SAPIENZA
UNIVERSITÀ DI ROMA

Bio-inspired swarming models for decentralized radio access incorporating random links and quantized communications

Paolo Di Lorenzo, Sergio Barbarossa

Ack's: Freedom

Social swarming systems

Social swarming systems are typically made up of a population of simple agents interacting locally with one another and with their environment

The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents

Natural examples include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling

Competition over available resources

Application: Cognitive radio

- System efficiency of infrastructure based networks is typically low
- Efficiency may be improved by allowing cognitive users to sense the radio channels and transmit over unoccupied channels (time slots, subcarriers, …)
- Cognitive (secondary) users are allowed to transmit, provided that the undue interference they induce to licensed (primary) users is tolerable
- Radio access by secondary users depends on their detection capabilities
- Secondary radio network is typically ad hoc (infrastructureless)

Problem: Coordinate the access from the secondary users in a totally decentralized fashion, requiring a minimal coordination among them

Distributed resource allocation by swarming

SAPIENZA
UNIVERSITÀ DI ROMA

Idea: Nearby agents interact with each other to identify spectrum holes and find out resource allocation that prevents collisions and maintains cohesiveness

The search of the most appropriate slot is modeled as the motion of a swarm of agents in the resource domain, looking for forage, representing a function inversely proportional to the interference level

The swarm tends to move in the time-frequency region where there is less interference, while satisfying two basic requirements:

- Minimize the spread in the resource domain
- Avoid collisions between the allocations of different users

Distributed resource allocation by swarming

Resource allocation is modeled as the decentralized minimization of the social foraging swarming potential function

$$J(\boldsymbol{x}) = \sum_{i=1}^{M} \sigma_i(\boldsymbol{x}_i) + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} a_{ij}\left[J_a(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|) - J_r(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|)\right]$$

where

$\sigma_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ represents the interference profile (inverse of the forage term)

$\boldsymbol{x}_i$ denotes the resource vector (e.g., time slot and frequency sub-channel) chosen by node *i*

a_{ij} are the entries of the adjacency matrix of the graph

$J_a(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|)$ and $J_r(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|)$ are the attraction and repulsion potentials

Social foraging swarming algorithm

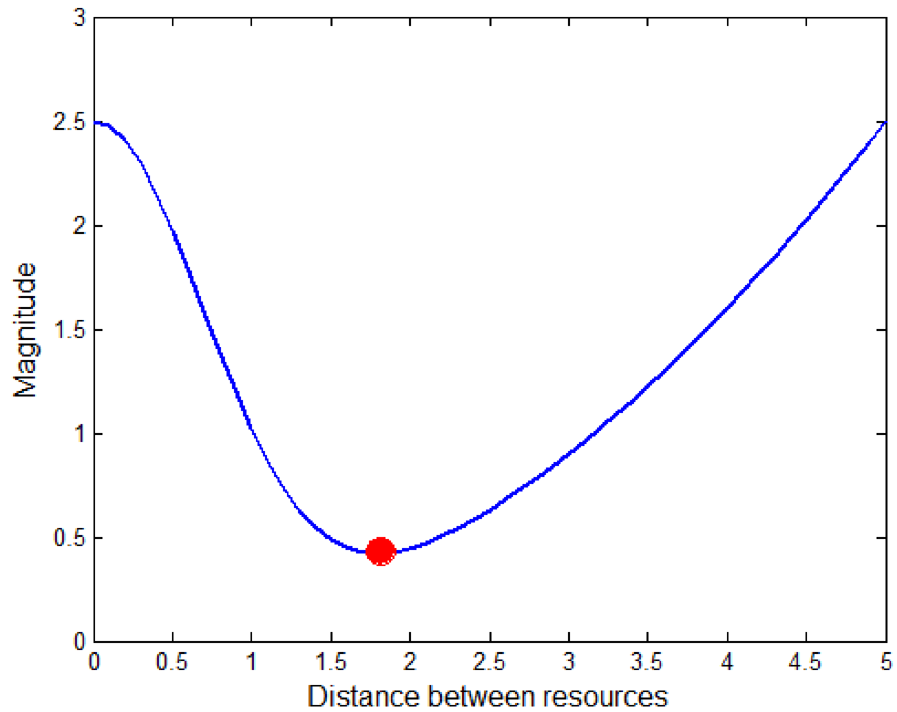
Attraction:

$$J_a(\|y\|) = c_A \|y\|^2 \qquad c_A > 0$$

Bounded Repulsion:

$$J_r(\|y\|) = -c_R \exp(-\|y\|^2) \quad c_R > 0$$

There exist a unique minimum of the attraction /repulsion potential function



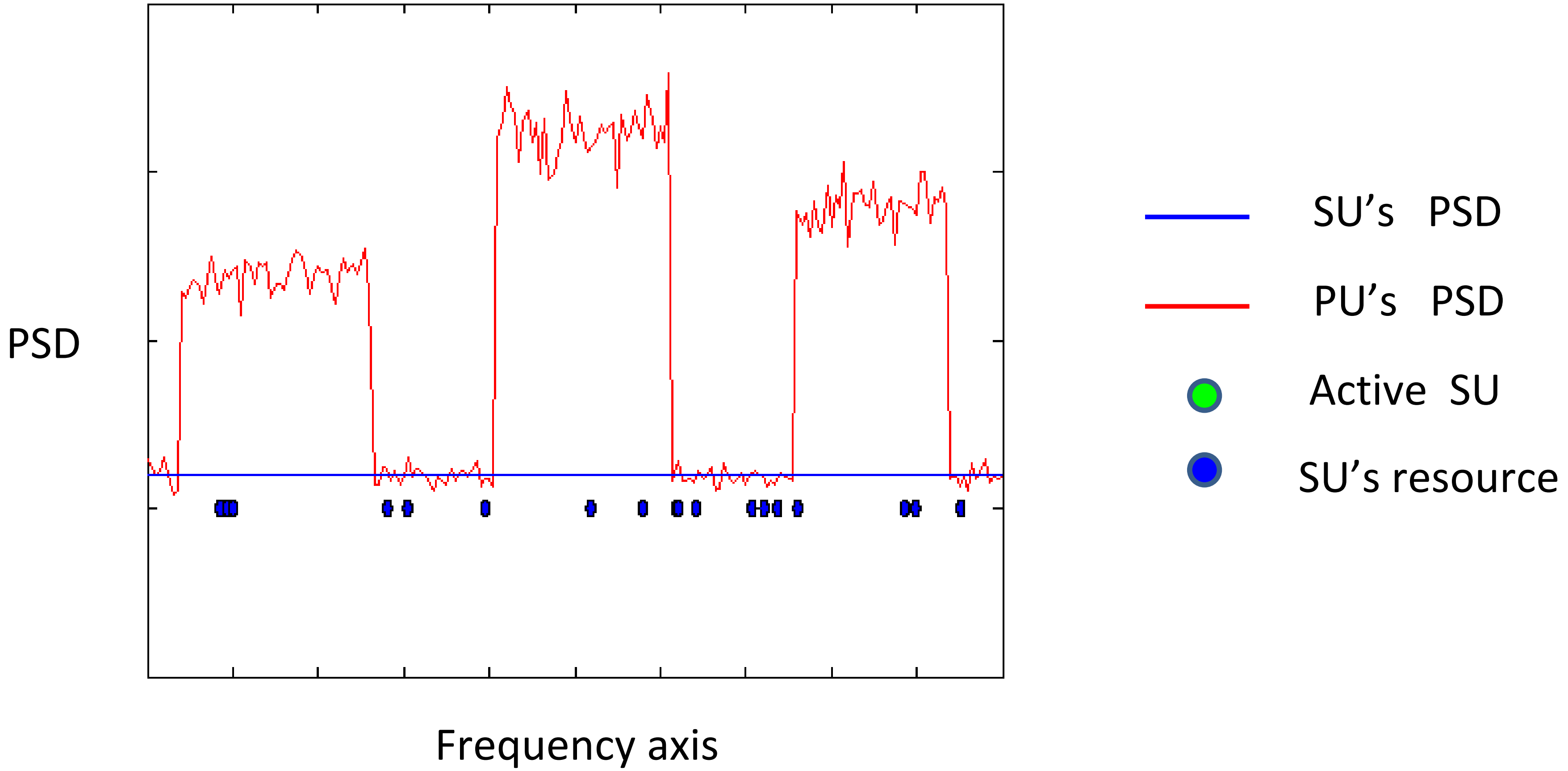
Solution: Steepest descent provides decentralized solution

$$\dot{\boldsymbol{x}}_i(t) = -\nabla_{x_i} J(\boldsymbol{x}(t)) = -\nabla_{x_i} \sigma_i(\boldsymbol{x}_i(t)) + \sum_{j=1}^{M} a_{ij}\, \boldsymbol{g}(\boldsymbol{x}_j(t) - \boldsymbol{x}_i(t)) \qquad i = 1, \ldots, M$$

Each node must estimate only the local gradient and to interact with the nearest nodes

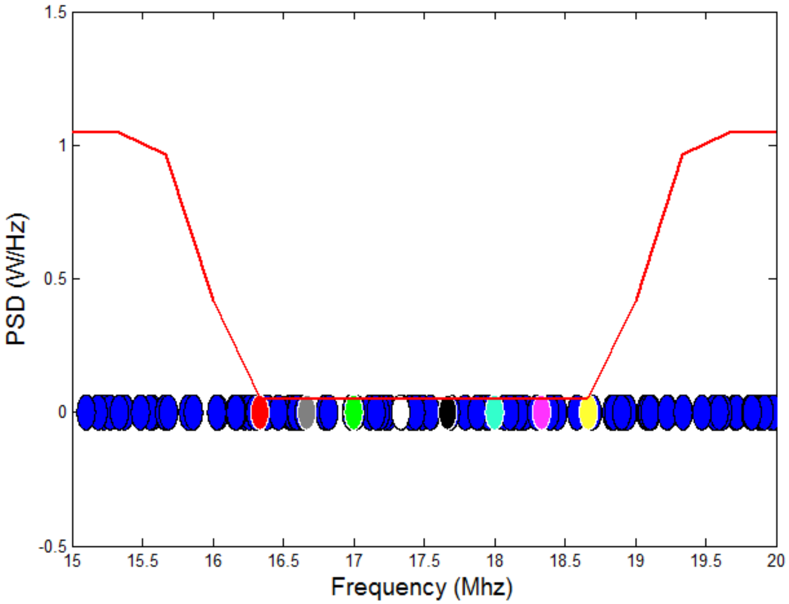
Swarming in the frequency domain

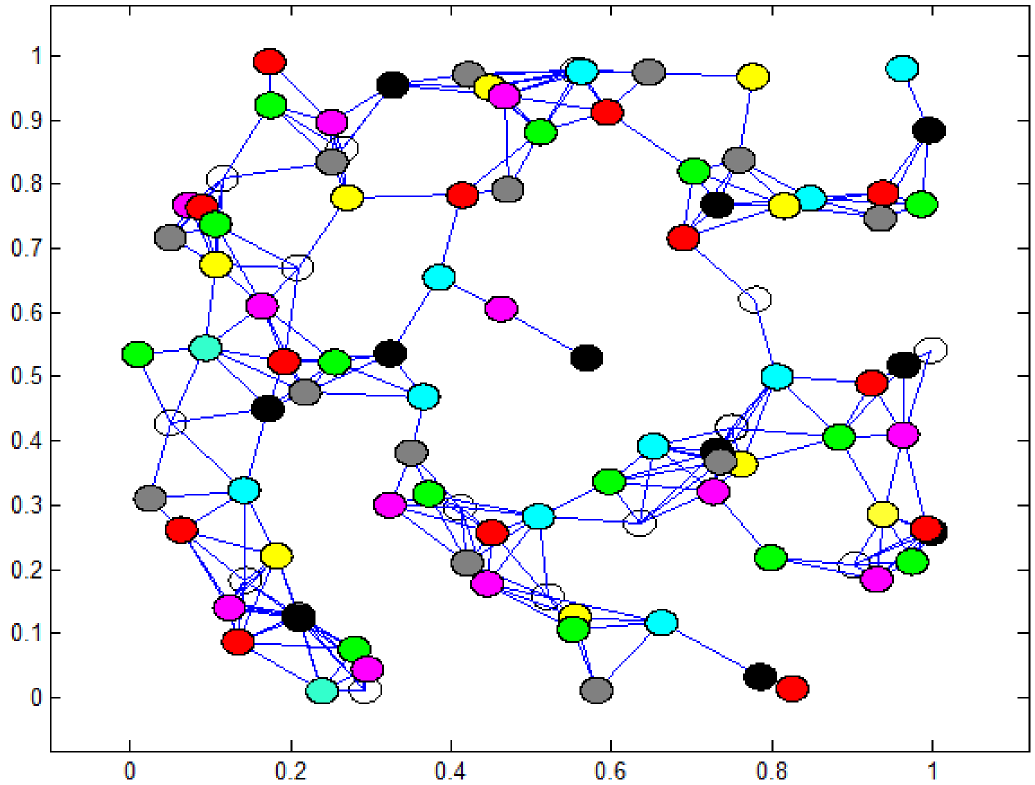
Example: Adaptation to PU activations – swarm against a predator



Swarming in the frequency domain

Example: Spatial reuse of channels





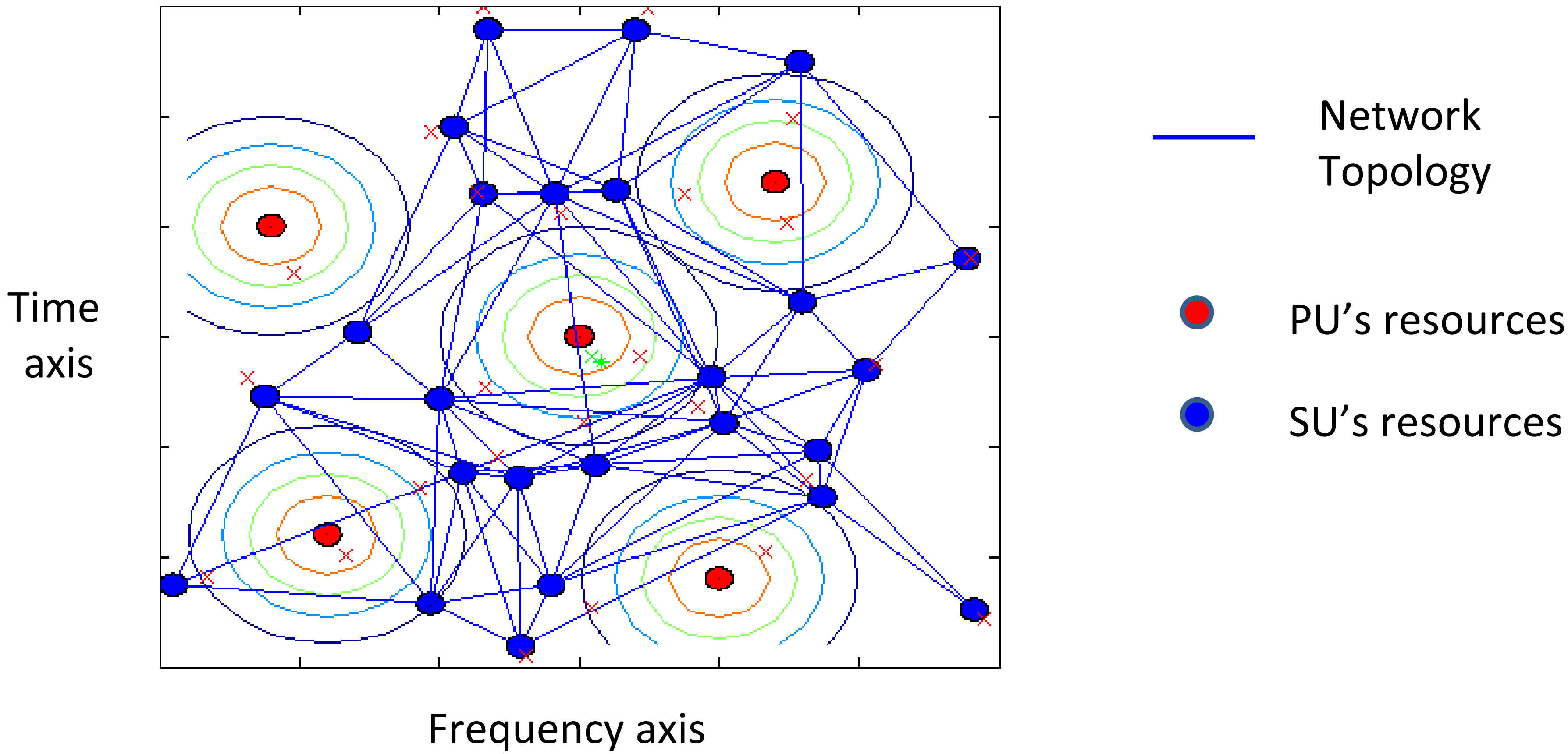
M = 100 nodes

8 available channels

The algorithm is capable of implementing a decentralized mechanism for spatial reuse of channels

Swarming in the time-frequency domain

Example: Dynamic resource allocation in time-frequency domain



Continuous-time Markovian interference

The PU's activity on each subchannel *i* can be modeled as a homogeneous continuous-time Markov process where

$$\boldsymbol{Q}_i = \begin{pmatrix} -\lambda_i & \mu_i \\ \lambda_i & -\mu_i \end{pmatrix}$$ is the transition rate matrix

On each channel, through preliminary estimation, it is known the average power p_i and the transition rates from idle to idle λ_i and from active to active μ_i .

Given a channel *i* sensend as idle at time t=0, the average power at time t is

$$p_i(t; \text{idle at } 0) = \frac{\lambda_i p_i}{\lambda_i + \mu_i}\left(1 - e^{-(\lambda_i+\mu_i)t}\right)$$

Given a channel *i* sensend as busy at time t=0, the average power at time t is

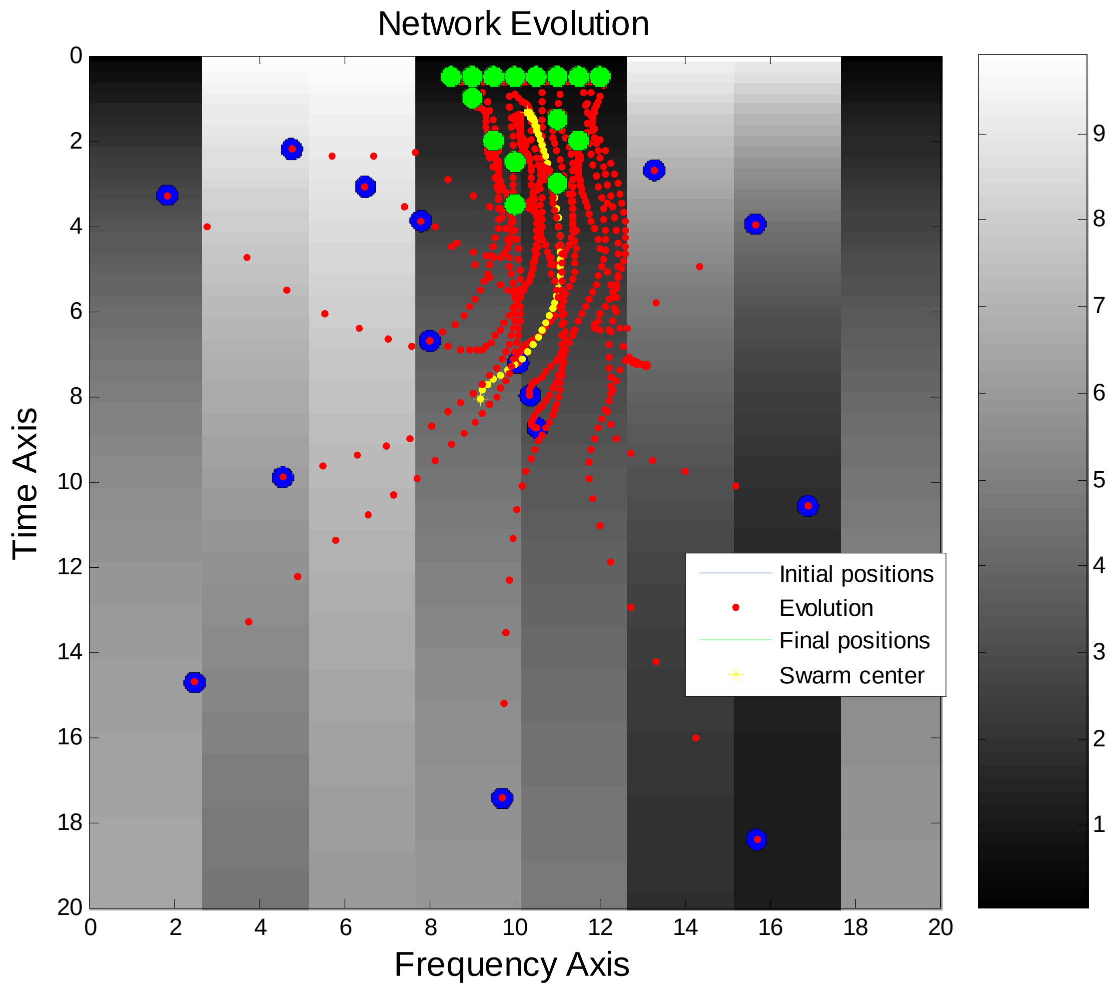
$$p_i(t; \text{active at } 0) = p_i - \frac{\mu_i p_i}{\lambda_i + \mu_i}\left(1 - e^{-(\lambda_i+\mu_i)t}\right)$$

Time-frequency allocation

Example: Resource allocation in time-frequency domain with Markovian interference

The primary users activity is modeled as a set of continuous time Markov processes

The resources find a final allocation close in time on an initially idle spectral band



The effect of realistic channels

Sources of randomness:

1. spatial positions
2. channel fading
3. quantization and channel noise

Basic question:

Can the swarming algorithm achieve the solution even in the presence of random interactions among the nodes ?

The effect of realistic channels

Random link failure model: At any time *k*, the graph Laplacians can be written as

$$\boldsymbol{L}[k] = \bar{\boldsymbol{L}} + \tilde{\boldsymbol{L}}[k]$$

where $\bar{\boldsymbol{L}} = E\{\boldsymbol{L}[k]\}$ is the expected graph and $\tilde{\boldsymbol{L}}[k]$ is a sequence of zero mean, independent identically distributed (i.i.d.) Laplacian matrices

Quantization: we assume that each inter-node communication channel uses a uniform quantizer, which is defined by the following vector mapping

$$\boldsymbol{q}(\boldsymbol{y}) = [b_1\Delta, \ldots, b_n\Delta]^T = \boldsymbol{y} + \boldsymbol{e}(\boldsymbol{y})$$

where $\boldsymbol{y}$ is the channel input, Δ is the quantization step and $\boldsymbol{e}(\boldsymbol{y})$ is the error

Conditioned on the input, the quantization error is deterministic

➡ It can influence the convergence of the algorithm

To avoid these effects, we consider dithered quantization, which endows the quantization error with some useful statistical properties

Swarming over realistic channels 1/2

The discrete time version of the swarming algorithm in presence of random link failures and dithered quantization noise can be written as

$$\boldsymbol{x}_i[k+1] = \boldsymbol{x}_i[k] + \alpha[k]\Big[-\nabla_{\boldsymbol{x}_i[k]}\sigma(\boldsymbol{x}_i[k]) + \sum_{j=1}^{M} \underline{a_{ij}[k]}\, \boldsymbol{g}(\boldsymbol{x}_j[k] - \boldsymbol{x}_i[k] + \underline{\nu[k] + \epsilon[k]})\Big]$$

$$i = 1, \ldots, M$$

The dynamic of the overall system can be expressed in compact form as

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{x}[k] + \alpha[k]\big[-\boldsymbol{\Sigma}^{\nabla}(\boldsymbol{x}[k]) - (\boldsymbol{L}\boldsymbol{x}[k] \otimes \boldsymbol{I}_n)\boldsymbol{x}[k] + \boldsymbol{\Upsilon}\boldsymbol{x}[k] + \boldsymbol{\Psi}\boldsymbol{x}[k]\big] \\ &= \boldsymbol{x}[k] + \alpha[k]\big[\boldsymbol{R}(\boldsymbol{x}[k]) + \boldsymbol{\Gamma}(k, \boldsymbol{x}[k], \omega)\big] \end{aligned}$$

where

$$\boldsymbol{R}(\boldsymbol{x}[k]) = -\boldsymbol{\Sigma}^{\nabla}(\boldsymbol{x}[k]) - (\bar{\boldsymbol{L}}_{\boldsymbol{x}[k]} \otimes \boldsymbol{I}_n)\boldsymbol{x}[k]$$ Deterministic function

$$\boldsymbol{\Gamma}(k, \boldsymbol{x}[k], \omega) = -(\tilde{\boldsymbol{L}}_{\boldsymbol{x}[k]} \otimes \boldsymbol{I}_n)\boldsymbol{x}[k] + \boldsymbol{\Upsilon}_{\boldsymbol{x}[k]} + \boldsymbol{\Psi}_{\boldsymbol{x}[k]}$$ Random function

Swarming over realistic channels 2/2

The swarming algorithm is equivalent to a Robbins-Monro stochastic approximation procedure attempting to find the zeros of a deterministic function $\boldsymbol{R}(\boldsymbol{x})$, whose value, measurable at each time instant, is corrupted by the random disturbance $\boldsymbol{\Gamma}(\boldsymbol{x}, \omega)$

Convergence condition on the step size:

$$\alpha[k] > 0, \quad \sum_{k=0}^{\infty} \alpha[k] = \infty, \quad \sum_{k=0}^{\infty} \alpha^2[k] < \infty \quad \Rightarrow \quad \alpha(k) = \frac{\alpha_0}{(1+k)^{\tau}}, \quad 0.5 < \tau \leq 1$$

Then,

Almost Sure convergence

$$\mathbb{P}\left[\lim_{k\to\infty} \rho(\boldsymbol{x}[k], B) = 0\right] = 1$$

B is the solution set

$\rho(\cdot)$ is the Euclidean norm

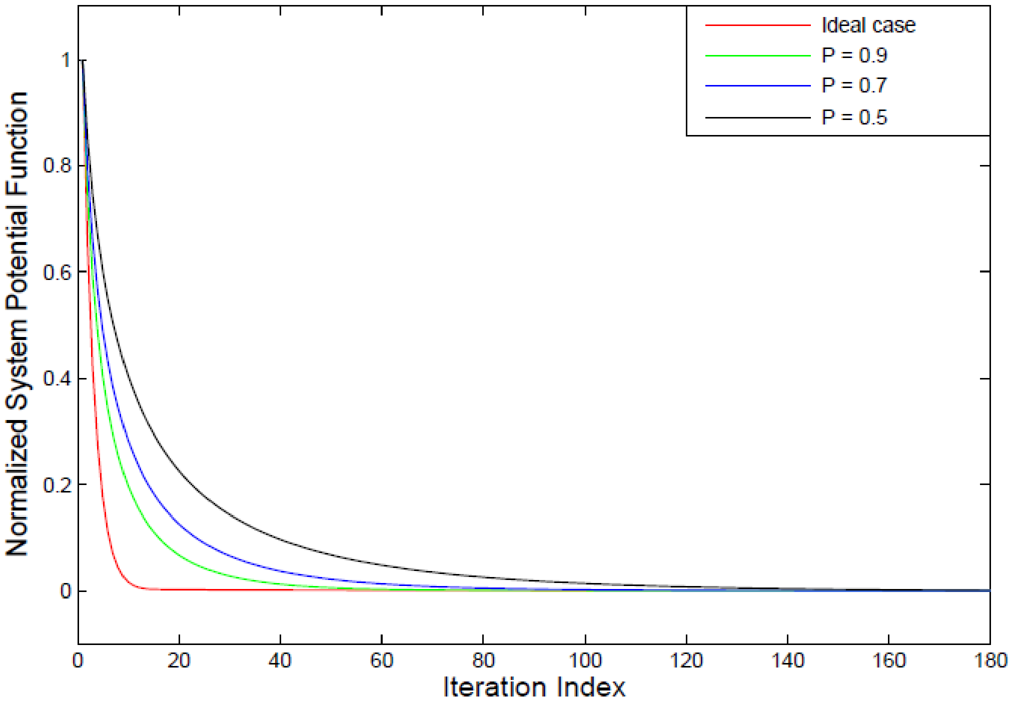
The effect of realistic channels

Example: Swarming in frequency domain in the presence of link failures and noise

In a realistic communication scenario, some packets may be received with errors because of channel fading or noise

→ the erroneous packets are dropped

The swarm reaches an equilibrium configuration dependent on the expected graph of the network



Reducing the probability to establish a communication link, the network requires a longer time to reach the final equilibrium state

Conclusions

We proposed a distributed resource allocation strategy for the access of opportunistic users in cognitive networks based on a swarm model mimicking the foraging behaviors of a swarm

Attraction and repulsion forces can help, in the application at hand, to identify suitable resource slots, while avoiding conflicts among SU's

Each node is supposed to be able to listen only to nearby nodes, producing an intrinsic capability of the system to provide spatial reuse of the channels, through a purely decentralized mechanism

The method is robust against packet drops due to random link failures, whose effect is to slow down the convergence process

Thank you