Designing Brooks-style Subsumption Swarms

Todd Wareham

Department of Computer Science
Memorial University of Newfoundland

October 14, 2013
Dagstuhl Seminar 1124
Introduction

- Many design methodologies proposed (Crespi et al, 2008; Brambilla et al, 2012), e.g.,
  - temporal-logic decomposition (Winfield et al, 2005a)
  - dataflow diagram decomposition (Winfield et al, 2005b)
  - interaction-graph decomposition (Wiegand et al, 2006)
  - evolutionary algorithm (Sperati et al, 2011)
- All existing methods inadequate.
Defining Swarms:
Swarm Entity Architecture

Our reactive robot will be a simplified Brooks-style architecture [?] consisting of sensors, a set of layers, a total ordering on these layers, and a set of subsumption connections between layers. These components are specified as follows:

- The sensors can see outwards in a radius $r$ around the robot in every direction up to the closest obstacle in that direction, and can verify, for each square-type $e \in E$, the presence of $e$ at any specified position $pos$ within that perceptual radius, i.e., $\exists(pos, e)$. Each robot also has a compass that allows it to orient itself relative to the north-south and east-west axes.

- Each layer has a trigger-condition that is a Boolean formula over the available sensory $\exists$-predicates and an action $a \in \{N, S, E, W\}$. If a layer’s formula evaluates to $True$, the layer produces output $a$; otherwise, it produces the special output null. Given a set of layers $L$, we will assume that the formula in each layer has length at most $f$ and no two layers compute the same Boolean function and produce the same output.
Make Me Work Late Will You
Defining Swarms: Swarm Entity Architecture

- Relative to the total order on the layers, a layer $i$ can have subsumption links to any layer $j$ that is lower than $i$ in the ordering; between any two layers, there can exist an output-inhibition or output-override link (but not both). An output-inhibition link from a layer $L$ to a layer $L'$ makes the output of $L'$ null if the output of $L$ is non-null. The set of output-override links to a layer $L'$ are assumed to be in a total order, and the output of $L'$ is either the value of the highest non-null layer-override link in the total order, if there is an output override link whose value is non-null, and the output specified by $L'$ otherwise. The output of any layer that subsumes at least one lower-level layer is not available directly for output; otherwise, that layer’s output is available.

The output of a set of ordered layers with subsumption links will be that of the highest layer relative to the order that is both available and non-null.
Defining Swarms: Overall Swarm Architecture

- Restrictions (this talk):
  - Synchronized entity movement.
  - No inter-entity communication.
  - No movement conflict allowed.

- Modifications:
  
  **Selection**: Add / delete up to \( c \) entities (relative to provided entity library \( A \))
**Defining Swarm Design**

**SWARM NAVIGATION WITH X**

*Input:* World $W$, swarm $S$, start and finish points $s$ and $d$ in $W$, integer $c$.

*Output:* A swarm $S'$ derived by at most $c$ modifications of type $X$ from $S$ that can move conflict-free from $s$ to $d$, if such an $S'$ exists, and special symbol $\perp$ otherwise.
Defining Swarm Design (Cont’d)

- **Given Swarm Navigation (GSN)**
  Given \( W, S, \) start-position \( s \) and destination-area \( d \), can \( S \) get from \( s \) to \( d \)?

- **Selected Swarm Navigation (SSN)**
  Given \( W, |S|, A, \) and areas \( s \) and \( d \), derive \( S \) and position of \( S \) in \( s \) such that \( S \) can get from \( s \) to \( d \).

- **Given Swarm Navigation with Rec. (GSN-REC)**
  Given \( W, S, M, \) start-position \( s \) and destination-area \( d \), derive \( S' \) from \( S \) wrt \( M \) such that \( S' \) can get from \( s \) to \( d \).

- **Selected Swarm Navigation with Rec. (GSN-REC)**
  Given \( W, |S|, A, M, \) and areas \( s \) and \( d \), derive \( S \) wrt \( A \) and \( M \) and position of \( S \) in \( s \) such that \( S \) can get from \( s \) to \( d \).
Computational Complexity Analysis

- A problem $\Pi$ is poly-time solvable if $\Pi$ is solvable in time $n^c$ for input size $n$ and constant $c$.
- In Computer and Cognitive Science, efficient solvability = poly-time solvability (see van Rooij (2008) and references).
- Basic questions about a computational problem $C$:
  1. Is $C$ hard, i.e., is $C$ poly-time solvable?
  2. If so, what can we restrict to make $C$ easy, i.e., (effectively) poly-time solvable?
- Use classical complexity to show problem is not poly-time solvable, i.e., $NP$-hardness (Garey and Johnson, 1979).
Computational Complexity Analysis (Cont’d)

Definition
Let $\Pi$ be a problem with parameters $k_1, k_2, \ldots$. Then $\Pi$ is said to be **fixed-parameter (fp-) tractable** for parameter-set $K = \{k_1, k_2, \ldots\}$ if there exists at least one algorithm that solves $\Pi$ for any input of size $n$ in time $f(k_1, k_2, \ldots)n^c$, where $f(\cdot)$ is an arbitrary function and $c$ is a constant. If no such algorithm exists then $\Pi$ is said to be **fixed-parameter (fp-) intractable** for parameter-set $K$.

Lemma
[?, Lemma 2.1.30] If problem $\Pi$ is fp-tractable relative to parameter-set $K$ then $\Pi$ is fp-tractable for any parameter-set $K'$ such that $K \subset K'$.

Lemma
[?, Lemma 2.1.31] If problem $\Pi$ is fp-intractable relative to parameter-set $K$ then $\Pi$ is fp-intractable for any parameter-set $K'$ such that $K' \subset K$. 
Computer Networks: Commercialization (Cont’d)
<table>
<thead>
<tr>
<th>Category</th>
<th>tractable</th>
<th>intractable</th>
</tr>
</thead>
<tbody>
<tr>
<td>classical</td>
<td>(n^c)</td>
<td>(NP)-hard)</td>
</tr>
<tr>
<td>parameterized</td>
<td>(f(p) \times n^c)</td>
<td>(W)-hard)</td>
</tr>
</tbody>
</table>
Complexity of Swarm Design

- Main results:
  - SSN, GSN-REC, and SSN-REC are poly-time intractable.

- Implications:
  - Swarm design problems are intractable in general (as GSN is not so much swarm design as swarm verification).
  - Need to restrict these problems if we are to get tractability.
### Complexity of Swarm Design (Cont’d)

|                | \( |L| \) | \( |E| \) | \( f \) | \( r \) | \( |S| \) | \( h \) | \( |A| \) | \( |M| \) |
|----------------|--------|--------|--------|--------|--------|--------|--------|--------|
| **SRSM**       |        |        |        |        |        |        |        |        |
| B              | 4      | 5      | -      | -      | \( p \) | 1      | 1      | X      |
| C              | 3      | -      | 13     | 2      | \( p \) | \( p \) | -      | X      |
| D              | 3      | 5      | -      | -      | \( p \) | \( p \) | -      | X      |
| **GRSMR**      |        |        |        |        |        |        |        |        |
| B              | 3      | 3      | -      | -      | -      | 2      | \( X \) | 1      |
| C              | 4      | -      | 13     | 2      | \( p \) | \( p \) | \( X \) | -      |
| D              | 4      | 5      | -      | -      | \( p \) | \( p \) | \( X \) | -      |
| E              | \( p \) | -      | 1      | 0      | 1      | 1      | \( X \) | -      |
| F              | \( p \) | 5      | -      | -      | 1      | 1      | \( X \) | -      |
| G              | -      | -      | 3      | 1      | 1      | 1      | \( X \) | 0      |
| H              | -      | 5      | -      | -      | 1      | 1      | \( X \) | 0      |
| **SRSMR**      |        |        |        |        |        |        |        |        |
| B              | 4      | 5      | -      | -      | \( p \) | 1      | 1      | 0      |
| C              | 3      | -      | 13     | 2      | \( p \) | \( p \) | -      | 0      |
| D              | 3      | 5      | -      | -      | \( p \) | \( p \) | -      | 0      |
| E              | \( p \) | -      | 1      | 0      | 1      | 1      | 1      | -      |
| F              | \( p \) | 5      | -      | -      | 1      | 1      | 1      | -      |
| G              | -      | -      | 3      | 1      | 1      | 1      | 1      | 0      |
| H              | -      | 5      | -      | -      | 1      | 1      | 1      | 0      |
Future Work

- Extend parameterized analysis to other aspects, *e.g.*, perceptual radius.
- Analyze swarm design relative to other types of worlds, tasks, and architectures.
- Investigate related problems, *e.g.*, reactive morphogenesis.