Technical Status Report **October 10, 2001**

DARPA Agent Based Computing (ABC) Program, Taskable Agent Software Kit (TASK)

PI: Lee Spector, Hampshire College
Research Assistant: Alan Robinson

Project: Multi-type, Self-Adaptive Genetic Programming for Complex Applications

---

Accomplishments This Quarter

• In collaboration with the MIT/BBN group, applied the PushGP genetic programming to automatically produce transport network agents. Several interesting agents have evolved and some of the evolved agents are adaptive. See "Evolved Transport Network Agents" below.

• Received code for the three-dimensional Opera problem from the Dartmouth group and began investigating the application of PushGP to the evolution of Opera agents. A meeting with the Dartmouth group was postponed but will be rescheduled.

• Added diversity constraint mechanisms to the PushGP and Pushpop systems and collected data on their efficacy using the 16-node cluster. These tests are ongoing.

• Applied PushGP to new problems in quantum computation, using genetic programming to discover new quantum communication protocols involving particular gates of interest to quantum computation researchers. The system, running on the 16-node cluster, has made several discoveries over the last month but their significance is not yet clear. This work is ongoing.

• Attended the CAHDE REF group meeting at the MIT Media Lab August 27-28. Our presentation provided brief tutorials on genetic programming and of the evolution of agents for dynamic environments. The PushGP and Pushpop systems were described, along with the heterogeneous, dynamic environments for which we are currently evolving agents. The slides for the presentation are available at http://hampshire.edu/lspector/CAHDE-Aug2001.pdf.

• Submitted an article, "Genetic Programming and Autoconstructive Evolution with the Push Programming Language," to the journal *Genetic Programming and Evolvable Machines.*

• Notified of acceptance of an article, "Hierarchy Helps it Work That Way," to the journal *Philosophical Psychology.* Revisions are in progress.

Current Plans

• Work with MIT/BBN group to compare recently evolved agents to previous (hand-constructed) agents based on elementary adaptive modules.

• Continue ongoing work on:

- Evolution of transport network agents in a variety of environmental conditions. Assess agent quality and adaptivity in each condition.
- Assessment of efficacy of diversity constraint mechanisms in PushGP
- Discovery of new quantum communication protocols using PushGP.

• Work with MIT/BBN group to provide elementary adaptive modules as primitives in PushGP and evolve agents that may make use of them. Compare agents produced with and without access to elementary adaptive modules.

• Meet with Dartmouth group to discuss the evolution of agents for the three-dimensional Opera problem.

• Attend:

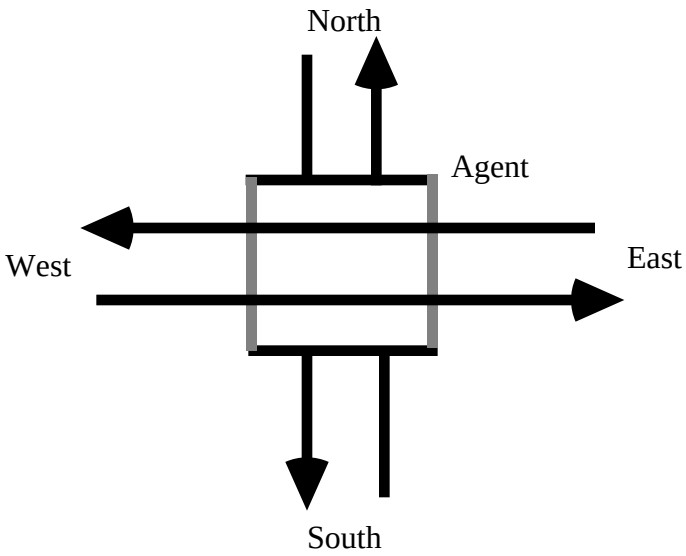
- DARPA QUIST meeting to connect this project to other DARPA quantum computation work.
- CAHDE REF meeting tentatively scheduled for late October in Boston.
- TASK PI meeting tentatively scheduled for November.

Evolved Transport Network Agents

Simulator Configuration

All current evolutionary results have been found with Hampshire College's transport network agent simulator. Our simulator works similarly to the BBN simulator so agents should work when transferred between systems, once any differences in data formats have been resolved.

The simplest possible transport network was used in order to maximize the speed of evolution. The network consisted of four one-way transit corridors crossing at one location. We designate the flow directions as North/South/East/West for convenience only—there is nothing inherently rectilinear or two-dimensional about the underlying simulation. In this setup, control agent status can be "green" (meaning "go") for North/South (and red/stop for East/West), or the reverse. Therefore, an agent is only required to specify the time-green value for North/South corridors.



A simple network transport scenario in which the agent is currently permitting only North/South flow. “North/South/East/West” are arbitrary designations—there is nothing inherently rectilinear or two-dimensional about the underlying network transit simulation.

Each corridor is long enough to fit 20 vehicles at one time, and the control agents are located at the halfway point of each corridor. Until the first time the evolved agent is called a default value for time-green is used—this is 0.5 (green for half a light cycle). The agent’s cycle length is a constant 20 time steps.

Metrics

The simulation made the following data available to each agent:

• Time green: amount of time the North/Sound corridor is in green mode. Expressed as float from 0 to 1. Higher = longer green.

• Average windowed wait: the amount of time spent waiting (instead of moving forward) for the twenty most recent vehicles to enter each corridor, who have also been in the grid for at least 5 time steps. This was gathered in terms of each corridor, as well as an aggregate value for all the vehicles in the network. Expressed as a float between 0 and 1, where 1 means "yet to have moved", and 0 means "never had to stop".

• Maximum wait: the longest any vehicle has spent waiting for the entire history of the network. Gathered in terms of each corridor, and an aggregate of all corridors. Expressed an integer equal the number of time steps where the vehicle could not move.

We also tried providing less data to the agent; namely just the average windowed wait values for each corridor along with the pervious time green value. So far evolution has not been able to find successful agents when provided with just that amount of information, which suggests it is

necessary to include the full set of metrics/data described above.

Fitness Cases

In order to encourage evolution to develop a control agent with general and adaptive ability to minimize wait times, fitness was measured in terms of average performance over 22 different flow densities. For each configuration of flow density the simulation was run for 200 time steps to measure how effective the agent was over a reasonable span of time. At every 20th time step, the evolved agent was evaluated to determine what new time-green value to use for the next 20 time steps. If the agent returned a value that was not between 0 and 1 then the pervious time-green value was used instead.

Flow rates for each fitness case:

Fitness Case	North Bound	South Bound	East Bound	West Bound
1	.25	.25	.25	.25
2	.1	.1	.9	.8
3	.1	.1	.8	.8
4	.1	.1	.7	.7
5	.1	.1	.6	.6
6	.1	.1	.5	.5
7	.1	.1	.4	.4
8	.1	.1	.3	.3
9	.1	.1	.2	.2
10	.1	.1	.1	.1
11	.9	.9	.1	.1
12	.8	.8	.1	.1
13	.7	.7	.1	.1
14	.6	.6	.1	.1
15	.5	.5	.1	.1
16	.4	.4	.1	.1
17	.3	.3	.1	.1
18	.2	.2	.1	.1
19	.1	.1	.1	.1
20	.9	.9	.1	.1
21	.3	.3	.5	.5
22	.9	.01	.01	.01

A small amount random variation was also added; each time the simulator runs a slightly different total number of vehicles is added to each corridor. This was done in part to promote the

evolution of general and adaptive agents. The amount of variation was the same across all flow corridors and all fitness cases.

What Evolved

So far, the most fit agent that was evolved, in terms of ability to keep the average-wait metric of all cars to a minimum, uses the following formula to determine a new time green value (for the North/South corridor) each time it is called:

```
NewTimeGreen = OldTimeGreen
               + WindowedAverageWait(northCorridor)
               + WindowedAverageWait(southCorridor)
               + WindowedAverageWait(eastCorridor)
               + WindowedAverageWait(westCorridor)
               + MaxWait(southCorridor)
               + MaxWait(westCorridor)
               - MaxWait(northCorridor);
```

Here is an example of a typical simulation run using the above formula:

Traffic Densities				All of an agent's output values (when in range). Number shown is TimeGreen scaled by 20 (the length of the agent's cycle)
North	South	East	West	
0.1	0.1	0.9	0.9	13
0.1	0.1	0.8	0.8	9
0.1	0.1	0.7	0.7	9
0.1	0.1	0.6	0.6	9
0.1	0.1	0.5	0.5	9
0.1	0.1	0.4	0.4	10
0.1	0.1	0.3	0.3	9
0.1	0.1	0.2	0.2	9
0.1	0.1	0.1	0.1	9
0.9	0.9	0.1	0.1	9
0.8	0.8	0.1	0.1	17
0.7	0.7	0.1	0.1	17
0.6	0.6	0.1	0.1	17
0.5	0.5	0.1	0.1	16
0.4	0.4	0.1	0.1	16
0.3	0.3	0.1	0.1	14
0.2	0.2	0.1	0.1	13
0.1	0.1	0.1	0.1	11
0.9	0.9	0.1	0.1	9
0.3	0.3	0.5	0.5	17
0.9	0.01	0.01	0.01	10 16 18 10

Note that for most flow densities the agent produced one single time green value at the beginning

of the simulation and used that constant value for the rest of that simulation. This is because for most subsequent evaluations of the evolved agent the value returned is outside the range of 0-1, signalling to continue using the previous value. In earlier generations of evolution we typically saw agents that dynamically varied the time-green value within a single flow density configuration. It is interesting that with further evolution this better performing solution was found, which typically does not change time-green dynamically.

The performance of the evolved agent is better than choosing a constant time-green value for all fitness cases, as shown in the following table:

Behavior	Fitness (summed average wait values across all fitness cases)
Evolved agent	1.3
Constant time-green of 0.5	3.1
Constant time-green of 0.2	3.0
Constant time-green of 0.8	2.4

Note that the evolved agent reduced wait time by roughly two and one half times over that of keeping the North/South corridor open (green) 50% of the time (the default condition). We also tested all constant green times (in increments of 0.1) and determined that the value of 0.8 was the optimal constant. Note also that the evolved agent performed almost twice as well as the optimal constant value.