Paul: Description

Adil Paul and **Eyke Hüllermeier**
Department of Computer Science
University of Paderborn, Germany
{adil.paul,eyke}@upb.de

Abstract

We present a CBR-based approach to an agent playing the Angry Birds game. We adopt a preference-based procedure to construct the case base. As the retrieve phase involves finding a game scene similar to a given one, we come up with a measure to assess the dissimilarity between two game scenes, which is based on solving appropriate assignment problems.

1 Our Approach

We describe in the following our preference-based method to construct the case base, how we represent the different objects involved in the game, and how we assess the dissimilarity between two game scenes.

1.1 A Preference-Based Approach for the Case Base Construction

We adopt a preference-based approach to construct the case base. The basic principle of the approach consists in randomly trying different solutions for a problem and maintaining the best one. In the context of Angry Birds, the steps of the process of the preference-based case base construction are as outlined in Figure 1 as a flowchart diagram.

1.2 Case Representation

We use the MBR representation of objects provided by the Angry Birds Basic Game Playing Software. For describing the rectangles, we adopt an interval-based representation. We reperesent a complete game scene through the set of the MBRs of all objects within it, together with their type when an object and colour when a bird. Shots, which constitute the solution part of a case, are represented in the form of a 6-dimensional vector as the case in the Basic Game Playing Software.

1.3 Case Retrieval

When the agent is playing, he gets a representation of the current game scene, searches the case base for the case with the most similar game scene, i.e. with the minimal dissimilarity, to the current one and adopts its shot.

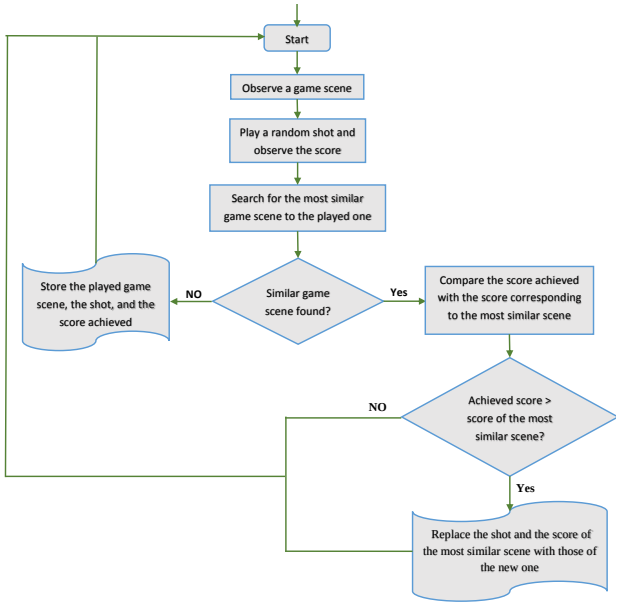

Figure 1: The steps of the case base construction process

We compute the overall dissimilarity between two game scenes as the sum of the dissimilarities between their individual components. For two slings it is just the dissimilarity between their MBRs. For the bird type, the dissimilarity is 0 if the types are equal and a constant otherwise.

Measuring the dissimilarity between two game scenes in each of the remaining components (hills, pigs, TNTs, and blocks) reduces to measuring the dissimilarity between the two sets of rectangles, with potentially different cardinality, corresponding to the MBRs surrounding them. This requires building pairs from the elements of the two sets, between which the dissimilarity is to be computed. The overall dissimilarity between the two sets is then the sum of the dissimilarities between all built pairs. We formulate the task of computing the dissimilarity between two sets of rectangles as a potentially unbalanced assignment problem, where the agents are the elements of one set, tasks are the elements of the other set and the total cost of an assignment is the overall sum of the dissimilarities between all built pairs.