Swarming with Logic

Robert L. Stewart
Department of Computer Science & Software Engineering
The University of Melbourne
Victoria, Australia
robertls@csse.unimelb.edu.au

Michael Kirley
Department of Computer Science & Software Engineering
The University of Melbourne
Victoria, Australia
mkirley@csse.unimelb.edu.au

ABSTRACT

A robot swarm is a distributed entity that can sense and perform many actions simultaneously at different spatial locations. But how, with all this sensory information and capacity for action, can the individuals unite for a common purpose, coordinating their actions in space and time? In this preliminary simulation study, we suggest that a swarm can be designed as an engineering control system characterised by a problem specific input/output (I/O) relationship that can be implemented through the simple (social insect inspired) indirect transfer of information between individuals. For such a system, the inputs represent sensory information acquired by one or more agents and the outputs are used to trigger actions that agents perform.

Simulations of a 2D bounded grid world have been conducted in which agents wander randomly and transport physical material (blocks) to and from differently labelled spatial locations called caches (Fig. 1). The presence or absence of blocks in caches are used to encode the binary states of the system which has an I/O relationship governed by a set of boolean expressions. To ensure deadlock is avoided, agents alternate between carrying and not carrying a block by loading/unloading at depot cells which serve as a temporary storage facility.

Whenever an agent encounters a cache cell (of type L) it senses whether or not the cell contains a block. By maintaining a fixed-length history of these observations, an agent can calculate the percentage (P_L) in which blocks have been observed, and use this information to form an assessment (opinion) about the cache's state. Provided a sufficient number of observations have been made, a cache state can be assessed as "on" ($P_L \geq min\_on$), "off" ($P_L \leq max\_off$) or "undefined" ($max\_off < P_L < min\_on$).

Using its cache assessments, an agent is able to determine if the state of an encountered cache cell is consistent with the problem specific I/O relationship. If it is inconsistent, the agent can undertake a remedial action (e.g. remove or deposit a block). For the situation depicted in Fig. 1, most of the agents are likely to assess both caches A and B as "off". Therefore they will fill any empty cells discovered in cache C in an endeavour to satisfy $C = A$ NAND $B = \overline{A.B}$.

A number of systems with different boolean I/O relationships have been implemented including a NOT gate, a NAND gate, a ring-oscillator and a 4-gate combinatorial logic device. These have been tested by observing whether the correct outputs are produced in response to externally provided inputs (note: in practice agents would modify the inputs in response to stimuli). Initial findings suggest that, for a given environment, the number of agents, the history lengths and the number of cells in each cache all play an important role in ensuring convergence to a consistent global state (i.e. they determine if consensus of opinion can be reached). It would appear that larger history and cache sizes improve convergence but with a degraded response time.

This initial study has raised many interesting avenues for future work including: (i) the analysis of convergence and response time for different system parameters, (ii) the study of direct communications in which virtual blocks are passed between agents that carry virtual cache cells, (iii) the use of a real robotic swarm with sensing uncertainties, (iv) the study of systems with an I/O functionality characterised by a frequency response, (v) the investigation of how caches might encode continuous variables, and, (vi) the implementation of other logic devices including a swarm computer and a swarm display where each cache acts as a pixel.

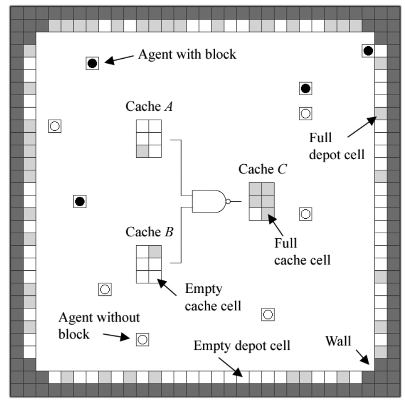

Figure 1: A simulated grid world containing two input caches (A and B), one output cache (C) and 10 robots. Here a NAND logic gate is implemented with the system I/O relationship given through the single boolean expression: $C = A$ NAND $B = \overline{A.B}$.

Categories and Subject Descriptors: I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Multi-agent systems, Coherence and coordination*

General Terms: Design

Keywords: Multi-agent systems, Swarm robotics, Swarm intelligence, Collective decision making, Control systems

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.