ABSTRACT

The Land Bridge – Cultural Algorithms Program Simulation (L-CAPS) system is a functional interface for learning group behavior using Cultural Algorithms (CA). It enables the Cultural Algorithms process to implicitly communicate with, modify, and evaluate autonomous game agents restricted to an external virtual world. The L-CAPS system extends the work of Kinnaird-Heether (Reynolds & Kinnaird-Heether, 2010), and is able to examine the viability of using CA in learning group behavior in games, as well as the ability for CA to be extended to more general game designs. Here it is applied to the learning of large group movement for the Land Bridge reality game. The goal of the Land Bridge Game is to recreate the virtual world of an ancient land bridge that extended across modern Lake Huron between 10,000 B.C. and 7000 B.C. Here, the authors are trying to predict the location of hunting sites, assuming that the positioning of those sites is related to the distribution of available game. In this paper they use L-CAPS to tune an algorithm to simulate the movement of large herds of caribou. In subsequent work the authors will then generate optimal paths for these herds through the ancient landscape using various path planning algorithms.

Keywords: Agent-Based Learning, Cultural Algorithms, Path Planning, Reality Games, Social Fabric, Swarm Intelligence, Virtual Worlds

DOI: 10.4018/jsir.2013040103
INTRODUCTION

One of the greatest obstacles to the creation of virtual worlds is to intelligently steer large numbers of agents through a virtual landscape in ways that makes their individual and collective movements look realistic (Karamouzas, Geraerts, & Overmars, 2009). Traditional approaches to this problem using path-planning algorithms have been extensively studied and have led to some very sophisticated algorithms. These algorithms were developed relative to robotic applications in static environments with few active agents (LaValle, 2006; Latombe, 1991).

In an interactive computer game, the virtual environments are dynamic and often populated by a large number of computer-controlled characters commonly referred to as “softbots”. The “bots” must move towards their own specific desired locations while being cognizant of both their surrounding environment and the other bots around them. As such, the algorithms needed to solve this problem must be capable of generating paths in real-time that look convincingly realistic to the player. Currently within the game development community there are several commonly used approaches:

1. One such approach is a grid-based method. Grid–based representations divide the landscape into a grid that can be searched using various algorithms such as the A* algorithm (Russell & Norvig, 2010). A* is a heuristic-based extension of the best-first algorithm that, under certain conditions, is guaranteed to produce optimum paths. In contrast, the worst-case complexity is exponential in nature. Therefore, A*–based methods are often computed off-line with commonly used paths stored in tabular form for retrieval during a game. The paths themselves tend to look unrealistic, and require some post-processing to produce more realistic trajectories;

2. Navigation meshes partition an environment’s terrain into polygons that are then created with an attached “waypoint” (De-Loura, 2000). Waypoints between polygons are connected together to form a mesh. The mesh is designed relative to the static environment and relative to the general locational goals that game characters will have. Path generation here is again done using the A* algorithm which has the same drawbacks as mentioned previously;

3. Reactive methods apply local environmental information in order to generate movement for characters towards their goal. In transferring local environmental information to a character, various algorithmic methods can be used to provoke a desired reaction from the character. The most popular approach uses a potential function. Goals and obstacles produce attractive and repelling fields, respectively. At any point in the space, the sum of the produced fields creates a movement vector towards the goal and away from obstacles. Unlike the approaches above that use A* to guarantee optimal paths, this approach provides no such guarantee. Characters can get stuck in sub-optimal points and their movement must be resolved, often through unrealistic movements. Such a case can be observed if the potential fields at a point cancel each other out. Then the character is stuck at a point with a 0 motion vector, and can remain stationary forever (Rimon & Koditschek, 1992);

4. Agent-based approaches are often used in computer graphics, where agent behavior can be specified in terms of rules, social forces techniques, or particle swarm methodologies (Rimon & Koditschek, 1992). These approaches typically make use of local information in order to produce their planning decisions, and work well in relatively open environments with a homogeneous group of agents. The recent
Improvement of Restaurant Operation by Sharing Order and Customer Information
www.igi-global.com/article/improvement-restaurant-operation-sharing-order/45759?camid=4v1a