Minutes from "Multi-agent Oriented Constraint Satisfaction"

Daniel Buettner
Lecture by Lin Xu
CSCE 976 Advanced Artificial Intelligence

May 2, 2002

1 Introduction

In the lecture from 1 May, we covered material from a brand new paper by Jiming Liu, Han Jing, and Y.Y. Tang entitled *Multi-agent oriented constraint satisfaction*. This paper discusses a "new" method of stochastic search that is "totally-different" from local-search. In fact, this "new" method can solve *both* the n-queens problem *and* coloring problems!

2 Lin's lecture

The paper begins with a definition of a constraint satisfaction problem. Lin noted that we were all quite familiar with this definition, but discussing it helped to use up time in his presentation. The paper describes two general methods for solving a CSP: generate and test (GT), and backtracking search (BT). GT generates possible combinations of variables and checks whether it has found a solution. BT systematically searches for solutions.

There are several improvements that have been made to increase the performance of BT. Thrashing can be mitigated by application of consistency techniques. Redundant work can be avoided by using a dependency directed backtracking scheme. Of course, general CSP problems are NP-complete, so even with these enhancements BT is still unable to solve sufficiently large problems.

GT can be improved by making use of heuristics and stochastic algorithms. Local search methods typically use iterative repair or hill climbing techniques. To escape from local optima, these techniques will generally make use of a random walk or random restart.

CSPs have also been solved by making use of neural networks and genetic algorithms. Of course, no one method is perfect. Systematic search is sound and complete: if a solution exists, it will be found. However, if the problem is too large, systematic search will exhaust the system's memory. Local search has a much better chance of solving large problems, but it is not complete: it makes no guarantee about ever finding the solution, nor can it report that no solution exists.

Multiagent systems utilize several agents that work together in order to achieve some goal. The agents themselves may have different goals, but their organization can ensure that some overall goal is accomplished. CSPs can be solved by a multiagent system by partitioning the variables into subproblems which can each be solved by an agent. The constraints of the problem are then placed between the agents. Finding a solution requires that all agents find the values for their variables that satisfy not only their own constraints but also interagent constraints.

Swarm-like systems are one method of simulating distributed multi-agent systems which involves

- a living environment
- agents with reactive rules
- schedule serving

One of the paper's authors developed an evolutionary autonomous agent system as well as an energy-based artificial-life model for solving the n-queens problem.

The paper proposes an approach that they call ERA which stands for Environment, Reactive rules, and Agents (this is somewhat confusing, as ERA traditionally stands for earned run average[1]). This approach is intended to provide a multiagent formulation for solving general CSPs as well as a method that can find an "approximate" "solution" without too much cost. These "approximate" "solutions" are variable assignments that break several constraints, and thus aren't actually solutions to the problem. The authors claim that the main difference between ERA and local search is in the evaluation function. Of course, if this is the only difference, they aren't actually different.

[1]The ERA for a pitcher is calculated by multiplying the total number of earned runs by nine, and dividing the results by the total innings pitched. This statistic represents the average number of runs that a given pitcher would give up if he were to pitch a complete game.

Lin then went on to discuss the fundamentals of the ERA approach. And agent is a virtual entity that lives and acts in some environment. The agent is able to sense the environment and is driven by certain objectives. The agent should have some reactive behaviors. A multiagent system contains an environment E in which the agent lives, a set of reactive rules R that govern the interaction between the agent and its environment, and a set of agents A.

The environment records the number of constraint violations in the current state. Each agent represents a variable and the position of the agent corresponds to the value of the agent. The objective of each agent is to move to a position where the number of constraint violations is zero. A solution state is defined as a state in which every agent is in a "zero-position".

To find a solution state, the agents will select and execute some predefined local reactive behaviors. The moves are guided by least-move, better-move and random-move (WalkSAT did this years ago). In time step 0, the system is initialized and each agent is assigned a random position. At each clock increment, all of the agents decide on a new location to occupy. The system terminates when all agents reach a zero-position or the the clock has exceeded a time threshold.

According to the authors, the ERA algorithm has properties of termination and correctness. They guarantee termination because the algorithm will either find a solution or exceed the allowed time slice. Correctness is ensured because the algorithm will only return a solution that violates no constraints.

However, the authors then go on to discuss their notion of an "approximate" "solution". This is a "solution" which violates some number of constraints. They claim that their algorithm will *always* evolve towards a state in which more constraints are satisfied.

Next, the authors compare their approach to those used by other researchers. They claim that their method is better than min-conflict heuristics because hill-climbing can get stuck in local minima. However, nobody uses a hill-climbing search that doesn't make use of random restart, so this is a somewhat silly argument.

They next compare their work to Yokoo's distributed constraint satisfaction. Yokoo's approach does not require global broadcasting but the authors believe that his strategy of abandoning partial solutions after one failure may be too costly. The authors comment that their use of broadcast is cheap and helps reduce overhead cost. This seems odd.

Despite the fact that the ERA method is supposed to involve agents operating simultaneously, it was implemented sequentially. One question

that they never address is how they would maintain coherency of information in a truly parallel implementation.

3 Student interaction

TIBOR:	Broadcasting seems like a bad idea.
A:	Yes, most people think multicasting is a much better way to share information.
CORY:	This was a bad paper. The author says nothing new. Rodney Brooks would not approve!
A:	It's true.
DANIEL:	Why is local search called incomplete, but this paper's method is called complete?
A:	That's a good question. Their approach seems equivalent to local search.
ROB:	Can I implement sort using artificial life?
A:	Why not?
ROB:	Their logic notation is very confusing.
A:	Very true. They seem to be using object-oriented dot notation instead of predicates.
SHABBIR:	At a later point in the paper, they say that ERA is incomplete.
A:	Wow! They contradict themselves!
PRAVEEN:	I read 2 pages and simply couldn't force myself to read anymore. I could not believe that this was a recent paper.
A:	It does seem strange that it was chosen for publication.
PRAVEEN:	What is this "Energy System" that they mention?
A:	It seems to use the idea of entropy to find some configuration that minimizes energy.
AMY:	Quote of the day: ".. the ERA approach can be very efficient and robust when applied in the right context."
LIN:	I apologise for making you read this paper.