Void Reduction in Self-Healing Swarms

(Using proximity detection)

Neil Eliot[1], David Kendall[1], Alun Moon[1], Michael Brockway[1], Martyn Amos[1]

[1]Department of Computer and Information Sciences
University of Northumbria

ALife, 2019

Video

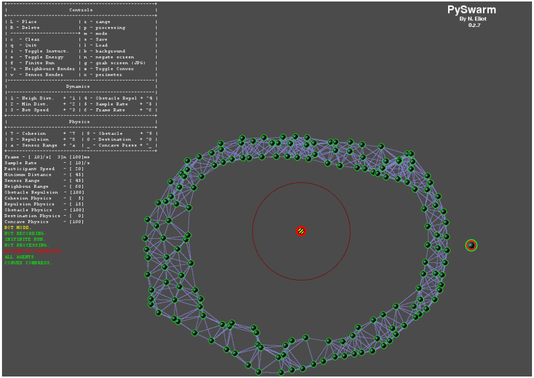

Figure: Simulator

https://www.youtube.com/watch?v=iyMSpj10elk

Introduction

Swarm Rules

- Swarms consist of many agents (mobile robots or drones) that interact according to a simple set of rules.
- We consider swarms of agents that:
  - Capable of detecting their neighbours (proximity detection).
  - Do not require any another form of communication.
- Swarms can be made fault tolerant (resilient to agent loss).

Why no communications?

- Communication propagation protocol overhead.
  - $n_1 \rightarrow n_3$
  - $n_1 \rightarrow n_2$
  - $n_3 \rightarrow n_2$ (decision!)
- Message propagation takes time which limits swarm size.

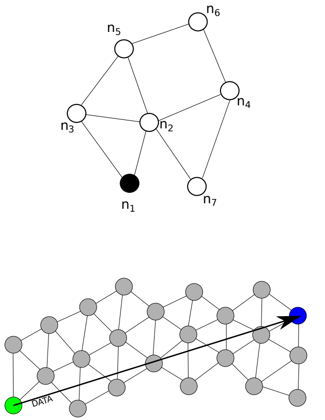

Figure: Swarm Communications

Proximity Sensing

- Proximity detection only, no other form of communication required.
- Arbitrary sized swarms possible.
- Agent attributes include various ranges (as shown in figure).

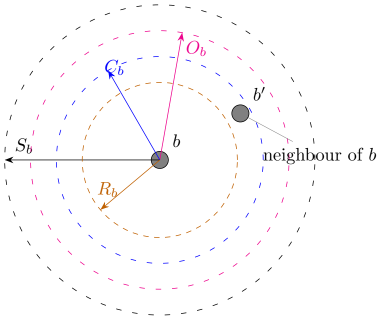

Figure: Ranges - R_b - repulsion, C_b - cohesion, S_b - sensing, and O_b - obstacles avoidance.

Agent Movement

- Agent movement is computed as the weighted sum of 4 vectors, as shown in equation 1

$$v(b) = k_c v_c(b) + k_r v_r(b) + k_d v_d(b) + k_o v_o(b) \tag{1}$$

- $v_c(b)$ - cohesion to ensure agents remain part of the swarm.
- $v_r(b)$ - repulsion to ensure agents do not collide.
- $v_d(b)$ - destination vector for goal based swarms.
- $v_o(b)$ - obstacle avoidance vector.
- k_c, k_r, k_d, k_o are weightings to allow modifications to the vector effects.

The Swarm

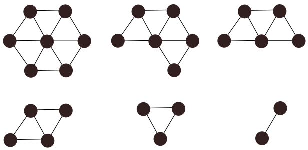

Figure: Stable Structures

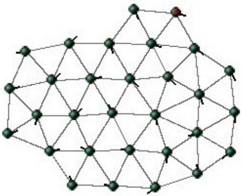

Figure: The Swarm - (Simulator)

Perimeter Detection

NOTE

Perimeter detection is used as part of the *void reduction* process. This will be discussed later.

- Perimeter detection allows for directional coordination with reduced resource usage.
  - ‘Internal’ agents don’t need to use their GPS.
- Reduces computational overhead in agents.

What is a Perimeter?

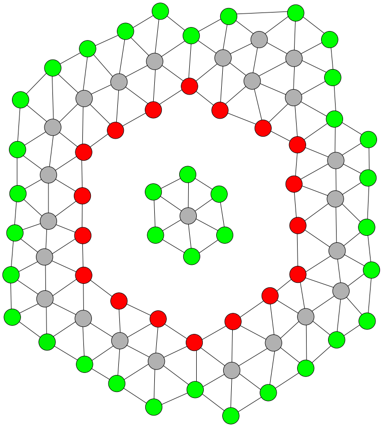

Figure: Internal (red) and external (green) perimeters

Perimeter Detection (Concave)

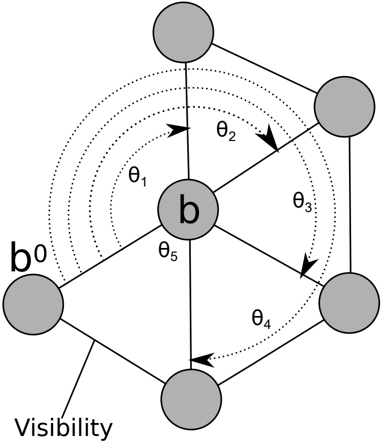

Figure: Concave gap (*Void Reduction*)

Perimeter Detection (Convex)

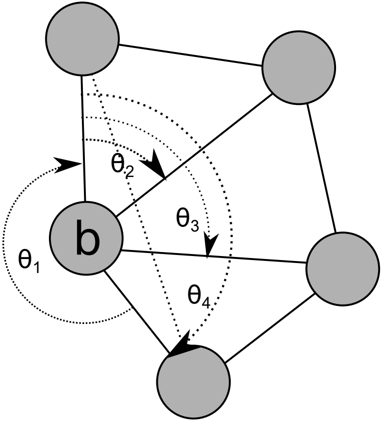

Figure: Convex gap

Void Reduction Movement

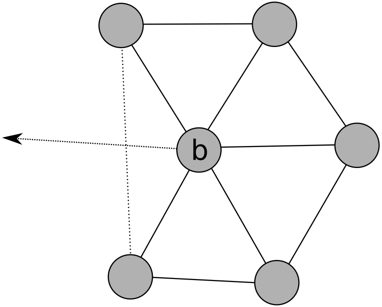

Figure: Concave detection

As part of the perimeter detection a pair (G_b) of agents is generated. This is the first two agents identified as creating a 'gap' in agent b's neighbours. Equation 2 calculates the centroid of the identified 'gap'.

$$D_{pos}(b) = \frac{1}{2}\sum_{n \in G_b} n \qquad (2)$$

Agent movement

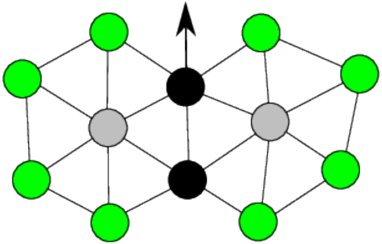

Figure: Initial positions

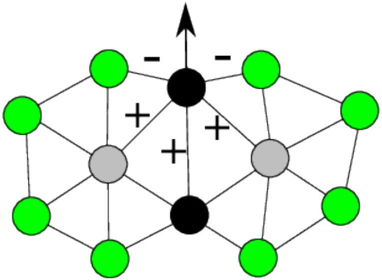

Figure: Agent movement

Agent movement

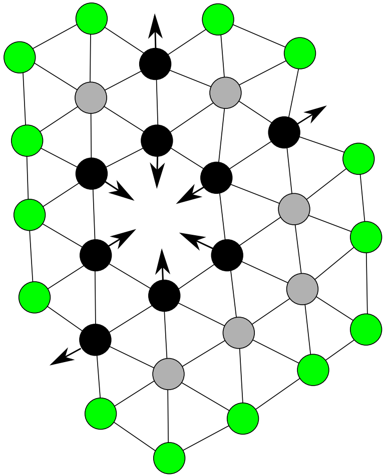

Figure: Initial positions

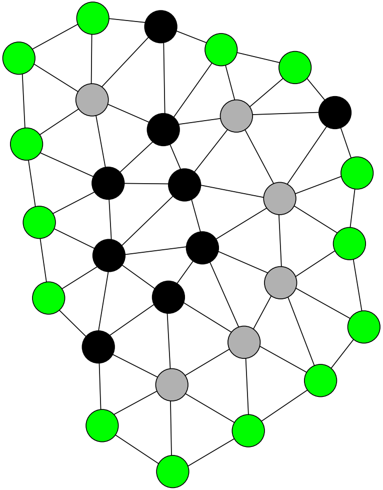

Figure: Agent movement

Scenario

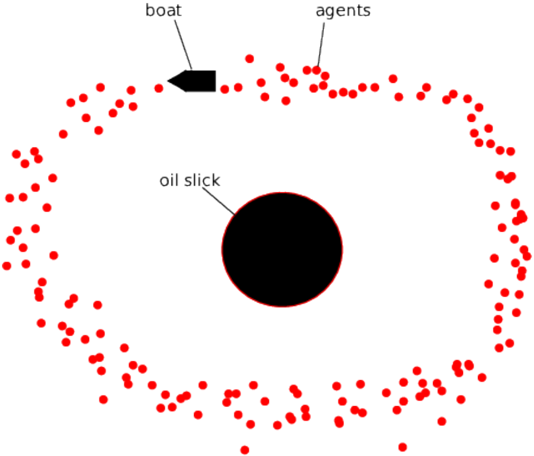

Figure: Oil Slick Encapsulation

Summary

- Void reduction has a local effect creating a global emergent behaviour that improves the shape and structure of a swarm.
- Void reduction removes anomalies on internal and external perimeters.
- Void reduction can be applied to both static swarms and directional swarms.

Thank You

THANK YOU!
QUESTIONS?

Agent Movement

$$v_r(b) = \frac{1}{|\mathcal{R}_b|}\left(\sum_{b' \in \mathcal{R}_b}\left(1 - \frac{|b'|}{R_b}\right)b'\right) \tag{3}$$

$$v_c(b) = \frac{-1}{|\mathcal{C}_b|}\left(\sum_{b' \in \mathcal{C}_b} b'\right) \tag{4}$$

Agent Movement

$$v_d(b) = d \tag{5}$$

$$\begin{aligned} v_o(b) &= O_b \hat{q}_o \\ \text{where } q_o &= \sum_{o \in \mathcal{O}_b} \hat{o} \\ v_o(b) &= O_b \left(\sum_{o \in \mathcal{O}_b} \hat{o} \right)^{\wedge} \end{aligned} \tag{6}$$