Real-world applications:
Motion planning using GAs

Craig Eldershaw and Stephen Cameron
Oxford University Computer Laboratory
Wolfson Building, Parks Rd
OX1 3QD, United Kingdom
{ce,cameron}@comlab.ox.ac.uk
+44 1865 273 838

Motion planning is a field of growing importance as more and more computer controlled devices are being used. This paper considers how to convert a general motion planning problem into one of global optimisation—one that can be solved with GAs. A program has been written which implements this.

Motion planning is the algorithmic calculation of a set of movements for one or more controllable objects. This motion aims to achieve a certain goal, while not violating any of the given restrictions[1]. Common examples of this problem are moving a robot vehicle between two given positions and orientations, or moving a machine tool between a start and goal poise.

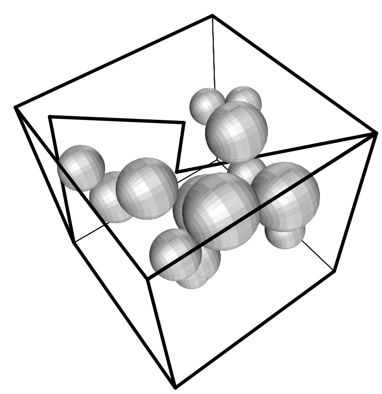
The environment in which the vehicle must move is often very complex. Furthermore, the vehicle itself may be a complex structure or possess a large degree of flexibility. To simplify the detection of collisions, the real-world space is mapped to *configuration space*. Each dimension in this corresponds to one of the vehicle's degrees of freedom (eg. two spatial and one rotational). In configuration space, the vehicle is represented by a single point which completely characterises the location/orientation of the obstacle.

Portions of the space are blocked out where those configurations correspond to forbidden regions (eg. those that involve collisions). These regions are very difficult to model exactly, however they can be approximated to arbitrary accuracy by a set of spheres[2]. This simplifies tests for if path segments cross any forbidden regions (configuration space obstacles).

The start and goal positions/orientations become two points in the configuration space. The motion planning problem can now be stated as trying to find a continuous line in configuration space joining these two points, and not intersecting with any configuration space obstacles. This path is the plan of motion which describes how the vehicle must behave.

This can be reformulated as an optimisation problem—the goal is to choose a path connecting the two points which crosses the least number of obstacles (with the eventual goal of zero crossings). We now have a global optimisation problem for which genetic algorithms are ideally suited. The population is a set of paths. To encode the path, all that is needed is that each of the m intermediate nodes (each of which is n-dimensional) be stored. Storing each of the n components of a node in a bit-string of length p allows an entire path (one genome) to be encoded in nmp bits. The evaluation function acts upon a whole path and returns the number of obstacles crossed.

One half of the population is replaced at every generation; reproduction is performed by uniform random cross-over of the bit strings. We have written a program which implements this and has proved successful in tests. The figure below shows a path found by the program through a randomly generated three dimensional environment.



[1] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[2] Joe Pitt-Francis and Roy Featherstone. Automatic generation of sphere hierarchies from cad data. In *IEEE Int. Conf. Robotics & Automation*, Leuven, May 1998.