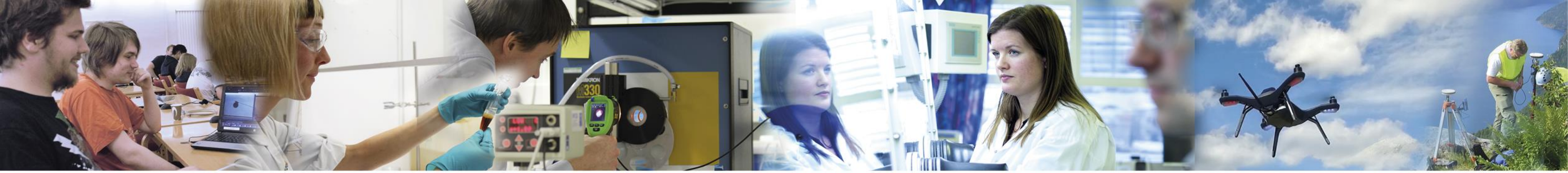


UiO : Department of Technology Systems
University of Oslo

TEK5010/9010 - Multiagent systems 2021
Lecture 4

Swarm robotics 1

Jonas Moen

Highlights lecture 4 – Swarm robotics 1*

- What is swarm robotics?
- Swarm performance
- Scenarios of swarm robotics
- Modelling approaches
  - Local sampling
  - Time-space dynamics
  - Fokker-Planck micro-macro link
- Formal design methods in swarm systems

*Hamann, 2018: chapter 1, 4 and 5

What is swarm robotics?

«Swarm robotics is the study of how to make robots collaborate and collectively solve a task, that would otherwise be impossible to solve by a single individual of these robots» [Hamann, 2018].

A way of handling system complexity inspired by natural swarm systems.

⇒ Swarm intelligence applied to robotics

How big is a swarm?

«Not as large as to be dealt with as statistical averages» and «not as small as to be dealt with as a few-body problem», [Beni, 2005].

$$10^2 < N \ll 10^{23}$$

(3-body Newtonian mechanics) (Avogadros number)

How big is a swarm?

Alternative approach:

«A swarm is not necessarily defined by its size but rather by its behavior.» [Hamann, 2018]

SI observables like self-organization, emergence and distributed control among simple autonomous agents.

Fault tolerant, flexible and scalable system design.

Swarm performance

«The average performance of a robot swarm depends on the density or the swarm size if the area in which the swarm operates is kept constant.» [Harmann, 2018]

Readily visualized by the example of a bucket brigade, [Anderson et *al*., 2002]

Bucket brigade

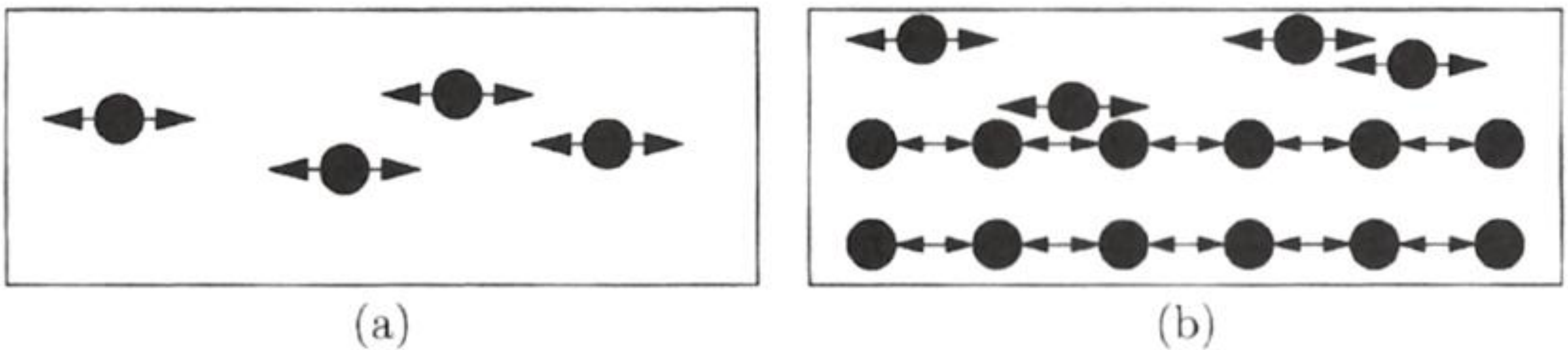


Image: Figure 1.5 a and b, Hamann, 2018

Bucket brigade

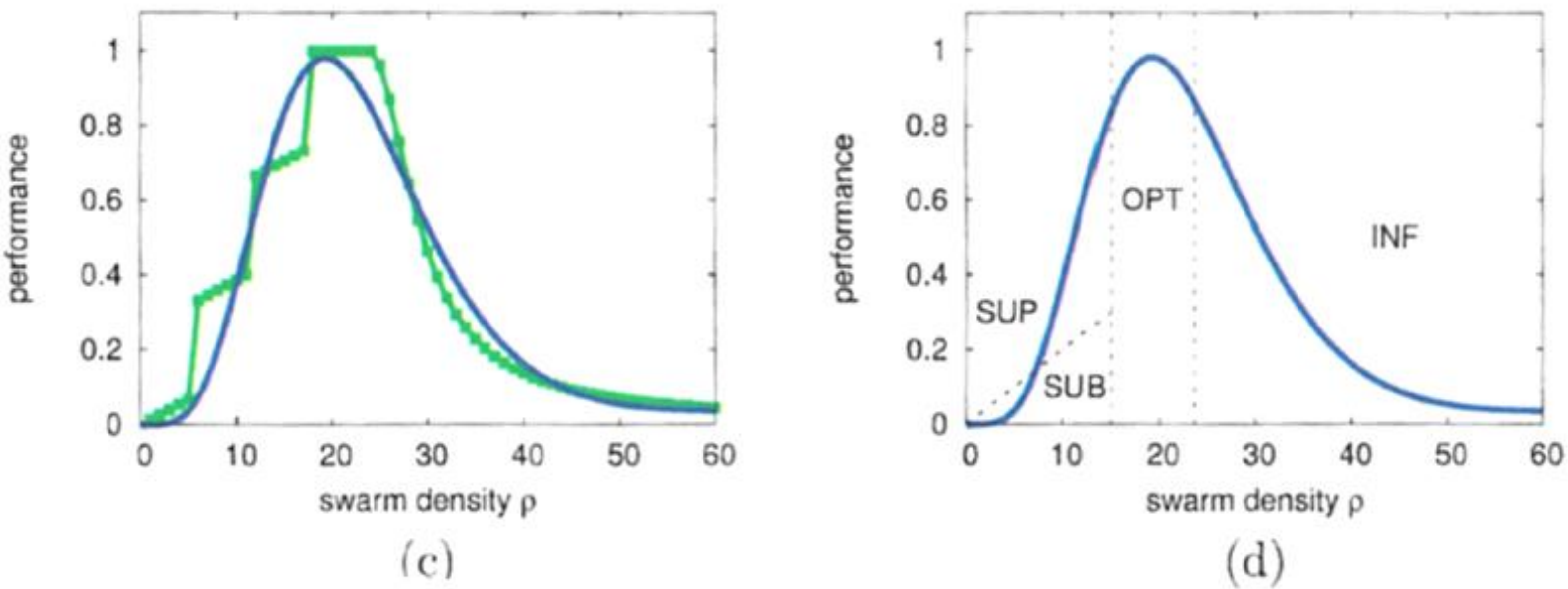


Image: Figure 1.5 c and d, Hamann, 2018

Swarm performance

Regions of performance:

1. Super-linear region
2. Sub-linear region
3. Optimal region
4. Inference region

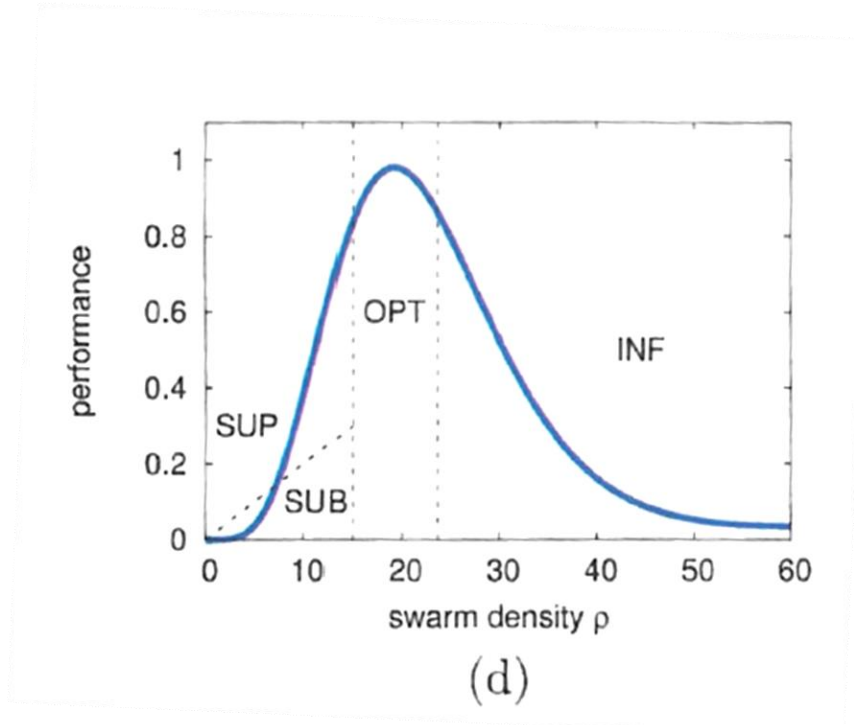


Image: Figure 1.5 d, Hamann, 2018

Swarm performance

Two processes affect performance:

1. Contention or inference by sharing limited resources
2. Lack of coherency in the distributed data due to local communication

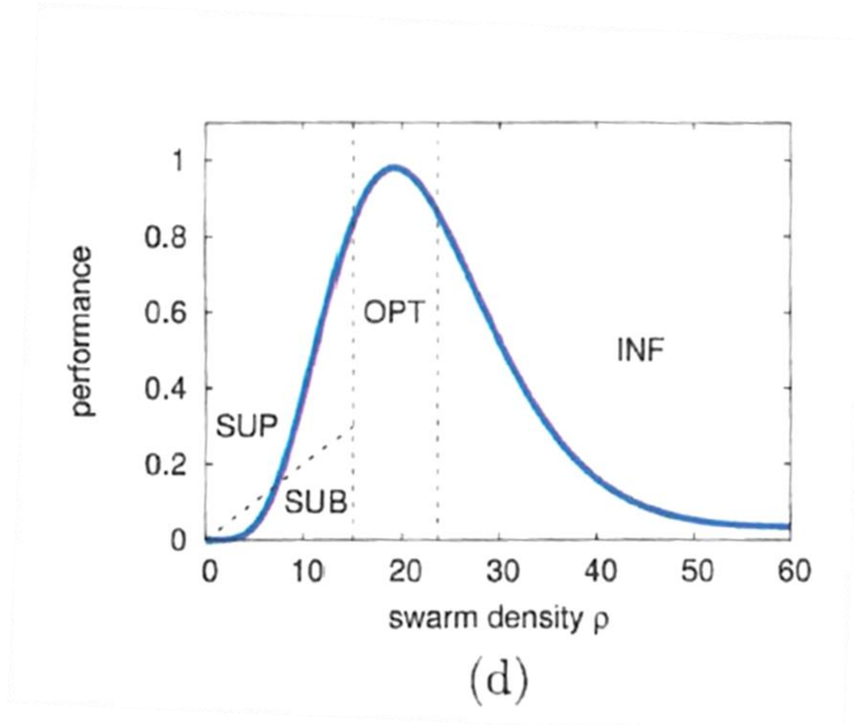


Image: Figure 1.5 d, Hamann, 2018

Swarm performance

The Universal Scalability Law [Gunther, 1993] of parallel processing systems:

$$R(N) = c\,\frac{N}{1+\alpha(N-1)+\beta N(N-1)}$$

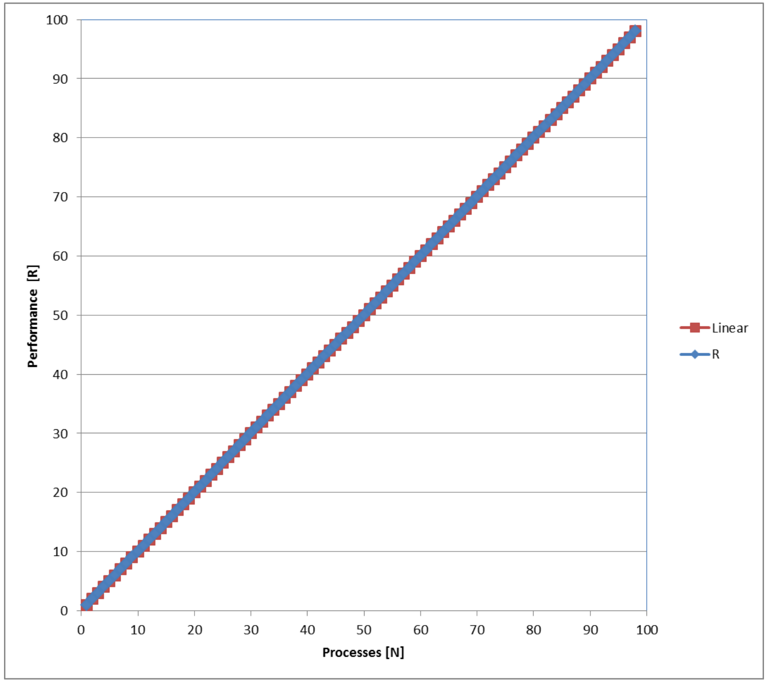
where $R(N)$ is performance as a function of N processors

α is degree of contention (inference)

β gives the lack of coherency in the distributed data

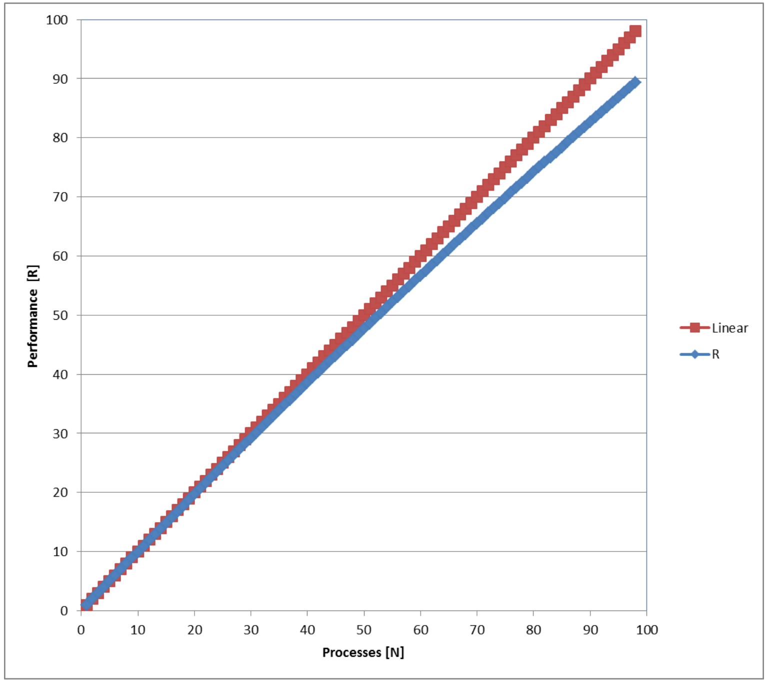
C is a scalar

The Universal Scalability Law [Gunther, 1993]



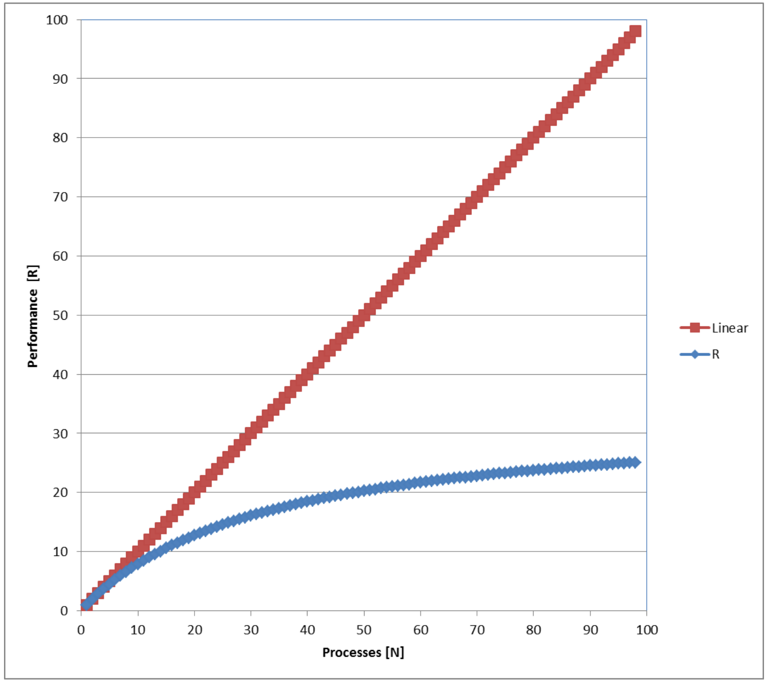
Linear speed up: $\alpha = 0, \beta = 0$

The Universal Scalability Law [Gunther, 1993]



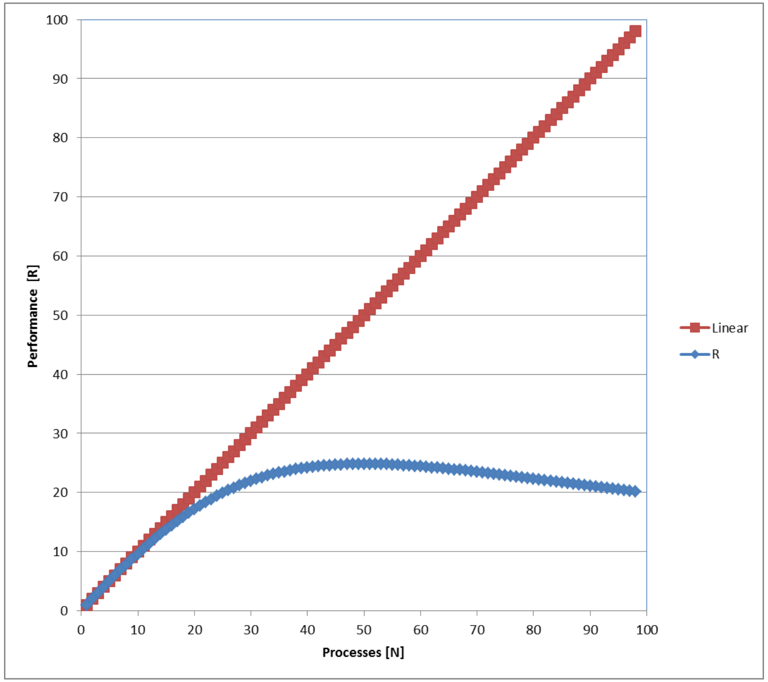
Sub-linear speed up: $\alpha = 0.001, \beta = 0$

The Universal Scalability Law [Gunther, 1993]



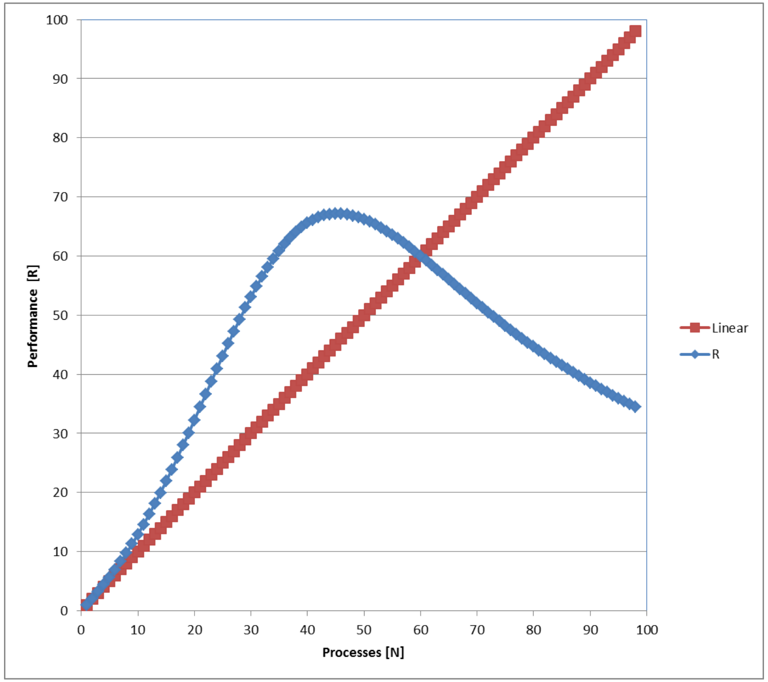
Speed up levels off: $\alpha = 0.03, \beta = 0$

The Universal Scalability Law [Gunther, 1993]



Decrease: $\alpha = 7 \times 10^{-4}, \beta = 3 \times 10^{-4}$

The Universal Scalability Law [Gunther, 1993]



Super-linear speed up: $\alpha = -3 \times 10^{-2}, \beta = 5 \times 10^{-4}$

Empirical support

[Anderson et *al*., 2002] found that bucket brigades occure in a number of ant species.

Also, increasing individual performance with increasing swarm size was observed in wasps [Jeanne & Nordheim, 1996].

Scenarios of swarm robotics

- Aggregation and clustering
- Dispersion
- Self-assembled pattern formation, object clustering and sorting
- Collective transport, manipulation and motion
- Shepherding
- Bio-hybrid systems
- Swarm robotics 2.0

Aggregation and clustering

In aggregation the task for the robots is to position themselves close to each other in one spot. Position may be specified or unspecified guided by collective decision-making

Example: cluster at the warmest, brightest or most radioactive spot.

Hard control problem due to the balance between exploration and exploitation.

Dispersion

A robot is supposed to position itself as far as possible from every other robot while staying in contact.

Example: large area monitoring and surveillance or minimal robot density for conservation of system resources.

Could give raise to clumped, random or uniform distribution of swarm depending on algorithm and underlying task.

Self-assembled pattern formation

Robots aggregate in defined shapes or shape their environment.

Example: emergent traffic flow of pedestrians, colourful animal patterns, clustering and sorting of resources, self-assembly of small robots into larger constructs like bridges and tools.

The self-assembly process should be robust and adaptable to dynamic environments.

Collective transport

Robots aggregate to transport resources not movable by single robots, i.e. direct cooperation.

Example: factory assembly lines (e.g. box-pushing) or collective transport of materials at building sites.

Collective manipulation

Robots cooperate to solve manipulation tasks, i.e. specialized direct cooperation.

Example: drill and mining operations.

In [Ijspeert et *al*., 2001] the authors investigates ‘stick pulling’ in terms of division of labour. Super-linear speed up is observed.

Other swarm applications

Shepherding of animals

Bio-hybrid systems for pest control

Flocking, i.e. PSO

Foraging, i.e. ACO

Task allocation, i.e. response thresholds and auctions

Swarm robotics 2.0

Swarm systems are now moving out of the labs and into the fields:

- Error detection and security
- Interfacing robots and robots as interfaces
- Swarm robotics as field robotics

Formal design methods in swarm systems

Modelling is introduced as a dimension reducing technique for understanding the relevant relations in swarm robotics

- Swarm model description and notation
- Local sampling
- Space-time dynamics using rate/differential equations
- Fokker-Planck equation for the micro-macro link
- Network models
- Formal design methods

Swarm model description and notation

«Formalization means to describe the modelled system by a formal system that allows to derive new insight by means of logic and mathematics.» [Hamann, 2018].

We strive to achieve abstraction and simplification.

Swarm model description and notation

A swarm system of size N in 2D space can be described by the state vector:

$$\boldsymbol{\gamma} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N, \boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_N, s_1, s_2, \dots, s_N)$$

where $i \in \{1,2,\dots,N\}$ is the index of N agents

$\boldsymbol{x_i} = (x_{i1}, x_{i2})$ is the 2D position of agent i

$\boldsymbol{v_i} = (v_{i1}, v_{i2})$ is the velocity of agent i

s_i is the discrete state of agent i

Swarm model description and notation

A swarm system of size N in 2D space can be described by the state vector:

$$\boldsymbol{\gamma} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N, \boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_N, s_1, s_2, \dots, s_N)$$

This system has a configuration space of $\boldsymbol{\gamma} \in \Gamma$

$$\dim(\Gamma) = 2N + 2N + N = 5N$$

Swarm model description and notation

In a swarm system of size N=1000 there is not many good ways of understanding such a huge system except for running direct simulations.

$$\dim(\Gamma) = 5N = 5000$$

The simulation tracks and updates each variable in each time step $\boldsymbol{\gamma}_t, \boldsymbol{\gamma}_{t+1}, \boldsymbol{\gamma}_{t+2}$, however we can only observe one configuration at a time starting from a specific initialization $\boldsymbol{\gamma}_0$.

Swarm model description and notation

We need to reduce the configuration space but at the same time retain the relevant swarm system dynamics:

$$f : \Gamma \to \varphi$$

where $\varphi \in \Phi$ and $\dim(\varphi) \ll \dim(\Gamma)$

Swarm model description and notation

Modelling represented as a series of mappings:

$$\begin{array}{ccc} \boldsymbol{\gamma}_t & \xrightarrow{g} & \boldsymbol{\gamma}_{t+1} \\ f\downarrow & & \downarrow f \\ \boldsymbol{\varphi}_t & \xrightarrow[h]{} & \boldsymbol{\varphi}_{t+1} \end{array}$$

$$g: \Gamma \rightarrow \Gamma \text{ giving } g(\boldsymbol{\gamma}_t) = \boldsymbol{\gamma}_{t+1}$$

$$h: \Phi \rightarrow \Phi \text{ giving } h(\boldsymbol{\varphi}_t) = \boldsymbol{\varphi}_{t+1}$$

Swarm model description and notation

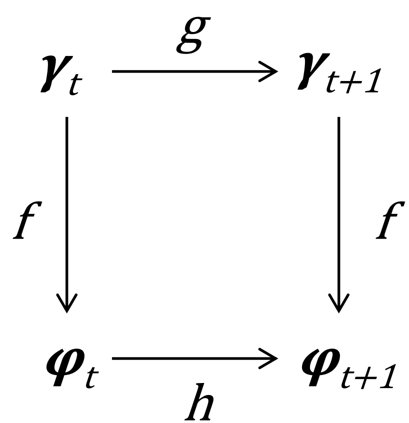
Modelling represented as a series of mappings:

$$
\begin{array}{ccc}
\boldsymbol{\gamma}_t & \xrightarrow{\;g\;} & \boldsymbol{\gamma}_{t+1} \\
{\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f} \\
\boldsymbol{\varphi}_t & \xrightarrow[\;h\;]{} & \boldsymbol{\varphi}_{t+1}
\end{array}
$$

we want $f: \Gamma \rightarrow \varphi$ giving $h\big(f(\boldsymbol{\gamma}_t)\big) = f\big(g(\boldsymbol{\gamma}_t)\big)$

Swarm model description and notation

Also note that we could have $f^{-1}: \varphi \rightarrow \Gamma$



but this is difficult/impossible due to the reduction of dimensions when going from $f: \Gamma \rightarrow \varphi$

Swarm model description and notation

An extreme example of dimension reduction:

Assume a binary collective decision-making scenario of option A or B with N=1000 agents.

$$f(\boldsymbol{\gamma}) = \frac{|\{s_i | s_i = A\}|}{N} = \varphi$$

where $\dim(\varphi) = 1$, but how does $h(\boldsymbol{\varphi}_t) = \boldsymbol{\varphi}_{t+1}$ look like?

Local sampling

Concepts for local sampling in swarm systems are important because of robots local perception of the world and their typically reduced communication abilities.

In essence, local sampling is to infer the ‘global picture’ from a set of samples, i.e. the representativeness of the samples are important in order to avoid bias.

Local sampling

Typical problems of local sampling in swarms:

1. Clustering not representable of swarm system
2. Local robots could be correlated
3. Low density could give strong bias

Local sampling

Estimation of swarm binary state:

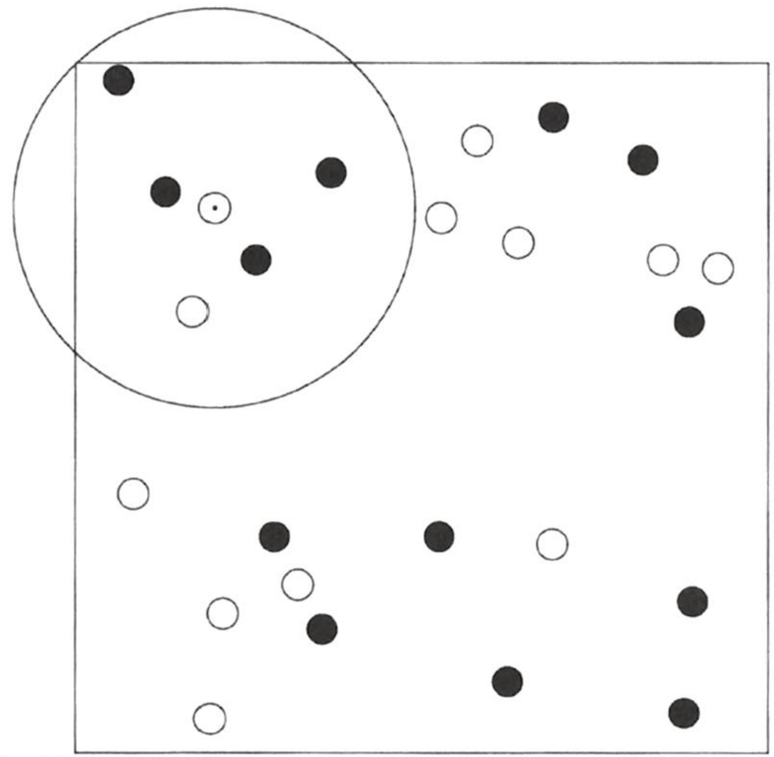


Image: Figure 5.2, Hamann, 2018

Local sampling

Estimation of swarm area [Mallon and Franks, 2000]

Inspiration taken from:

1. The ant species *Leptothorax albipennies* search for a new nest site
2. Buffon’s needle

Local sampling

Estimation of swarm area
[Mallon and Franks, 2000]

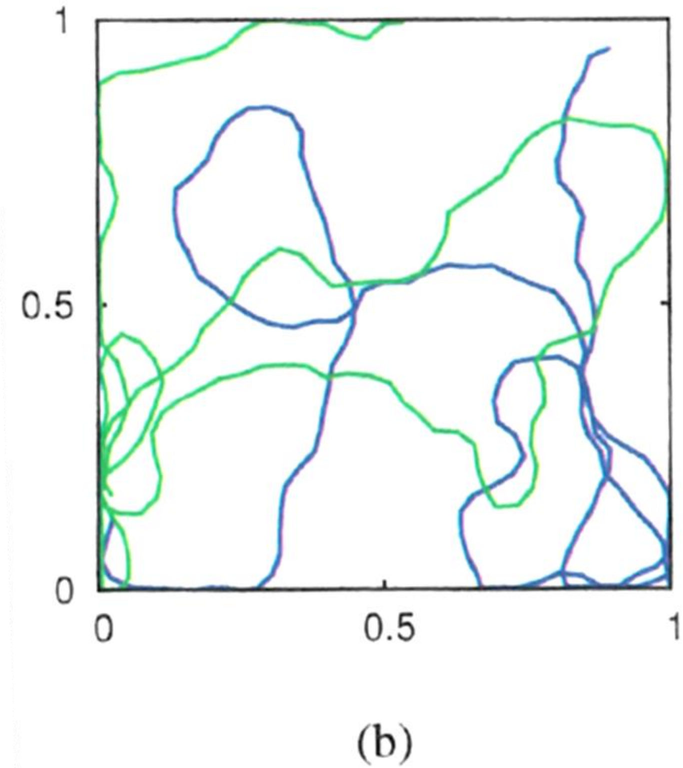


Image: Figure 5.4b, Hamann, 2018

Local sampling

Estimation of swarm area [Mallon and Franks, 2000]

$$Area = \frac{2L_1L_2}{n\pi}$$

where L_1 is pheromone path, a random walk from A to A

L_2 is path without pheromones, also A to A

n is number of intersections of lines L_1 and L_2

Local sampling

Buffon's needle
[Ramaley, 1969]
(originally estimation of π)

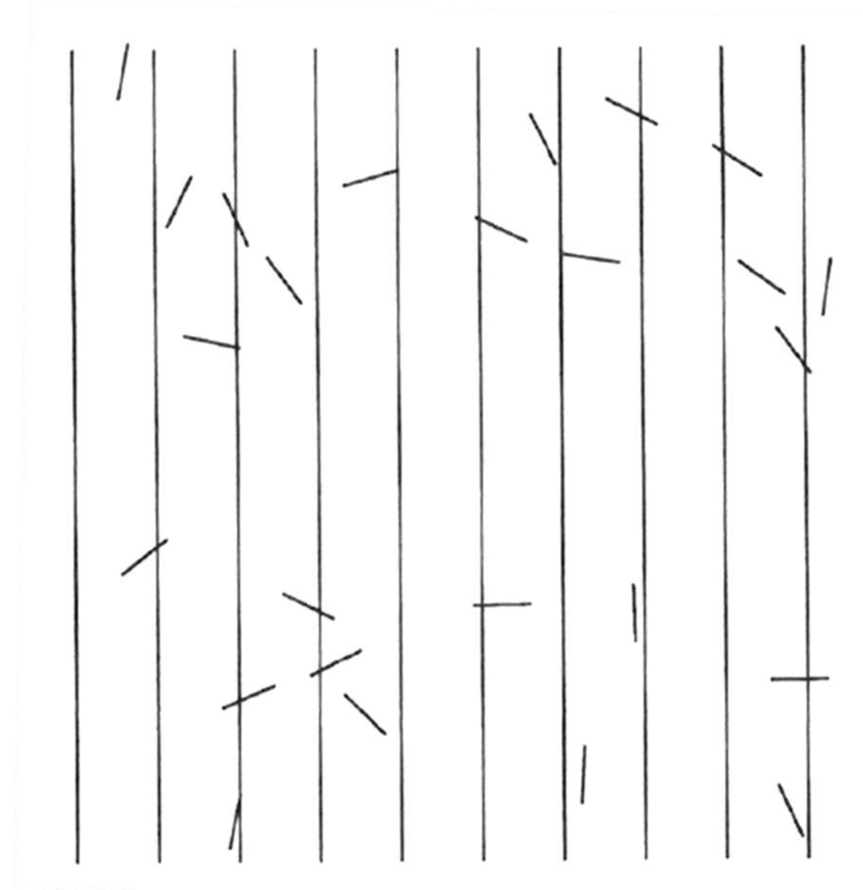


Image: Figure 5.3, Hamann, 2018

Local sampling

Buffon's needle

$$P = \frac{2b}{s\pi}$$

where P is probability of needle intersecting a line

b is needle length

s is distance between the parallel straight lines $s > b$

Local sampling

Bounds on Buffon's needle for 95% confidence

$$\hat{P} \pm 1.96\sqrt{\frac{1}{n}\hat{P}\left(1-\hat{P}\right)}$$

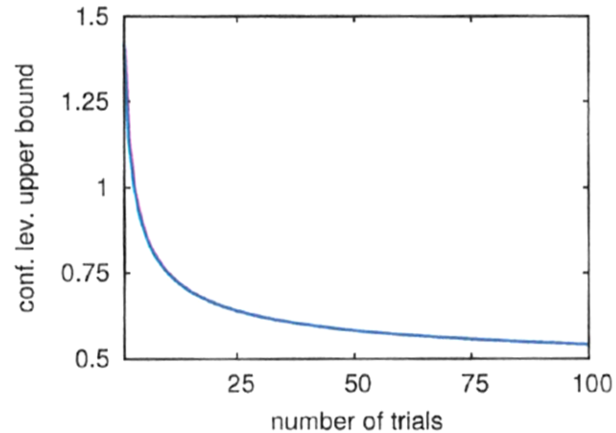
where $\hat{P}$ is estimated probability of intersection

n is number of samples, i.e. needles

1.96 is a scalar related to the required percentile

Local sampling

Bounds on Buffon's needle for 95% confidence



(a)

Good estimates can be achieved with few samples.

Image: Figure 5.4 a, Hamann, 2018

Modelling approaches

«In swarm robotics we are still on the search for an appropriate general modelling technique.» [Hamann, 2018]

- Rate equation for time dynamics
- Differential equations for spatial problems
- Fokker-Planck equation for micro-macro link
- Network models

Modelling time dynamics

Ordinary differential equations (ODE) rate equations (inspiration from chemistry)

$$\frac{dC}{dt} = kAB$$

where $A + B \rightarrow C$ is the concentration between A, B and C

k is the rate coefficient

Modelling time dynamics

Swarm system collecting pucks [Lerman & Galstyan, 2002], two states available search or avoid:

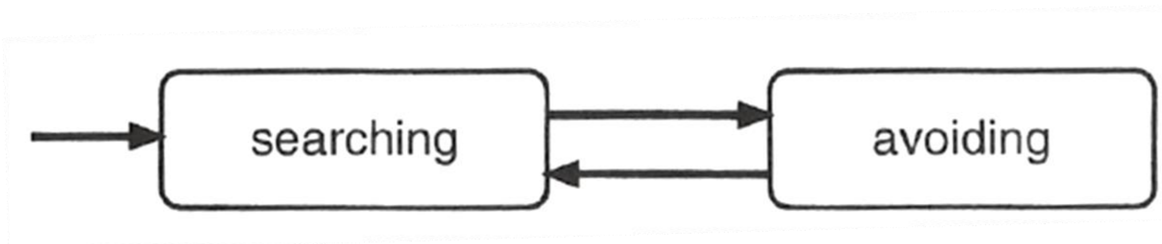


Image: Figure 5.5, Hamann, 2018

Modelling time dynamics

Swarm system collecting pucks [Lerman & Galstyan, 2002], two states available search or avoid:

$$\frac{dn_s}{dt} = -\alpha_r n_s(n_s - 1) + \alpha_r n_s(t - \tau)(n_s(t - \tau) + 1)$$

where n_s is the probability of an agent is in search modus

α_r is the rate coefficient of detecting other robots

τ is the time spent in avoid modus

Modelling time dynamics

Swarm system collecting pucks [Lerman & Galstyan, 2002], two states available search or avoid:

$$\frac{dm}{dt} = -\alpha_p n_s m$$

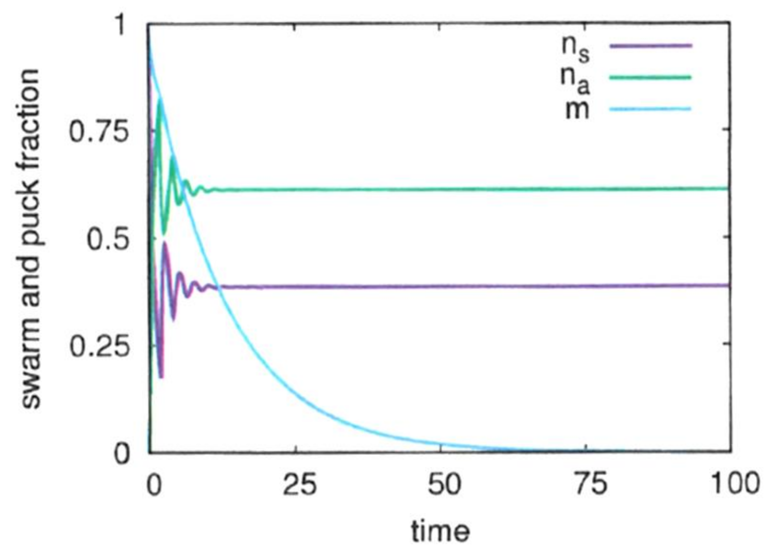
where m is fraction of uncollected pucks

α_p is the rate coefficient of detecting a puck

n_s is the probability of an agent is in search modus

Modelling time dynamics

Swarm system collecting pucks [Lerman & Galstyan, 2002], two states available search or avoid:



This is a delay differential equation solved numerically

Image: Figure 5.6, Hamann, 2018

Modelling space dynamics

Partial differential equations (PDE) allow for more detailed spatial modelling (compared to grid-world models).

Model of stochastic motion of one agent with drift:

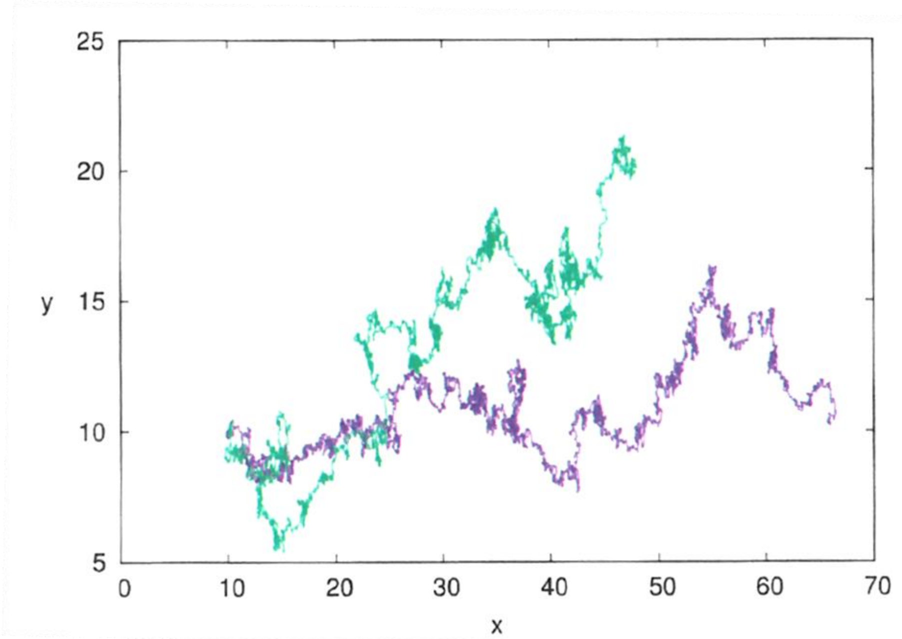
$$\dot{\boldsymbol{R}}(t) = \boldsymbol{F}_t + \boldsymbol{C}$$

where $\boldsymbol{R} = (r_x, r_y)$ is the position of an agent in 2D space

$\boldsymbol{F}_t = (X_t, Y_t)$is the stochastic term of random motion

$\boldsymbol{C} = (c_x, c_y)$ is the non-stochastic drift term

Modelling space dynamics



$$c_x = 0.1$$

Image: Figure 5.8, Hamann, 2018

Modelling space dynamics

The Langevin equation:

$$\dot{\boldsymbol{R}}(t) = \underbrace{\boldsymbol{A}(\boldsymbol{R}(t), t)}_{\substack{\text{Non-stochastic} \\ \text{drift term}}} + \underbrace{B(\boldsymbol{R}(t), t)\boldsymbol{F}(t)}_{\substack{\text{Stochastic} \\ \text{random term}}}$$

where $\boldsymbol{R}(t)$ is the position of an agent in 2D space

$\boldsymbol{F}(t)$ is the stochastic term of random perturbation

$\boldsymbol{A}$ is the non-stochastic drift term

B is a scalar for the stochastic term

Modelling space dynamics

The Langevin equation, typically:

$$\boldsymbol{A}(\boldsymbol{R}(t), t) = \nabla P(\boldsymbol{R}(t), t)$$

where $\nabla P(\boldsymbol{R}(t), t)$ is a gradient in a potential field P

Modelling space dynamics

The Fokker-Planck equation:

- The Fokker-Planck equation is the macroscopically corresponding piece to the microscopic approach described by the Langevin equation.
- Originally used in physics for modelling Brownian motion with drift, describing diffusion processes in thermodynamics. [Fokker, 1914; Planck, 1917]

Modelling space dynamics

The Fokker-Planck equation:

- The combination of the Fokker-Planck equation with the Langevin equation establish a direct mathematical micro-macro link.
- The complicated mathematical modelling shows the general challenge of creating explicit micro-macro links in complex systems.

Modelling space dynamics

The Fokker-Planck equation:

Derived from the Langevin equation assuming

- System noise F is white noise (gaussian with zero mean)
- High enough density of particles/agents interacting
- etc

$\Rightarrow$ Only valid for special and idealistic cases

Modelling space dynamics

The Fokker-Planck equation:

$$\frac{\partial \rho(\boldsymbol{r},t)}{\partial t} = \underbrace{-\nabla\big(\boldsymbol{A}(\boldsymbol{r},t)\rho(\boldsymbol{r},t)\big)}_{\text{Non-stochastic drift term}} + \underbrace{\frac{1}{2}Q\nabla^2\big(B^2(\boldsymbol{r},t)\rho(\boldsymbol{r},t)\big)}_{\text{Stochastic diffusion term}}$$

where ρ is the robot density at position $\boldsymbol{r}$ and time t

Q is a scalar for the stochastic term

∇ is the nabla operator and ∇^2 is the Laplacian

Modelling space dynamics

The Fokker-Planck equation:

$$\frac{\partial \rho(\boldsymbol{r},t)}{\partial t} = \underbrace{-\nabla\Big(\boldsymbol{A}(\boldsymbol{r},t)\rho(\boldsymbol{r},t)\Big)}_{\text{Non-stochastic drift term}} + \underbrace{\frac{1}{2}Q\nabla^2\Big(B^2(\boldsymbol{r},t)\rho(\boldsymbol{r},t)\Big)}_{\text{Stochastic diffusion term}}$$

We can derive the fraction of robots at time t over area W by

$$s(t) = \int_{r \in W} \rho(\boldsymbol{r},t)$$

Modelling space dynamics

The Fokker-Planck equation:

Fig. 5.9 Evolution of a probability density modeled by the Fokker–Planck equation, Eq. (5.25). Example in 1-d space with a drift of $A = 0.1$ and diffusion $B = 0.3$. Shown is the probability density and four example realizations of the corresponding Langevin equation

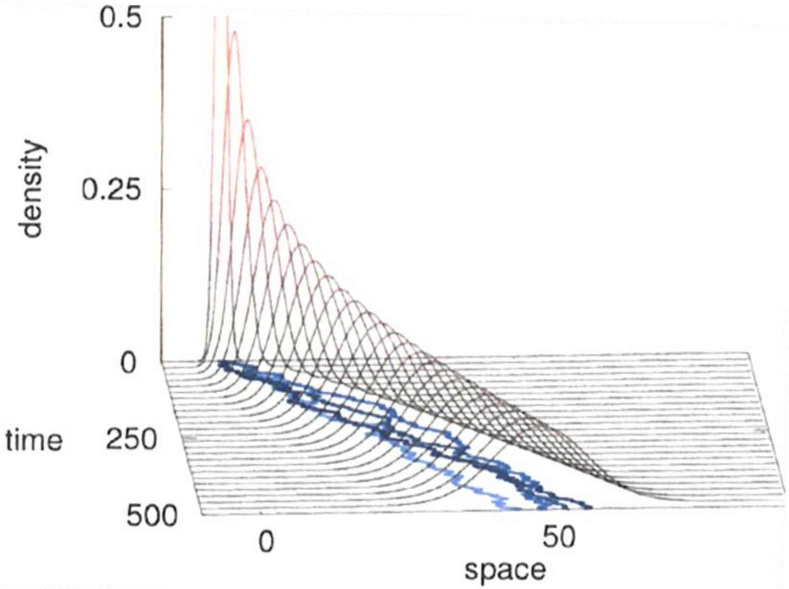


Image: Figure 5.9, Hamann, 2018

Network models

A swarm of robots can be interpreted as a network with robots as nodes and edges indication mutual neighbourhood relations, typically:

1. Erdos-Renyi random graphs, does not model spatial structures
2. Geographic random graphs, static models only
3. Need adaptive networks [Gross and Sayama, 2009] allowing dynamic updates of their topology

Formal design methods

Top-down or bottom-up approaches?

Multi-scale modelling for algorithm design

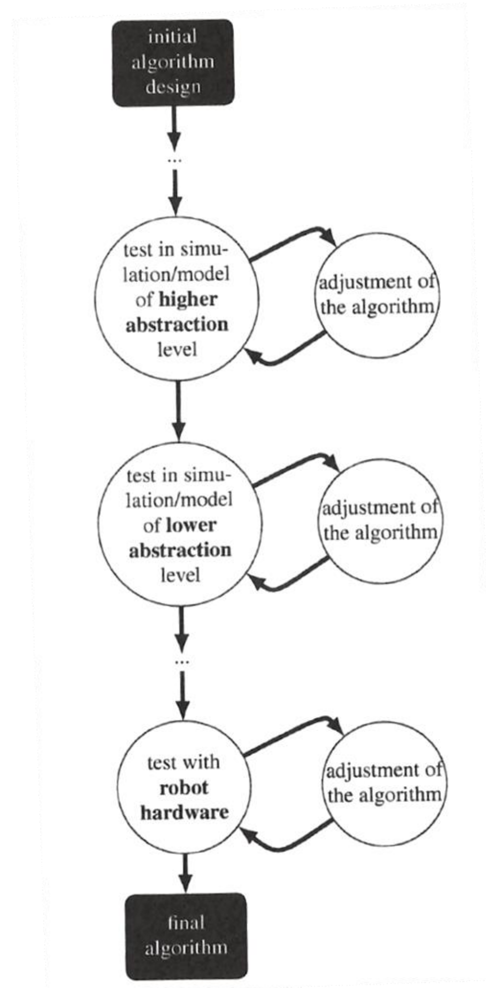


Image: Figure 5.12, Hamann, 2018

Formal design methods

Automatic design of robotic controllers, e.g. reinforcement learning, ANN, EC, etc,

Typically,

1. The required amount of training data is often hard to obtain
2. Well-defined objective functions for swarms are hard to derive
3. System state space is growing exponentially with swarm size

Formal design methods

Software engineering and verification

Formal verification of unrestricted swarm systems are hard to perform and validate.

How do we define emergence as a property?

Summary lecture 4 – Swarm robotics 1*

- What is swarm robotics?
- Swarm performance
- Scenarios of swarm robotics
- Modelling approaches
  - Local sampling
  - Time-space dynamics
  - Fokker-Planck micro-macro link
- Formal design methods in swarm systems

*Hamann, 2018: chapter 1, 4 and 5