levy dynamics of enhanced diffusion: Application to turbulence. *Physics Review Letters* 58 11, 1100–1103. DOI:http://dx.doi.org/10.1103/PhysRevLett.58.1100

M. F. Shlesinger, G. M. Zaslavsky, and J. Klafter. 1993. Strange kinetics. *Nature* 363 (May), 31–37.

D. W. Sims, E. J. Southall, N. E. Humphries, G. C. Hays, C. J. A. Bradshaw, J. W. Pitchford, A. James, M. Z. Ahmed, A. S. Brierley, M. A. Hindell, D. Morritt, M. K. Musyl, D. Righton, E. L. C. Shepard, V. J. Wearmouth, R. P. Wilson, M. J. Witt, and J. D. Metcalfe. 2008. Scaling laws of marine predator search behaviour. *Nature* 451, 1098–1102.

G. M. Viswanathan, S. V. Buldyrev, Shlomo Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley. 1999. Optimizing the success of random searches. *Nature* 401, 911–914.

Received March 2014; revised July 2014; accepted October 2014