Behavioral Learning of a Fuzzy Decision-Maker

Andrew R. Buck, *Student Member IEEE*, and
James M. Keller, *Life Fellow IEEE*
Electrical and Computer Engineering Department
University of Missouri
Columbia, MO 65211 USA
arb9p4@missouri.edu; kellerj@missouri.edu

***Abstract*—In this project, we aim to understand the behavior of a decision-making agent. In particular, we present the agent with an environment, represented as a fuzzy weighted graph, and observe the preferred routes between any two points in the graph. We assume the agent follows the principle of bounded rationality, implemented as an alpha-level OWA operator, to determine the path with the smallest perceived cost. We design an experiment to generate many such agents and environments, and to observe the preferred route between any two points. The agent parameters are then learned from the observed data using a genetic algorithm. We present our findings, in which we show a high degree of reproducibility between the original and learned agent.**

***Keywords*—behavior learning; fuzzy decision-making; bounded rationality; fuzzy rose diagrams**

I. Introduction

There are many applications in which a decision-maker is faced with several competing alternatives. Often, the distinguishing features of each alternative can be difficult to quantify numerically. In these cases, fuzzy numbers can be used to represent vagueness in the representation. One example is the problem of route selection in a partially-known environment. Consider the environment in Fig. 1. There are three route choices for the agent, each offering different tradeoffs. The shortest route goes directly over the hill, requiring a significant elevation change and a low path quality. An easier route goes around the hill on a paved path, but has a much longer distance. A third option balances distance and elevation by going through the woods, but requires a water crossing.

We can represent this environment as a fuzzy weighted graph as shown in Fig. 2. Here, we have represented the various features describing each path (distance, elevation, path quality, sun exposure, and water crossings) as a vector of fuzzy numbers attributed to each edge. We use the method of fuzzy rose diagrams presented in [1] to visualize this graph. Intuitively, the larger shapes represent larger values, with the imprecision of the fuzzy numbers encoded in the shapes. For full details, we refer the reader to [1].

The decision-maker in this example can choose between three alternative paths in order to reach the goal. We can represent the full set of alternatives by expanding the graph into a decision tree such as in Fig. 3. In this tree, the agent starts at node 1 and proceeds to node 2, which results in the accumulation of the features along the edge 1-2. Here the agent must make a choice, choosing either the path 2-3, 2-4, or 2-5. Assuming the

Fig. 1. Example scene for a decision-making agent with three route choices. "Over the Hill" gives the shortest distance, but also has the greatest elevation change. "The Long Way Around" is the easist route in terms of elevation and path quality, but the longest distance. "Through the Woods" is a balance of distance and elevation, also offering some shade, but requires a water crossing.

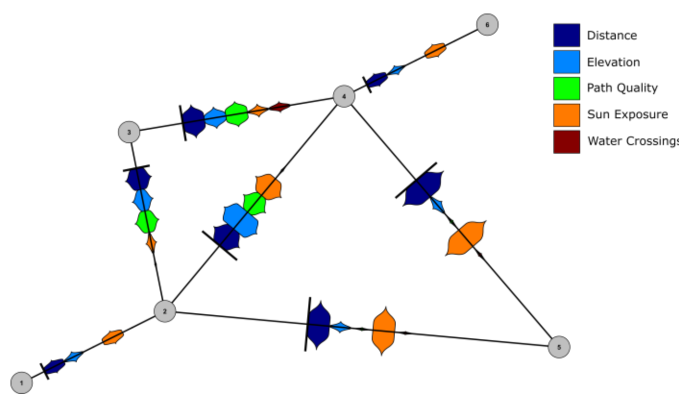

Fig. 2. A fuzzy weighted graph representation of the environment in Fig. 1.

agent cannot visit a node twice, each path has only a single route to complete the journey to the goal. Each time a node is visited, the features of the preceding edge are accumulated to the agent's total, resulting in the final values shown as the leaf nodes in Fig. 3.

The agent must now choose between the three alternatives. Given a vector of fuzzy feature values, $\boldsymbol{F} = [f_1, \ldots, f_N]$, we assume that the agent has an internal bias for each feature type,

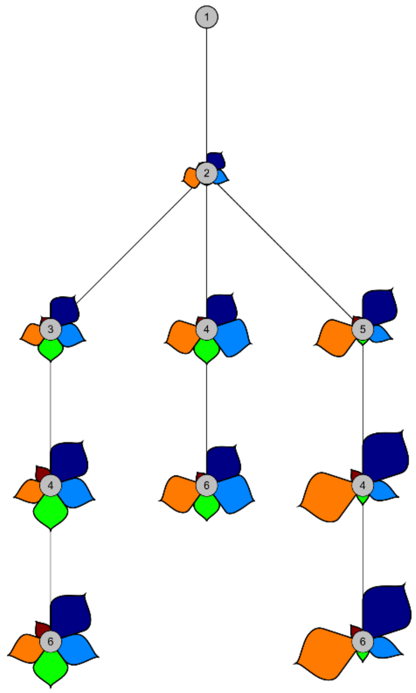

Fig. 4. A decision tree representation of the three possible routes between nodes 1 and 6 in the fuzzy weighted graph shown in Fig. 2.

$\boldsymbol{\beta} = [\beta_1, \dots, \beta_N]$, resulting in an agent-interpreted feature vector $\boldsymbol{A} = [a_1, \dots, a_N]$ where $a_i = \beta_i f_i$. Furthermore, we assume the agent cannot always make perfectly rational decisions and instead follows the principle of bounded rationality. We implement this using an alpha-level OWA operator, defined by a vector of weights $\boldsymbol{\omega} = [\omega_1, \dots, \omega_N]$. Essentially, this ranks and aggregates the agent-interpreted features to obtain a single fuzzy cost $C = \omega_1 a_{\sigma(1)} + \cdots + \omega_N a_{\sigma(N)}$ where σ is a permutation of $[1, N]$ such that $a_{\sigma(i)} \geq a_{\sigma(i+1)}$. The details of this computation are discussed in [2], particularly for the case when the agent bias terms and ordering weights are also fuzzy numbers. In this project, we consider only crisp scalar values for the $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ terms.

Finally, the agent must compare the computed cost values for all paths and rank the results. For this, we use the Liou and Wang index [3] which takes a single parameter λ, which can be viewed as an optimism/pessimism value. Numerically, the index calculates the integrals of the rising left and falling right portions of the fuzzy number C and uses λ as a linear weighting of the two extremes. When $\lambda = 0$ the agent is optimistic, considering only the smallest parts of the fuzzy number, whereas when $\lambda = 1$ the agent is pessimistic and considers the largest part of the fuzzy number. The result is a single crisp value that can be used to rank multiple fuzzy numbers.

For our three-route example, we consider three different agent profiles, shown in Fig. 4. The first agent has a strong weight associated with elevation and sun exposure, resulting in a weighting of the leaf-nodes from Fig. 3 that prioritizes these features. The resulting OWA ranking gives the first route ("Through the Woods") the lowest cost, making it the agent's top choice. Similarly, the second and third agents can be defined such that routes 2 and 3 are the top choices.

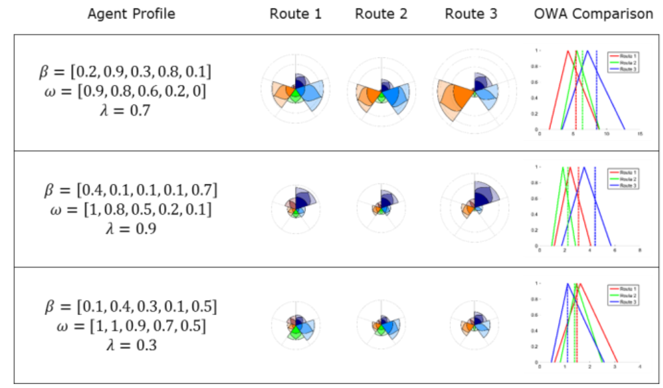

Fig. 5. Three example agent profiles, each with a different ranking of the three possible routes from the example three-route scenario.

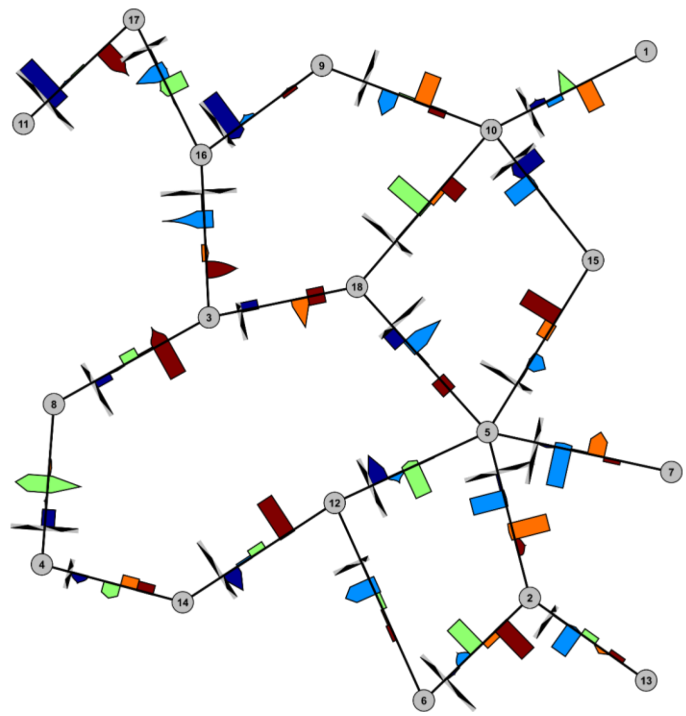

Fig. 3. Example of a fuzzy weighted graph generated for our experiments.

By representing an agent's decision making preferences as a vector of feature biases, OWA weights, and an optimism parameter, we can generalize how the agent makes decisions in an environment represented as a fuzzy weighted graph. The focus of this project is to recover the agent profile from a set of examples in which we observe the agent's choice from among a set of alternatives. The remainder of this paper describes our method for generating such a data set, a genetic algorithm for learning the agent parameters, and finally our conclusions.

II. Generating Decision-Making Data

To create our training dataset, we begin by randomly sampling 100 different agent profiles. Each agent profile can be represented as a vector $\boldsymbol{x} = [\beta_1, \dots, \beta_N, \omega_1, \dots, \omega_N, \lambda]$ where each $x_i \in [0, 1]$. For our experiments, we set $N = 5$. We also sort the OWA weights so that $\omega_i \geq \omega_{i+1}$. (Although this ordering is not strictly necessary, we have found that it leads to

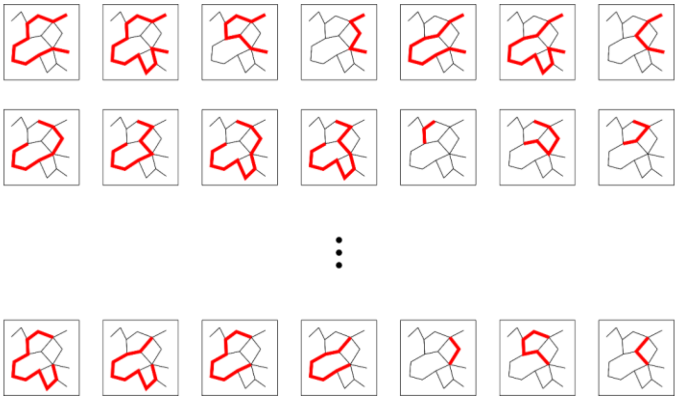

Fig. 6. A subset of the decision making scenarios generated for the graph in Fig. 5. Each row represents the set of all possible non-looping paths between a pair of points in the graph.

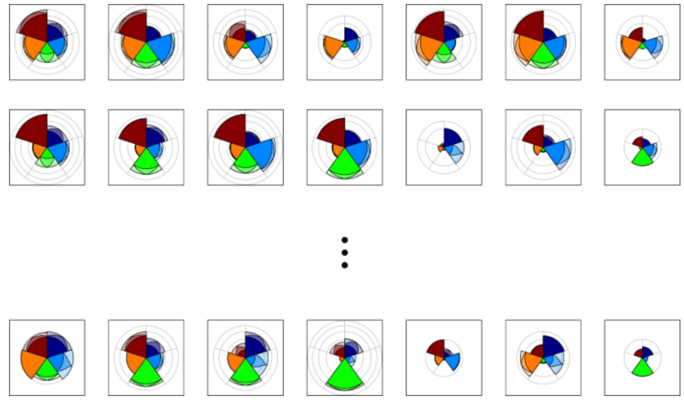

Fig. 7. Aggregation of the edge features for each path in Fig. 6. Each row repersts a decision scenario for which the agent must apply the OWA operator to determine the best choice.

more realistic agent behavior.) We then generate 3 random fuzzy weighted graphs for each agent like the one shown in Fig. 5. Each graph is a directed Urquhart graph [4] computed from a set of uniformly sampled vertices in the range $[0, 10]^2$ with a minimum separation of 2. The Urquhart graph is obtained by removing the longest edge from each triangle in the Delaunay triangulation of the vertices, and results in a good approximation of the relative neighborhood graph in which two points p and q are connected only if there is no third point r that is closer to both p and q than they are to each other. Each edge is assigned a random fuzzy weight vector with each feature randomly chosen from the range $[0, 10]$. When sampling a fuzzy number, first a type is chosen such that triangular fuzzy numbers are twice as likely as trapezoidal fuzzy numbers, crisp intervals are twice as likely as triangular fuzzy numbers, and crisp singletons are again twice as likely as crisp intervals. The final fuzzy vector is randomly permuted and scaled such that $f_i' = s^{i-1} f_i$ where we have chosen $s = 0.25$. This increases the sparsity of the feature vector, leaving only a few significant features. The resulting fuzzy weighted graph is highly irregular yet still structured like a road network, resulting in interesting agent behavior.

Next, we create the set of all possible decision scenarios for the agent within the 3 graphs. For each graph, we generate a decision scenario for each pair of points. We determine the set of all possible non-looping paths between these two points (Fig. 6) and aggregate the fuzzy edge features corresponding to each of these paths (Fig. 7). The agent then applies the OWA operator and determines a ranking of the paths, from which it can select the best choice (i.e. the path with the minimum cost as determined by the OWA operator and the Liou and Wang index). Our dataset consists of the original unbiased aggregation of features for each path in every decision scenario and the index of the path chosen by the agent. While the exact number of scenarios and the number of alternatives depends on the randomly generated graphs, we typically have several hundred scenarios for each agent, each usually having between 10 and 30 different alternatives.

III. Learning the Agent's Profile

For each agent generated by our method, we set up a genetic algorithm to learn the agent parameters from the observed decisions made in each scenario. First, we construct a chromosome representation consisting of a vector of floating-point variables $\boldsymbol{x} = [\beta_1, \dots, \beta_N, \omega_1, \dots, \omega_N, \lambda]$ where each $x_i \in [0, 1]$. We then define a fitness function which uses the chromosome variables to determine an OWA ranking for each decision scenario. The fitness is measured as the percentage of scenarios for which an incorrect choice is made.

We randomly divide the decision scenario data for each agent into 5 equal sets. Our training consists of a 5-fold cross validation for each agent in which we learn the agent parameters using 4 of the sets and test the accuracy on the remaining set. We use the Genetic Algorithm Toolbox for Matlab to learn the optimal agent parameters. Most parameters were left at their default values, with an explicit fitness limit to stop the algorithm when a perfect solution is found and a stall limit of 10 generations. The default algorithm uses a population size of 200, a uniform scattered crossover rate of 0.8, a Gaussian mutation rate of 0.2, and an elite size of 10.

The results of the algorithm for 10 selected agents are shown in Fig. 8. Here, we show the distribution of the learned parameters for the best solution in each of the 5 folds using a box plot for each parameter. The true agent parameters are shown with a dotted line. The algorithm is able to recover the agent feature weights $\boldsymbol{\beta}$ and optimism parameter λ with a high degree of accuracy, however the OWA weights $\boldsymbol{\omega}$ show a wide variation. This suggests that the aggregation operator may not play as significant a role as the agent-specific feature weights.

The best fitness evaluation for each generation averaged over all folds for the first 5 agents tested is shown in Fig. 9. The algorithm concluded within 30 generations and quickly approached the ideal limit of perfect fitness. The average accuracy of our algorithm averaged over 5 folds for each of 100 agents was an exceedingly high 99.34%.

IV. Conclusions and Future Work

This project has established a framework for decision-making in partially-known environments. We use the idea of a fuzzy weighted graph to represent a decision space and the concept of bounded rationality to implement a decision-making agent. We have shown that by representing each alternative as a vector of fuzzy features and applying an alpha-level OWA

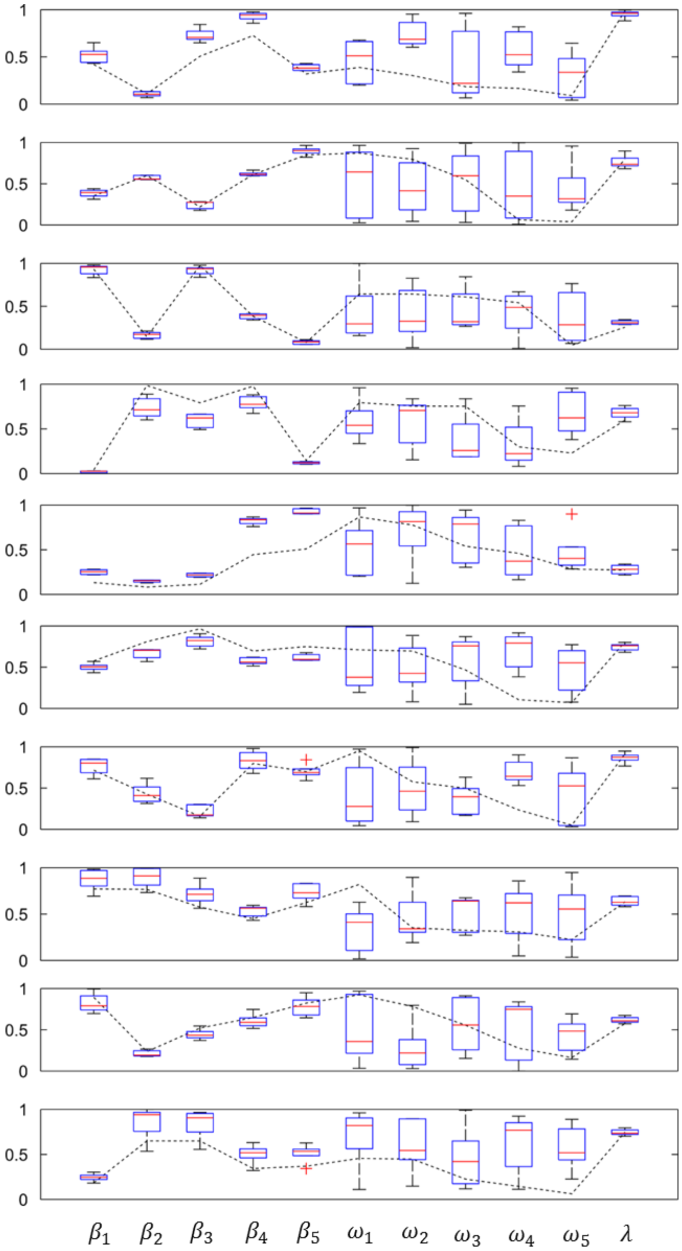

Fig. 8. Results of the genetic algorithm training shown for 10 selected agents. Each row shows a box plot for each parameter indicating the distribution of the best learned results for each of the 5 folds. The dotted line shows the true agent parameter values.

operator, we can design an agent model that is capable of choosing an action that minimizes the perceived cost. Furthermore, we have shown that we can recover the agent parameters by only observing the chosen actions in a set of decision-making scenarios.

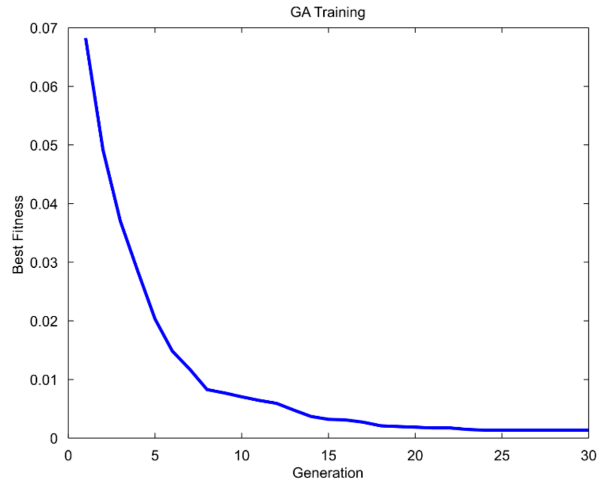

Fig. 9. Best fitness value for each generation averaged over all folds for the first 5 agents tested.

Although we have achieved a high degree of accuracy in our experiments, it should be emphasized that we use the same model for generating and learning the data. In real-world situations, the true model of the decision-maker is often unknown and the learned model is only valuable insofar as it is able to reproduce the observed behavior or give insight into the underlying mental representation of the environment. Further work in this area will focus on established methods of behavior learning like inverse reinforcement learning or apprenticeship learning. In these approaches, the problem is posed as learning a policy or reward function from an observed action sequence through a Markov decision process. Many existing models assume a linear mapping from environmental features to rewards. The inability of our method to accurately recover the OWA weights suggests that these linear models may be sufficient.

References

[1] A. R. Buck and J. M. Keller, "Visualizing Uncertainty with Fuzzy Rose Diagrams," in *Computational Intelligence for Engineering Solutions (CIES) , 2014 IEEE Symposium on*, 2014, pp. 30–36.

[2] A. R. Buck, J. M. Keller, and M. Popescu, "An alpha-level OWA implementation of Bounded Rationality for Fuzzy Route Selection," in *World Conference on Soft Computing*, 2013.

[3] T.-S. Liou and M.-J. J. Wang, "Ranking fuzzy numbers with integral value," *Fuzzy Sets Syst.*, vol. 50, no. 3, pp. 247–255, Sep. 1992.

[4] R. B. Urquhart, "Algorithms for computation of relative neighborhood graph," *Electron. Lett.*, vol. 16, no. 14, pp. 556–557, 1980.