Chemicals detection with limited-robots clouds

Charles Sutton & Yohai Nadiv

July 11, 2017

1 Introduction

In this paper we will present the work of Yotam Elor and Alfred M. Bruckstein [EB14] designing the control for a group of robots so that they will follow the gradient of a scalar field defined over space. Their claim is to have the swarm of agents motion as the result of simple memoryless interactions between the robotic agents. We then will consider the boundaries of the solution regarding multiple maximum points and will offer an improve to the algorithm which will try to cope with multiple maximum points at the scalar field.

It is often assumed that the scalar field is the concentration of some chemical material and is generated by a diffusion process originating at a source. Therefore, the value of the scalar field becomes weaker as one moves away from it. Hence, the problem is sometimes called source-seeking with applications varying from tracking a plume of gas to finding the source of leakage of an hazardous chemical.

A large group of miniature (possibly nano) and very simple robotic agents with very low capabilities is considered. Very small robots, even equipped with multiple sensors, are unable to directly sense the gradient at any point on the scalar field, because the spatial separation between the on board sensors is not large enough. Hence, it is assumed that the agents can only take point measurements of the scalar value at the current point. We limit the discussion to memoryless, or reactive, algorithms in the sense that an action performed by an agent is determined solely by the agent's sensors readings at the time the action is taken. Since they are memoryless, the agents cannot estimate the gradient by comparing the currently measured value to previously sensed values and, by assumption, they cannot communicate directly thus cannot explicitly share their point measurements. Full mutual sensing of position is assumed, i.e. every agent can sense the relative position of all other agents in the group. This assumption might seem quite strange and restrictive, however our results prove that by tuning the algorithm parameters the swarm can be made as cohesive as desired. Specifically, the swarm can be made cohesive enough to guarantee that at (almost) all times, all agent pairs are close enough to allow mutual sensing of relative position. The proposed process can be described as a gathering or swarming algorithm augmented to allow gradient climbing. The agents follow the swarming algorithm while their speed is controlled by the scalar field.

When considering multiple maximum points the algorithm will find only 1 maximum and the swarm will climb there. We propose then to control the vision of the agents so that as the scalar measurement of an agents grows his range of vision grows as well. When the measurement is below the minimum at the algorithm the agent will perform random walk. This will create clusters of ants near the maximums and then each cluster will perform the same gradient climbing from the original algorithm. Furthermore we to prevent all the agents to gather to 1 cluster and therefore find only 1 maximum point, we added a dependency between the temperature of the system and the vision range. The temperature will add an initial random walk to the agents which will let them spread throw the field and will prevent them from gathering too early thus missing some maximum points, as the time will pass the temperature will cool down and then the agents will start to follow the original range vision law.

2 Introduction to gradient climbing with point measurements

As aforementioned robots have very limited properties. They are memoryless and unable to communicate and thus an agent can solely measure the value of the scalar field in its current position

but cannot share it with other agents. The agents' task is then to perform a gradient ascent[1] in the absence of direct communication. This section describes the solution proposed by [EB14] and show its limits.

2.1 Notations

This work is considered in the 2-dimensional space $\mathbb{R}^2$. By assumption there is a scalar field, $\rho(\cdot)$ defined over the space $\rho : \mathbb{R}^2 \to \mathbb{R}$. For natural scenari[2] the scalar field can be considered as levels pollution over $\mathbb{R}^2$, coming from a chemical source. The goal of the swarm is to perform gradient ascent and move toward higher scalar zones or where maximal[3] scalar values are reached.

The swarm is composed by N agents denoted by $\{r_1, ..., r_N\}$ and the position of the i^{th} agent is denoted by $X_i = \left[x_i^1, x_i^2\right]^T$. Note that the global coordinate frame is unknown to the agents and is defined here solely for convenience of analysis.

Let X_{cm} be the agents' center of mass, which, in the global coordinate system, is given by :

$$X_{cm} = \sum_{i=1}^{N} X_i \tag{1}$$

And let $\rho(X)$ denote the value of the scalar field at point X. In our model the time is discrete and the state of the i^{th} agent position at time t is denoted by $X_i(t)$. The system is initialized at $t = 0$ and the initial position of the N agents is determined at each experiment.

2.2 A gradient-climbing process

The solution to "climb the gradient" is proposed by a recent article [EB14]. It combines elegantly randomness with a gathering process.

We first describe and recall the gathering process for a better understanding of the proposed solution. If Ψ denote the sum of inter-robot distances :

$$\Psi(X_1, ..., X_N) \triangleq \frac{1}{2} \sum_{i,j=1}^{N} dist(X_i, Xj)^2 \tag{2}$$

where $dist(X_i, X_j)$ is the Euclidian distance between agent i and agent j.

$\Psi(X_1, ..., X_N) = 0$ trivially means that all the agents are standing in the same position. Therefore, a natural gathering algorithm is given by the following gradient descent[4] process in which every agent follows the local gradient of $-a\Psi$:

$$\frac{\partial X_i}{\partial t} \triangleq -a \frac{\partial \Psi}{\partial X_i} = a \sum_{j=1}^{M} (X_j - X_i) \tag{3}$$

In fact if $a = \frac{1}{N}$ in Eq.3, every agent moves toward the group's instantaneous center of mass and the agent's speed is proportional to the distance to the center of mass, since :

$$\frac{1}{N} \sum_{j=1}^{M} (X_j - X_i) = \left(\frac{1}{N} \sum_{j=1}^{M} X_j \right) - X_i = X_{cm} - X_i \tag{4}$$

and, in the gathering proposed above, the speed of each agent toward the center of mass is proportional to the distance between them. Therefore, a well known property of this gathering process is that to the center of mass is stationary and unaffected by agents motion toward it over time.

Inspired by a previous work they made [EB12], the method proposed in [EB14] is to augment this gathering process to ensure gradient climbing. They propose to make speed of each agent dependent on its current point measurement where the higher is the measurement the lower is the speed. Intuitively, agents in positions with higher measurements will move relatively slower than

[1] Ascent is more relevant to our project but it also works for gradient descent.
[2] The plural of scenario.
[3] And eventually local maxima
[4] descenting the gradient of Ψ

agents with lower measurements and therefore, the center of mass will drift in the direction of higher scalar values.

In order to ensure center of mass drifting and also to prevent early gathering, they add randomness in the move of each agent. **This addition of randomness is critical to the success of the algorithm**. Finally in order to perform gradient ascent they propose following law of motion :

$$X_i(t+1) = X_i(t) + R_i(t) + (1 - \alpha - \beta\rho(X_i))(X_{cm}(t) - (X_i(t)) \tag{5}$$

where α and β are constants chosen so that $0 < \alpha < \frac{1}{2}$ and $0 < \beta < \frac{(1-2\alpha)}{\rho_{max}}$. $R_i(t)$ is the random step taken by agent i at time t. $R_i(t)$ is distributed as a 2-dimensional Gaussian vector with zero mean and a covariance matrix $\Sigma = I \cdot \sigma_R^2$

Each member of the law of motion in Eq. 5 has its importance in the success of the algorithm :

- $X_{cm}(t) - X_i(t)$ each agent goes toward to the center of mass of every agent exactly as the law of motion described in the gathering process (cf. Eq. 3).
- $R_i(t)$ is the addition of randomness that actually makes the swarm drift. It also prevents the swarm from (early) gathering into a single point.
- $1 - \alpha - \beta\rho(X_i)$ corresponds to the speed of the agent toward the center of mass. α and β are positive constants and are designed so that the speed is always positive. This member shows that the higher is the measurement the lower is the speed and therefore agents on higher scalar levels are slower than the rest of the swarm.

2.3 Experiments and analysis

2.3.1 Constant gradient

The first experiment presented in the seminal paper [EB14] consider the nano-case where in most scenario the differential of temperature of pollution is minimal. The scalar field is denoted by the following function :

$$\rho(x_1, x_2) = 0.01x_1 \tag{6}$$

The gradient is constant since ρ is linear in the first coordinate. The experiment is considered in a square where boundaries are $100 < x_1 < 108$ and $-4 < x_2 < 4$.

We have reproduced this experiment on NetLogo [Wil99] illustrated in figure 1

They also provide a lower bound on the speed of the center of mass (Theorem 6 in [EB12]).

For every $t \geq t_{stable}$:

$$\mathbb{E}\left[x_{cm}^1(t)\right] \geq x_{cm}^1(t_{stable}) + \beta\rho_0\sigma_R^2 \cdot (t - t_{stable}) \tag{7}$$

$$\mathbb{E}\left[x_{cm}^2(t)\right] = x_{cm}^2(t_{stable}) \tag{8}$$

where t_{stable} designs the time when the distribution of the agents around the center of mass is close enough to the normal distribution.

For the first coordinate, this theorem shows a lower bound linear in time. We have reproduced the figure 1 of [EB14] showing that this bound corresponds well to simulations

Since the scalar field is only dependent on the first coordinate, the average position on the second dimension is centered on the position when the swarm is stable.

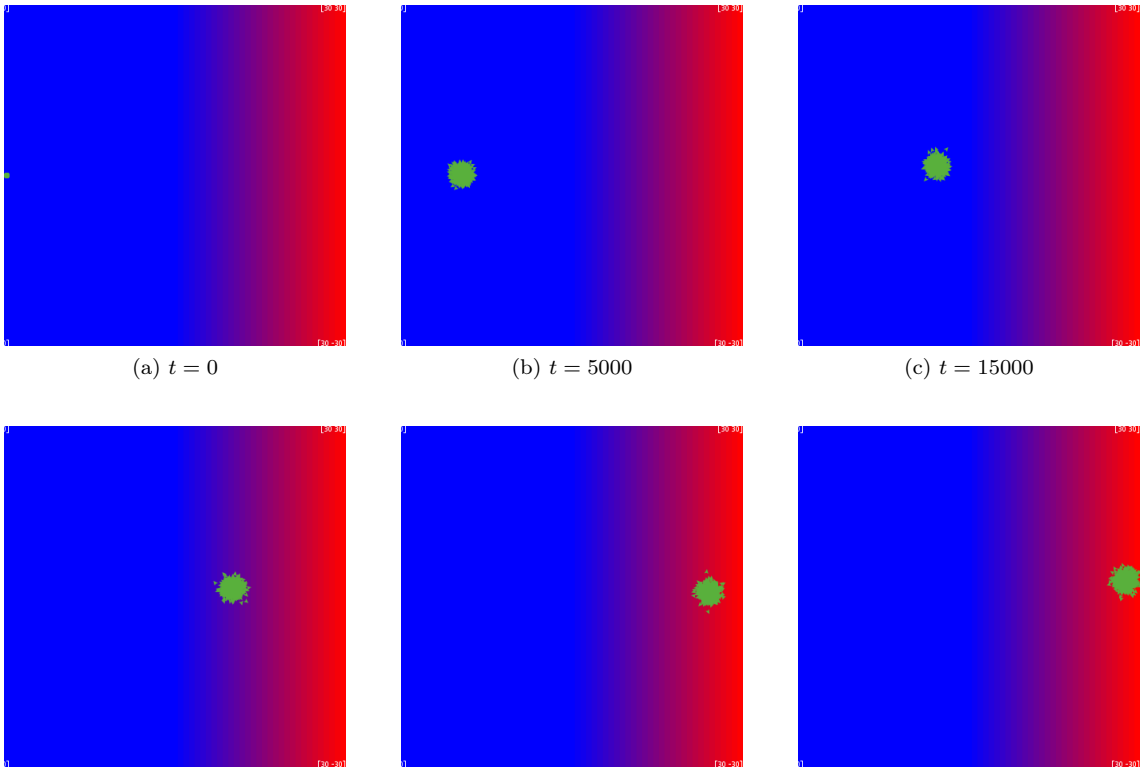
??

2.3.2 Gaussian diffusion

Another case of interest is when we consider the case where it exists only a single maximum in the scalar field. In natural scenario it corresponds to the case of the diffusion of a chemical pollutant.

We have in our research considered the case of chemicals with Gaussian diffusion. This scalar field is similar to the one presented in figure 3 in [EB12].

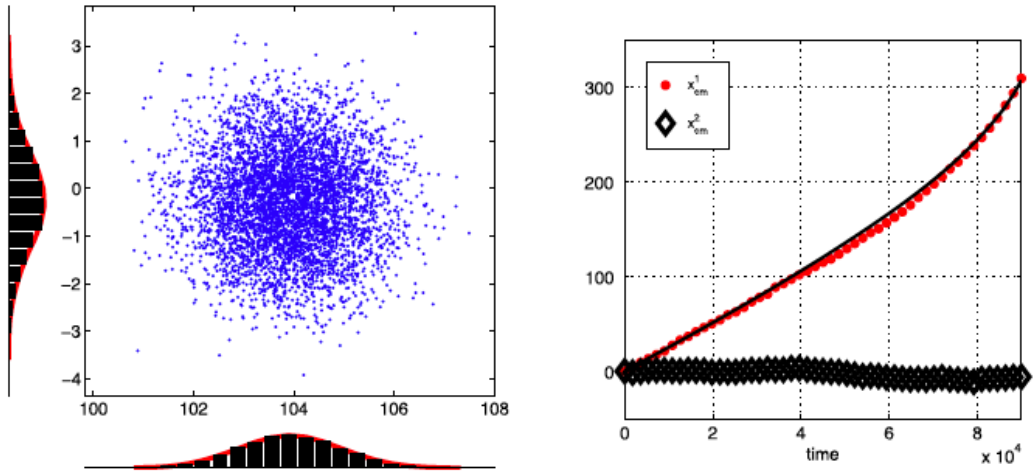
If we consider a chemical C having the position (c_1, c_2) the the scalar field is defined by :



(a) $t = 0$ (b) $t = 5000$ (c) $t = 15000$

(d) $t = 25000$ (e) $t = 35000$ (f) $t = 40000$

Figure 1: Snapshot of the swarm climbing the constant gradient field defined in Eq. 6. The swarm is composed by 2000 agents following the motion law of Eq. 5. In this experiment $\sigma_R = 1$, $\alpha = 0.1$ beta $= 0.25$.



(a) A snapshot of the agents' distribution while executing the algorithm with the histograms of the marginal distributions for $\hat{x}^1$ and $\hat{x}^2$. The solid red lines are the predicted marginal distributions

(b) The time course of the components x^1_{cm}, x^2_{cm} of X_{cm}. The solid lines are the theoretical predictions and the markers represent the actual results

Fig. 1. Results of a single simulation run of a swarm comprising 5000 agents with $\sigma^2_R = 1$, $\alpha = 0.1$, $\beta = 0.25$. A constant gradient field was assumed i.e. $\rho(X) = 0.01 \cdot x^1$ ($\nabla\rho(X) = 0.01\hat{x}^1$).

Figure 2: This figure is a reproduction of the figure 1 of [EB14]. It shows the Gaussian distribution over the two coordinates of the agents' position. It also shows that the lower bound in the speed of the first coordinate is tight to the speed reported in simulations with different parameters.

$$\rho(x) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{dist(x,c)^2}{2}\right) \tag{9}$$

The gradient climbing also works in this case, we report in 3 the simulation we performed in NetLogo.

2.4 Limits of this solution

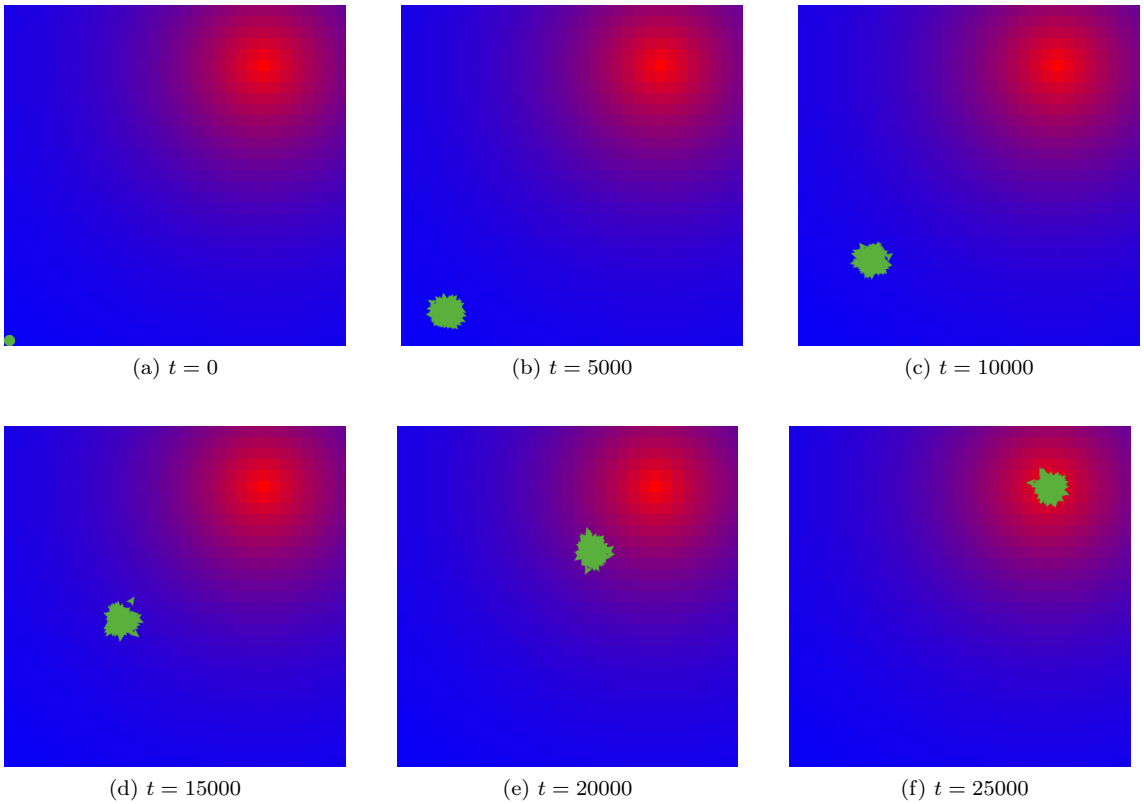
The solution of gradient climbing proposed by [EB14] seems to work greatly on the scenario presented in the paper. The swarm is able to climb a gradient even each agent cannot directly communicate and can perform point measurement and mutual sensing. Our research work focused on improving this algorithm. We have imagined to make it more efficient or adapt it to more complicated scenario.

In fact due to the mutual sensing, we have remarked that the gathering process induced by this algorithm make impossible for the swarm to separate in sub-swarms. We have performed in figure 4 simulations in two cases where two chemicals pollutants are diffusing. With the original algorithm the swarm still climb the pollution gradient, but it fail to "detect" multiple chemicals by separating in sub-swarms.

In the case of chemical detection the ability to separate is critical for the swarm if there are multiple sources to detect. Here, in the case of multiple chemicals diffusing in the scalar field, the swarm will move only in the direction of a single chemicals.

Therefore, our research consisted in improving this algorithm to enable the swarm to separate and therefore solve the issue of efficiently detecting multiple chemicals pollutants diffusing in the scalar field.

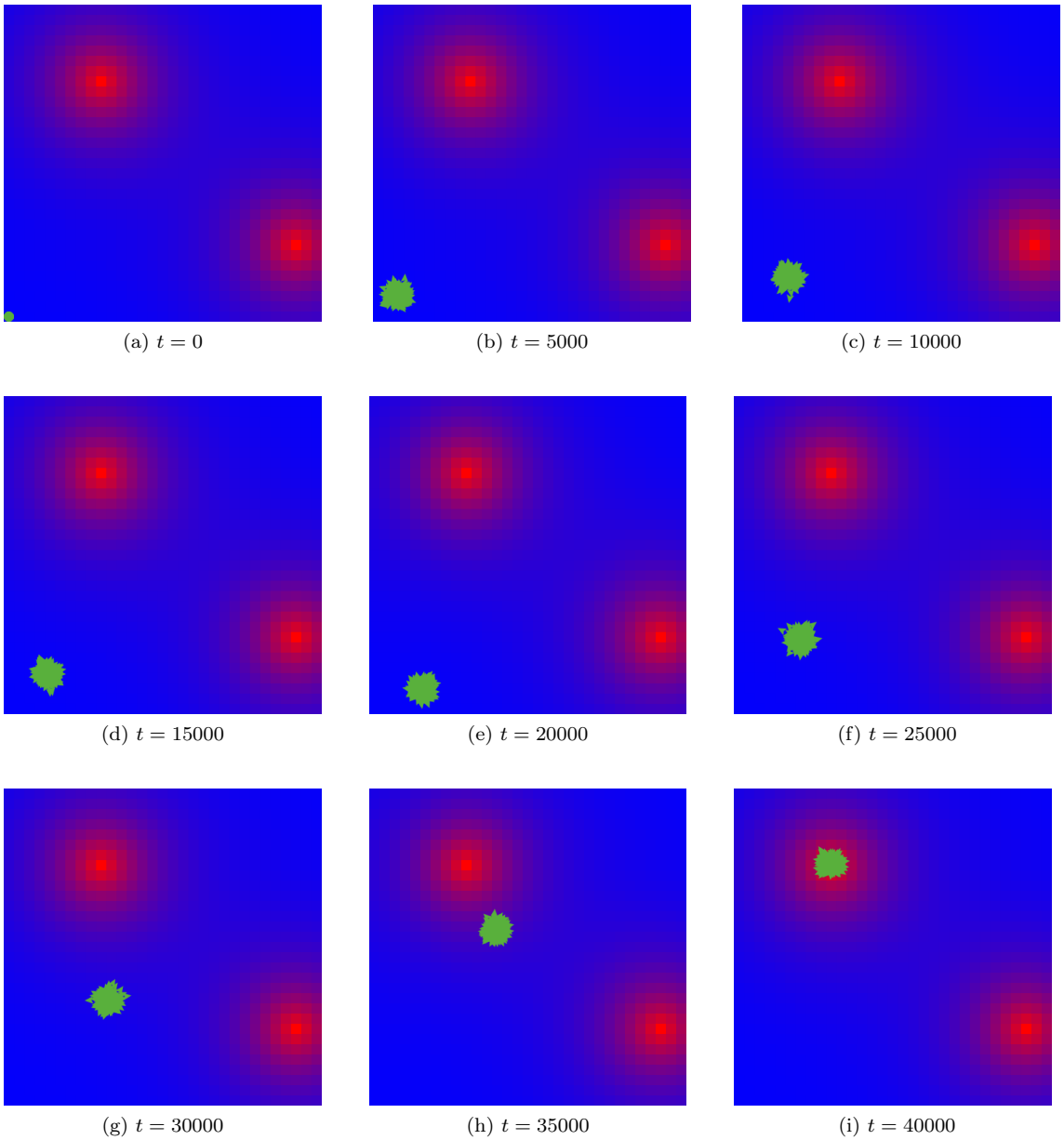
For chemical detection application this is critical to allow the swarm to separate in sub-swarms this is the purpose of the next section.



(a) $t = 0$ (b) $t = 5000$ (c) $t = 10000$

(d) $t = 15000$ (e) $t = 20000$ (f) $t = 25000$

Figure 3: Snapshots of a swarm of 1000 agents climbing the gradient coming from a chemical with Gaussian diffusion and with $\sigma_R = 0.8$, $\alpha = 0.1$ and $\beta = 0.25$. The swarm starts from the lower-left corner and move cohesively toward the chemical source.



(a) $t = 0$ (b) $t = 5000$ (c) $t = 10000$

(d) $t = 15000$ (e) $t = 20000$ (f) $t = 25000$

(g) $t = 30000$ (h) $t = 35000$ (i) $t = 40000$

Figure 4: Snapshots of a swarm of 1000 agents climbing the gradient coming from two chemicals with Gaussian diffusion. The experiment is in the same condition than figure 3. Because of the full mutual sensing the swarms can only climb a single chemical diffusion and fail to separate in sub-swarms to detect all the chemicals.

3 Multiple maximum points

3.1 Vision range in proportion to the point measurement

The issue raised in section 2.4 is that the swarm is unable to separate in multiple sub-swarms when there are multiple ways of ascent. In order to solve the problem presented in section 2.4 we want the agents to separate into multiple clusters near all the maximum points and only then start the gradient ascent, as previously presented. The solution we propose is first to limit the vision of the agents, and then to broaden their vision proportionally to the current point measurement of the scalar field, so that the higher is the point measurement the broader is the specific agent's vision range. Agents which have point measurement lower than a fixed scalar lower bound perform random walk. The lower bound is 0 in our experiments, but you can imagine cases with a background noise, for an example agents seeking a radioactive radiation source, as there possibly could be a background radiation unlinked to the source which could affect the gradient ascent. Hence our solution augment the original algorithm by changing the vision according to each agent's point measurement.

To apply our algorithm, we have release agents in a "cold point" meaning a position where the measurement is lower or equal to our scalar lower bound. Therefore, agents start to walk randomly until they first measure a point with a scalar value higher than the bound, meaning they are getting near an "hot" point (a maximum point in the scalar field). This will broaden their vision range to see other agents near them, therefore, it will invoke them to gather in clusters[5] near the "hot" points. The clusters then drift to the hot points exactly as they would have done in the original algorithm as they are **locally** in the same conditions as the one presented in the original problem. In their local environment, limited by their limited vision, the swarm .

The general rule for vision range we propose is :

$$VR(X) = \gamma \times \frac{\rho\,(X)}{\rho_{max}} - min \tag{10}$$

- ρ is the functional defining the scalar field. We assume reasonably to know ρ_{max} which is defined as the maximum value of diffusion of a the chemical of interest.
- γ is a strictly positive variable that is most often close to 1. In our experiment γ defined as the range capturing most of the diffusion of the chemical :

$$\int_{\mathcal{B}_{C,\gamma}} \rho(X)\,\mathrm{d}X = 0.9 \times \int_{\mathbb{R}^2} \rho(X)\,\mathrm{d}X \tag{11}$$

  where $\mathcal{B}_{C,\gamma}$ denotes the ball of center C and radius γ.

  The higher is γ the smaller the more agents will gathers in clusters and find maximum points (as we saw previously that every cluster is able find only 1 maximum point). On the other hand, the higher is γ is the faster the agents will gathers, because there vision range will grow faster as they come close to an hot point, which means that the system will get faster to a stable point with no agents performing a random walk.

- min is lower bound on scalar values as discussed. It can be the maximum or the average value of the noise observed in the background. In our experiments the noise is absent hence $min = 0$.

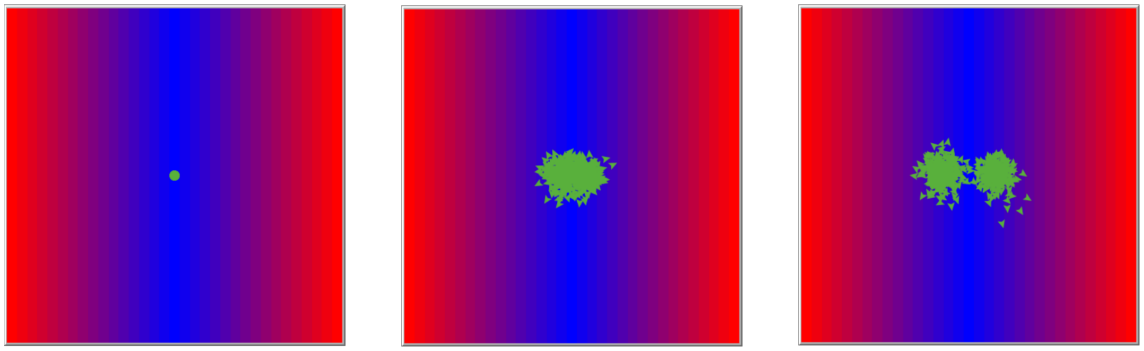
3.2 Experiments

This section illustrates how the law of motion introduced in Eq. 10 solves the issue raised in section 2.4.

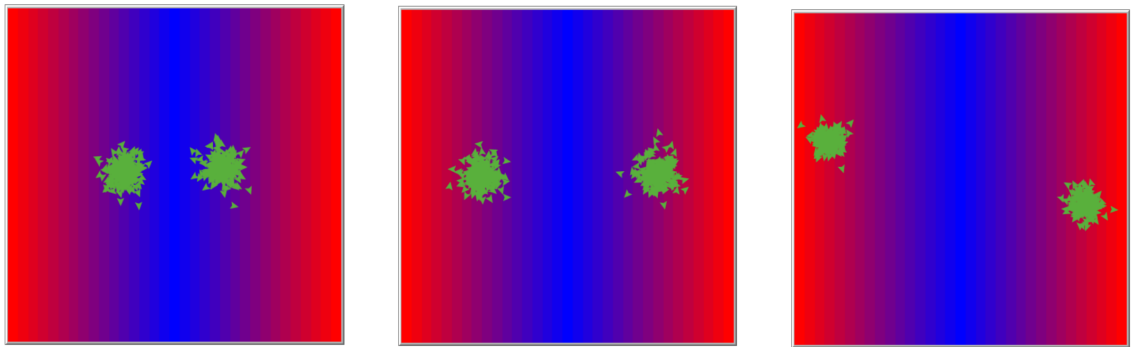
3.2.1 Two constant gradient fields

In the first experiment we release 500 agents in the middle of 2 constant gradient fields and we set $\gamma = 0.5$. As we can see in figure 5, agents briefly start by a random walk before and then separate to form 2 clusters very quickly. Each cluster starts its gradient ascent towards the 2 maximum

[5] Due to the gathering process used in the original algorithm.



(a) $t = 0$ (b) $t = 3$ (c) $t = 9$



(d) $t = 15000$ (e) $t = 20000$ (f) $t = 25000$

Figure 5: Snapshot of 500 agents released in a cold spot between to linear fields. In this experiment $\gamma = 0.5$

point simultaneously, similarly as the performance we observed with the original law of motion in the single constant gradient field.

3.2.2 Two chemicals sources with Gaussian diffusion fields

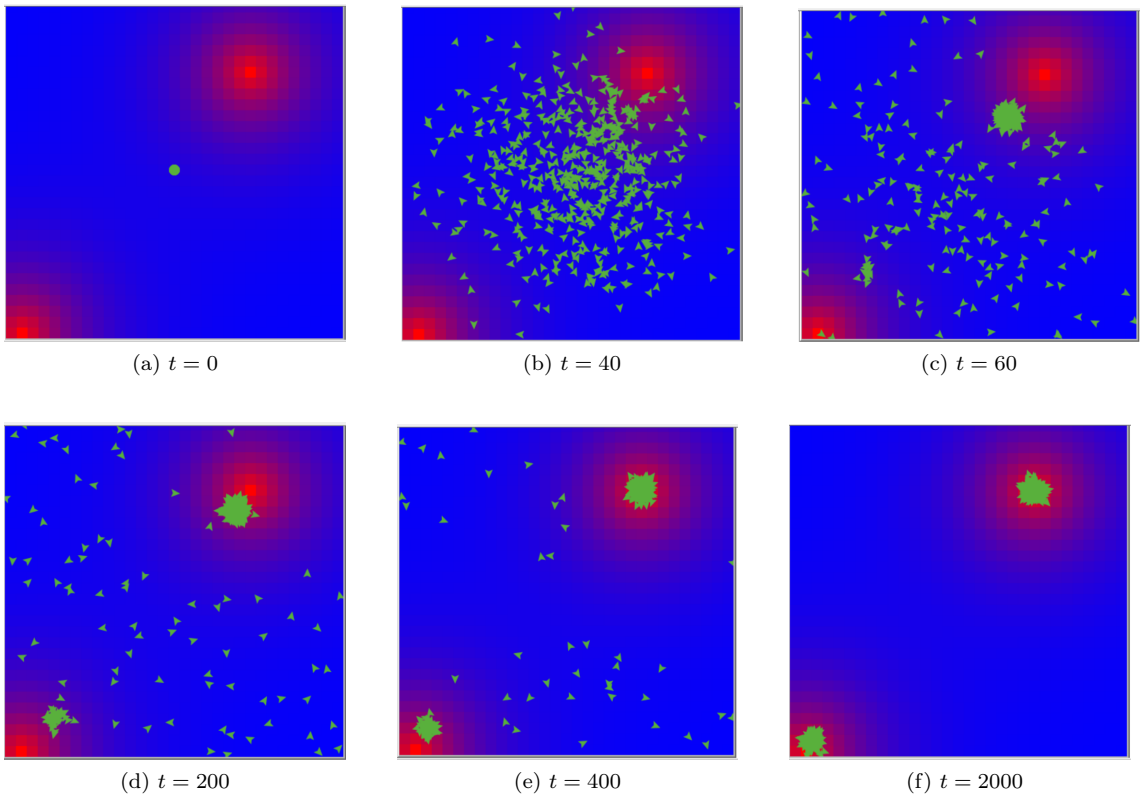
In this experiment we release 500 agents in a "cold point" surrounded by 2 chemical with Gaussian diffusion as described in equation 9 and we set $\gamma = 0.5$. As expected, they start with a random walk until some agents get closer to hotter positions. These agents gather in clusters near the chemical sources and start the gradient ascent towards the maximums. In the meanwhile, the unconstrained agents perform their random walk until they finally got near one of the hot point and join a cluster already performing its ascent. If the ascent is already finished they just join the cluster stabilized on the chemical source.

3.2.3 Multiple chemical sources with Gaussian diffusion fields

In this experiment we release 1000 agents in a cold point surrounded by 5 chemicals sources with Gaussian diffusion fields and we set $\gamma = 0.3$.

In this experiment the distribution of the maximum points over the area affects greatly the ability of the agents to find all the maximum points successfully. For instance, if a field is "covered" by all the other fields; 4 hot points surrounds the initial point of the agents and the 5th point beyond them with regard to the initial starting point of the agents. The agents are not able to find it, this will be discussed more thoroughly in section 3.3.

For this experiment let us assume that the chemical sources are distributed so that no source is "shadowed" by other fields. We use 1000 agents so we will have enough agents distributed over all the hot points. We don't need prior knowledge on the number of hot points as we can certainly limit from above the number of hot points in the area and just release 200 per hot point. That is not part of our research but we found it a suitable ratio,this can be studied furthermore. As we can see in the snapshots presented in figure 8, the agents are able to find all 5 hot points exactly as before.



(a) $t = 0$ (b) $t = 40$ (c) $t = 60$

(d) $t = 200$ (e) $t = 400$ (f) $t = 2000$

Figure 6: Snapshots of a swarm of 500 agents released in a cold spot surrounded by 2 chemicals with Gaussian diffusion. $\gamma = o.5$

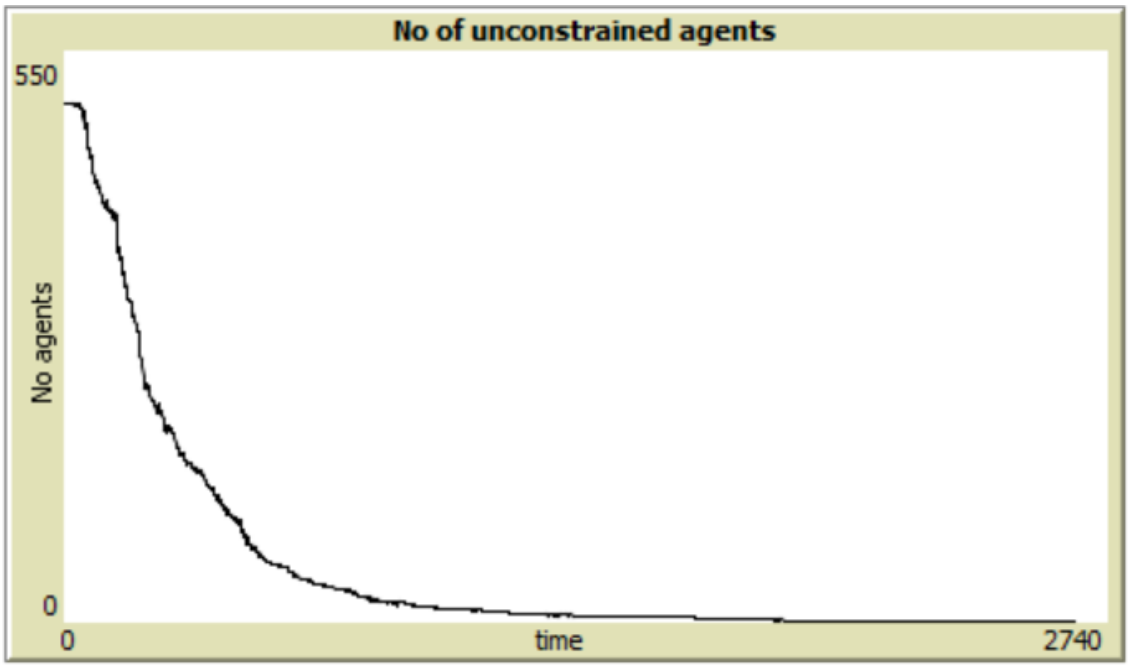
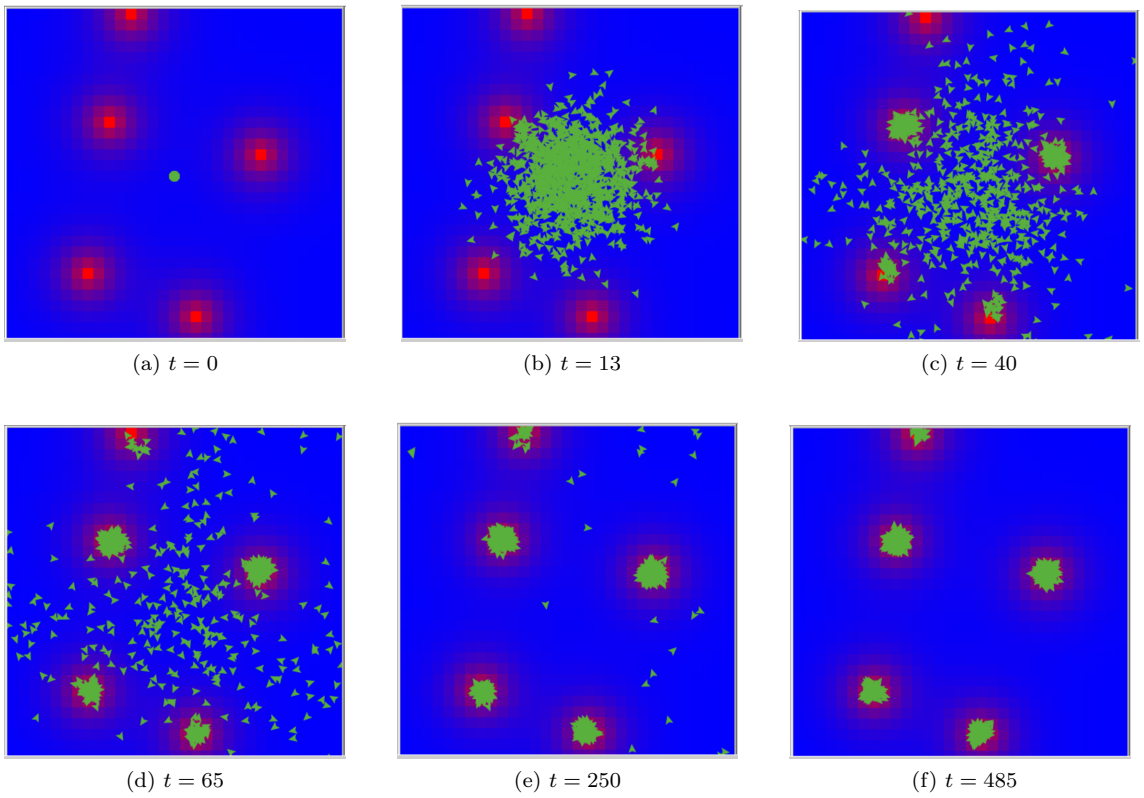


Figure 7: A graph describing the number of unconstrained agents in relation to the time, as we can see this is an exponential decline



(a) $t = 0$ (b) $t = 13$ (c) $t = 40$

(d) $t = 65$ (e) $t = 250$ (f) $t = 485$

Figure 8: Snapshots of a cloud of 1000 agents released in a cold point surrounded by 5 Gaussian diffusion fields. In order to prevent merging of clusters we set $\gamma = 0.3$

It is important to point out that if we pick γ to be too big the agents will not be able to find 2 close hot points. That is because when they get near the hot points, their vision range will be too big and they will see the agents gathering around the 2 hot points, this will invoke them to gather to 1 big cluster and reproduce the limitation of the original algorithm. Even if initially they are separated to multiple clusters, the clusters will ultimately join together to 1 cluster, then they will randomly pick 1 hot point and will eventually ascent to it. The ratio between the minimum of the distance, between 2 close hot points, and γ can be studied furthermore, we found that $\gamma = 0.3$ is enough to deal with 2 close hot points as shown in the snapshots above.

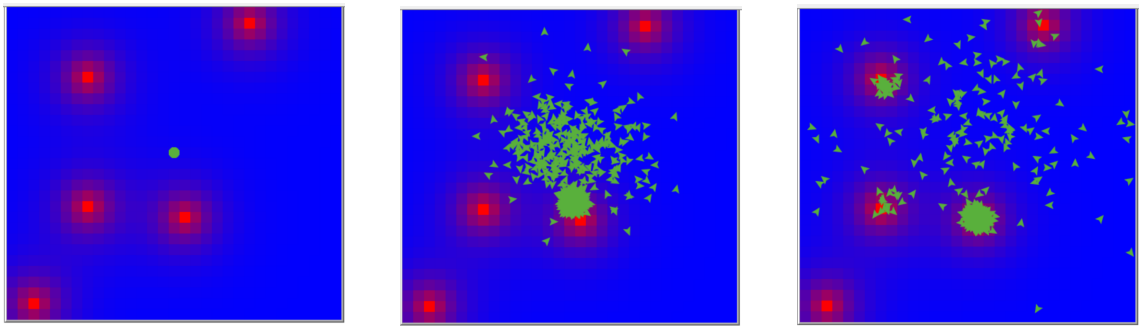
3.3 Limits of the solution

The solution we proposed is limited to the fact that agents have to start in a "cold point" so that they don't gather too early and find only a single point as previously discussed. This require us to assume some prior knowledge on the distribution of the fields, an assumption that weakens the solution comparing the original solution with 1 hot point which didn't require any prior knowledge.

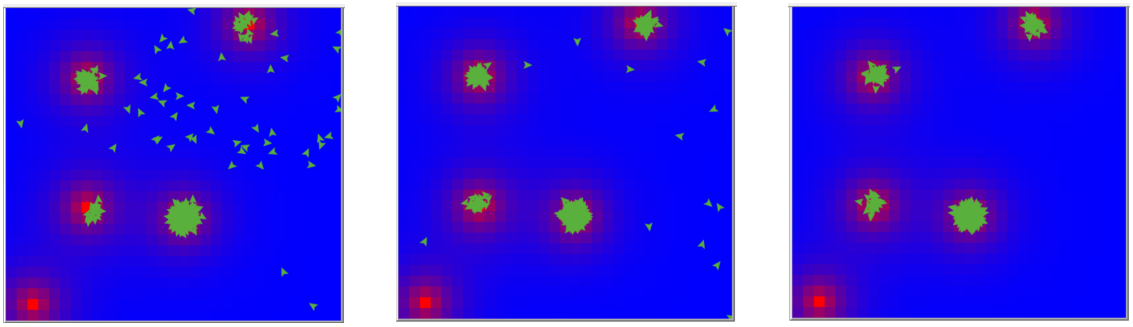
The problem of our solution is that even if we assume such knowledge is reasonable, as we saw in the experiment in section 3.2.3 the success of the algorithm is very dependent on the system constellation, on the sources position in relation to the initial positions of the agents. The example showed in section 3.2.3 presents this problem very clearly.

In the snapshots presented in figure 9 we can see the case of a masked hot point as presented above in the experiment 3.2.3. In this case most of the agents gather to clusters near the shadowing hot points. Then the unconstrained agents perform a random walk, but even if they get near to the shadowed hot point there is not enough agents to initiate a stable gradient ascent. Even if the agent is on the maximum point there is not enough agents to constraint him to their center of mass and it will randomly get off the hot point (little more slow due the low of motion), so they just come and go from the hot point until they get to 1 of the other hot points which has enough agents to keep them in their center of mass.

Therefore we needed to improve the algorithm and find a solution to those problems



(a) $t = 0$ (b) $t = 13$ (c) $t = 40$



(d) $t = 80$ (e) $t = 146$ (f) $t = 450$

Figure 9: Snapshots of 1000 agents released in a cold point surrounded by 5 Gaussian diffusion fields. 1 of those fields is masked by the other fields and therefor is not detected.

3.4 Improvement by Temperature of the system

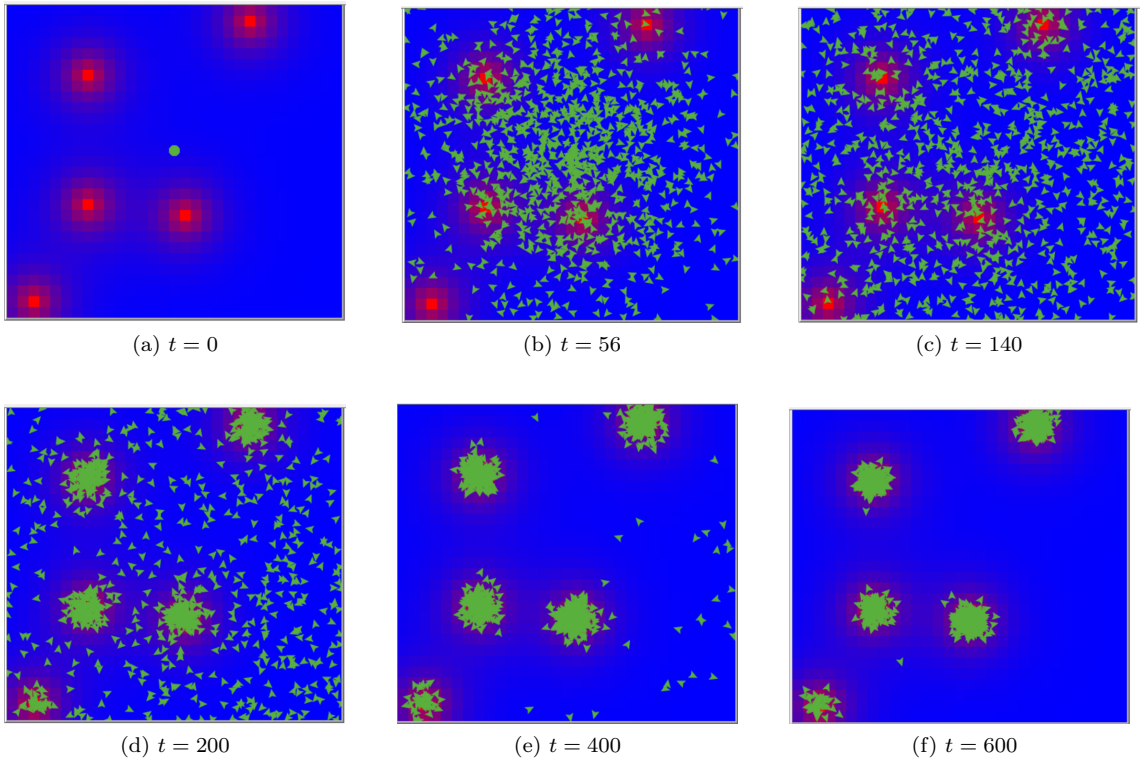
In order to solve all the problems discussed in section 3.3 we added the temperature of the system as a variable influencing the vision range. At temperature 0 the agents will have vision range 0 thus they will perform a total random walk, as the time passes and the temperature rises the vision range will become more and more as the original vision range as presented in equation 10. The general idea is to allow the agents to spread all over the area before starting the algorithm. The temperature will prevent the agents from gathering right away in the case they were initially put near an hot point thus removing the necessity of prior knowledge on the system. Moreover, if we look at the case of a covered hot point, as presented in section 3.3, we expect the agents to initially make a random walk thus some agents will surely pass through the covering points and get near the covered point. Then when the temperature of the system will rise they will be able to make a cluster near the covered point and therefore will start drifting toward it in a gradient ascent hopefully finding all the hot points in the system. Although it will suffice to give the agents some ticks to make total random walk a period of time that will do to spread them equally in the terrain (will be obviously proportional to the area's size), we found it better to rise the temperature in a fashion that will allow agents already passing through the hot points themselves (or very close to them) an option to make a cluster and thus to save time afterward finding and drifting toward the hot points.

The new vision range law is:

$$VR(X,t) = \tau(t) \times \gamma \times \frac{\rho(X)}{\rho_{max}} - min \tag{12}$$

Where $\tau(t)$ is a function denoting the temperature of the system over time. For instance we can have:

- $\tau(t) = \begin{cases} 0 & \text{if } t \leq t_0 \\ 1 & \text{if } t > t_0 \end{cases}$ where t_0 is fixed a priori.
- $\tau(t) = \frac{2}{\pi} arctan(t)$



(a) $t = 0$ (b) $t = 56$ (c) $t = 140$

(d) $t = 200$ (e) $t = 400$ (f) $t = 600$

Figure 10: Snapshots of 1000 agents released in a cold point surrounded by 5 Gaussian diffusion fields. Although 1 of those fields is masked by the other fields he is detected due to the temperature element in the algorithm.

In the snapshots presented in figure 10 we can see the exact same setup as in figure 9. Now, with the temperature improvement, the agents were able to find all 5 points. In this experiment we set the temperature function to increase by 33 percent every 100 ticks so at 300 ticks the agents will have the original vision range presented in equation 10.

4 Conclusion

The solution of the paper is a great idea for how to find a maximum point using gradient ascent/descent without having all the necessary information for the known algorithm. The solution works well for one cluster of agents and one maximum point. The problem begins when we get multiple fields in the system.

Our solution depends totally on the original solution. We isolated the problem of multiple fields, and after understanding that the key is to separate the cloud of agents to little clusters, so each little cluster will perform a local gradient ascent using the original algorithm.

Our key idea for separating the cloud is through the vision of the agents, if the agents "saw" only part of the agents they wouldn't gather to one center of mass. We understood that those gathering should be dependent on the number of fields and that is where we got the idea to limit the vision in reverse proportion to the current point field measurement.

It is interesting to see that our solution finds the maximum points faster than the original algorithm, although we start with total random walk. The reason is that the random walk is less constraint to the center of mass and the first time the agents become constraint to any center of mass is near a maximum point (when their vision is broaden), that save a great deal of time in finding all the maximum points. If we don't limit ourselves to 100 percent unconstrained agents we can even get a faster result, as we can see in our experiments, but this requires further studying(from which threshold etc.).

Another interesting difference from the original algorithm is that while the original algorithm was globally responsive our algorithm is locally responsive. This is one interesting case that even though we use cheaper agents (the maximum vision range we need on the agent is not more than

half the width of the Gaussian bell width, Mew) the solution is faster and more practical.

It is important to point out that our solution deals mainly with a 2 dimensional problem. On a higher dimension we might have problems in the algorithm due to the random walk and the spread of the agents.

We showed one application to the algorithm but in the real life there could be numerous applications to unintelligent (small and cheap) agents finding a maximum point in a certain field. There could be the chemical detection as presented and the similar nuclear radiation detection problem. In those problems it is reasonable to assume the option of uniform distribution of the agents which improves the accuracy and speed of the solution. Another interesting application might be cancer detection, where we might in the future be able to inject a swarm of nano bots (therefor the size is important) and given the cell density as the scalar field, The agent could find a maximum point which represents a tumor which has a higher cell density and an represent a maximum point.

References

[EB12] Yotam Elor and Alfred M Bruckstein. A "thermodynamic" approach to multi-robot cooperative localization. *Theoretical Computer Science*, 457:59–75, 2012.

[EB14] Yotam Elor and Alfred M Bruckstein. "robot cloud" gradient climbing with point measurements. *Theoretical Computer Science*, 547:90–103, 2014.

[Wil99] Uri Wilensky. Netlogo. 1999.