2022 Fourth International Conference on Transdisciplinary AI (TransAI) | 978-1-6654-7184-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/TransAI54797.2022.00026

Learning to Evolve Procedural Content in Games Using Cultural Algorithms

Sarah Saad
Dept. of Computer Science
Wayne State University
Detroit, USA
fb5308@wayne.edu

Thomas Palazzolo
Dept. of Computer Science
Wayne State University
Detroit, USA
dz3786@wayne.edu

Chencheng Zhang
Dept. of Computer Science
Wayne State University
Detroit, USA
gl4705@wayne.edu

Robert G. Reynolds
Senior Member IEEE
Dept. of Computer Science
Wayne State University
Detroit, USA
and
Museum of
Anthropological
Archaeology
University of Michigan
Ann Arbor, USA
robert.reynolds@wayne.ed u

Ashley Lemke
Sociology and Anthropology
University of Texas
Arlington, USA
ashley.lemke@uta.edu

John O' Shea
Anthropological Archaeology
University of Michigan
Ann Arbor, USA
joshea@umich.edu

Cailen O'Shea
Family Life Center
North Dakota State University
Fargo, USA
cailen.oshea@ndsu.edu

***Abstract*—The procedural content of a virtual reality system is a key contributor to its success. However, it is often the case that the content needs to be adjusted. This may take place for a variety of reasons. This research investigates the possibility of the use of machine learning technology to facilitate the modification of a games content. Here, the Deep Dive system was designed originally to predict ancient site locations. Recently it was repurposed to be used as an educational tool to facilitate aspects of STEM education. This required the modification of the content to support this novel use. An evolutionary learning algorithm, Cultural Algorithm, is employed to facilitate the addition of the new content required for the educational application**.

Keywords— Procedural Content Generation, Evolutionary Algorithms, Cultural Algorithms, Virtual Reality, Stem education.

I. INTRODUCTION

Game design is a great model to the power of procedural content generation and its capabilities when generating content. Games need content to engage users and keep them playing[1], [2]. It is also used in game level design and the generation of entire landscapes. The size and scale of many large budget games presently takes hundreds of people and a lot of space to store this content. This has become a bottleneck especially since the prices of games have remained consistent over the years[3]. Procedural content generation provides the solution and allows games to be flexible and maintain the scale and amount of content they have.

Procedural content generation is not a new concept, but it is an important one. It can be a useful solution to problems. It is capable of being performed using a variety of methods including evolutionary, random search, and nature based. The content generated will take the form of what is needed to solve the problem at hand. Some applications of procedural content generation can include virtual testing, simulation, and game design. Simulations can be a great asset in fields like archaeology where it is not always possible to directly observe the environment being studied due to physical restrictions.

Some of the basic challenges in procedural content generation in games are as follows:

C1. The need for more realistic visualizations that reflect improved hardware capabilities for graphical rendering.

C2. The need to scale up the content generation process to support open worlds as they are explored by users.

C3. The need to support more complex visualizations associated with augmented and virtual reality.

C4. The need to generate content for a more dynamic AI engine. For example, in an artificial ecosystem there is the need to support food chains of organisms that reflect changes in organism behaviors over time.

C5. The need to produce more realistic dynamic group behavior-behaviors that reflects the dynamic changes in their environment using machine learning technologies such as Cultural Algorithms-technologies designed to model group adaptations over time.

C6. The need to provide a consistent game experience across a variety of different gaming platforms.

C7. The need to display the AI-generated content effectively in Graphic User Interface to facilitate decision-making.

While it is a basic challenge to produce procedural content in the first place that content often needs to be updated over time. There are several reasons for modifying the procedural content for an existing software system. They can include the following: improvements in supporting hardware technology; changes in supporting software; changes in required functionality; and shifting demographics and expectations. This

978-1-6654-7184-8/22/$31.00 ©2022 IEEE
DOI 10.1109/TransAI54797.2022.00026

paper represents a case study of a scenario in which the target functionality and demographics experiences a major shift over time. In this case there was shift from being a basic scientific tool for site prediction to one targeted towards developing critical thinking as part of a STEM educational program.

The Deep Dive Land Bridge simulation system was initially developed to aid underwater archaeologists in the discovery of ancient prehistoric sites located underwater in Lake Huron, one of the Great Lakes in the United States [3]. It utilized Artificial Intelligence and Virtual Reality to recreate the archaic semi-artic landscape and has facilitated the discovery of several ancient underwater sites [4]. The Land Bridge was above the Lake Huron water level for about 2,000 years from 10,000 B.P. to 8,000 B.P. Fig. 1 shows the location of the Land Bridge relative to the State of Michigan in the USA and Canada. The two cells on the bridge represent areas that are the focus of original exploration. They were initially selected due to their location relative to the widest part of the Land Bridge.

While the Land Bridge program was initially designed as an aid for the discovery of submerged prehistoric sites, the system's potential as a means for understanding traditional hunting practices, its use as an educational tool became rapidly apparent in two ways. First, it could be used to record hunters' observations about potential uses of the ancient landscape. Traditional Alaskan hunters were invited to enter the Virtual World and describe what they saw. By tracking the hunter's movements over the virtual landscape and by listening to their commentary, insight was gained into how traditional hunters view and conceptualize the landscape in general. In addition, their assessment of locations that might have served as sites for hunting structures and activities in the distant past were recorded. [5].

These results suggested that the system might be successfully repurposed to be a valuable educational tool. In order to test this theory out the system was used as part of the STEM high school curriculum in Alpena Michigan. The first group of high school students used the Virtual Reality system for two weeks. Their goal was to identify areas of potential hunter activity. During the process of using the system it became clear that the system was in need of more detailed procedural content.

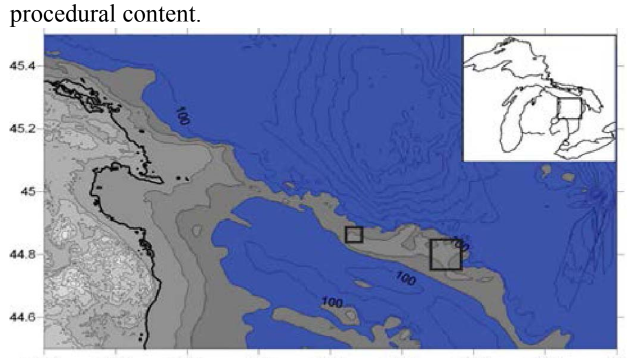

Fig. 1. The location of the Alpena Amberley Land Bridge. The explored areas are denoted as squares on the map.

To use this system as an educational tool rather than just a predictive one, the system needed to have a more immersive quality. In order to do that, the procedural content had to undergo some major revisions. How these revisions reflect each of the basic challenges described earlier is given below:

C1. It needs to display geographical and ecological content that covers over 3 billions1 by 1-meter squares and their change in content over several thousand years. The visualization needs to vary from coarse grained to fine grained in nature.

C2. It needs to support the movement of users as they move dynamically through the environment which is at most 8 miles wide and 60 miles long at a resolution of 1 by 1 meters at times.

C3. It needs to support both virtual and potentially augmented reality to recreate a more embedded experience for the user.

C4. It needs to support dynamic environment parameters such as food chains, and vegetational biomes that make up the ancient subarctic environment 10,000 years ago.

C5. It needs to produce a more dynamic model of herd behavior over a variety of biomes and herd sizes. One that provides a more immersive experience than the current herd models.

C6. The Landbridge models need to be portable enough to be run on a variety of different platforms such as a laptop, the Oculus Rift Virtual Reality System, and online. But also, generic enough to support the description of similar but different environments such as that in modern Alaska, an environment with a similar climate and ecosystem to that of the Land Bridge.

C7. The support of GUIs for both laptops and virtual reality interface that allow exploration of discovered artifacts by the user.

In this paper the focus will be on Challenges 4 and 5. More specifically the goal will be to integrate the predictions of several different herd mobility models for a variety of herd sizes and biomes and to subsequently express these predictions with the corresponding VR environment. This will provide a more compelling environment for student discovery and exploration. In Section II the three basic models of optimal herd behavior used in the Deep Dive system are described. This is followed in section III by the introduction of a Herd Stress model that captures the overall behaviors of the three algorithms.

However, the exact shape of the STRESS model is a function of the parameters that are employed for a given herd size. In section IV an Evolutionary mechanism, the Cultural Algorithm, is proposed to generate the optimal setting of the parameters for each of three representative herd sizes during their Fall movement over the entire Land Bridge. These settings are then compared against each other in order to identify the similarities and differences. Next, in Section V the actual pathways generated by each of the optimal settings for all herd size algorithm combinations. The goal is to identify those cells that are traversed by a herd regardless of the herd size or algorithm. These "hot spots" will then be inserted into the Virtual Reality system in order provide focal points for students to begin their explorations. Given that there are over three billion cells in the Land Bridge this will be an important contribution to the user experience. Section VI concludes the paper and describes future challenges.

II. THE MULTI-AGENT PLANNING FRAMEWORK FOR THE DEEP DIVE SIMULATION COMPONENT

Figure 2 give an overview of the overall Deep Dive system. It has three basic components: The Pathfinder MAP Simulation system; the Graphical User Interface (GUI) for the simulation system; and The Virtual Reality system.

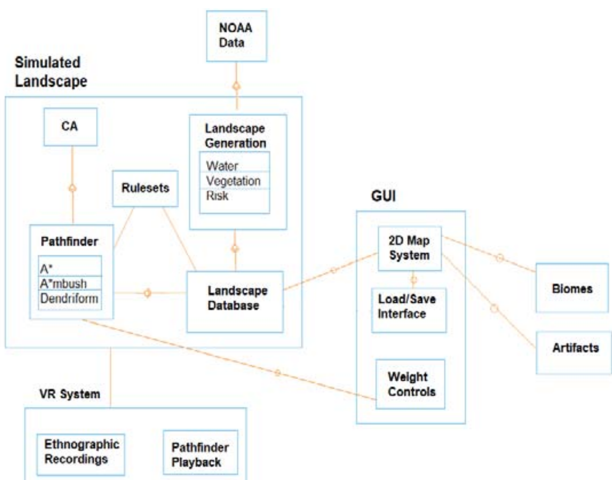

Fig. 2. The Overall Organization of the Deep Dive system.

The topographic data acquired from the National Oceanographic and Atmospheric database (*NOAA*) of the area was fed into the AI pipeline to *Generate* AI content via the Landscape. This created content includes the water level of various cells of the landscape to simulate which areas of the Land Bridge were above the current water level or not for a given year between 10,000 and 8,000 B.P. For a given year height map data for those portions of the landscape was calculated along with derived slope. Hydrologic information including the location ponds, swamps, and rivers that are present in the location at that time were then calculated. Given the location, water content, slope and sun angle the AI pipeline predicts the cells potential vegetation at each location on the Land Bridge. This information is stored in the *Landscape Database* for use by the *Pathfinder* system.

The basis for the simulation system is *Pathfinder*, a Cooperative Multi-Agent Planner (CMAPP). There are several deterministic general purpose MAP solvers available [9]. They include MAPR (MAP Planning by Reuse), CMAP (Cooperative MAP), mu-SATPLAN (Satisfiability based planning), among others. The different CMAP solvers can be classified by the mechanisms that they employ to address the planning process. The main features that can be used to characterize cooperative MAP solvers are:

1) Agent Distribution: The MAP process here involves multiple agents who are involved in the planning process either as active participants or as target for the planning process.
2) Computational Process: Whether the computational process is performed using a centralized monolithic processor or distributed among several processing units.
3) Plan Synthesis: This involves how and when the coordination activity is applied among agents. Coordination activities represent how information is distributed among agents and how their actions are to work together.
4) Communication Mechanism: How agents communicate with each other.
5) Heuristic Search: MAPs that use local heuristics allow individual agents to assess their estimated distance to their individual goals. Those with global heuristics calculate them for all the agents.
6) Privacy Preservation: Multi-Agent problem solvers can be distinguished in terms of their use of various privacy algorithms.

The CMAP, *Pathfinder*, used here was developed especially for the computational needs of this project. It is a monolithic, hierarchical Multiagent Planner based upon the A* Algorithm with the caribou agents as the target of the planning process. The planner uses a global heuristic to generate a single optimal path. This optimal path is used as basis for A*mbush. That algorithm decomposes the original path into waves of agents. The number of waves is given as a parameter. Then the results are given to Dendriform A* which decomposes the waves into smaller subgroups. The result is to generate a set of two-dimensional waypoints that support the optimal path across the Land Bridge. To keep the location of the found and predicted structures only those individuals with privileged access were able to display the exact locations.

The three algorithms comprising the Pathfinder approach are now briefly described:

A*: A* is a popular search-based pathfinding algorithm that's an adaptation of Dijkstra's Algorithm. The difference being an additional heuristic allowing it to attribute cost to actual and estimated distance from the goal. Since the algorithm calculated point by point it allows the caribou agents to freely traverse the landscape while focusing on effort, risk, and nutrition.

A Pseudocode*

```
Add pathStart to openNodes
    Initialize pathStart scores
    While (openNodes count greater than 0)
    {
```

currentNode = openNodes [0]

If (currentNode is goalNode)

{ assemblePath() and return true}

Remove currentNode from openNodes

Find currentNode's neighboringNodes.

ForEach(neighboringNode) calculate f and g score

If (neighborNode is not in openNodes) add to openNodes.

Else {adjust neighborNode's position in the openList based on total score}

}

Ambush: Also integrated into the system is A*mbush. A*mbush incorporates A* at its root. It uses the algorithm of A* but does so in separate waves instead of a single path. The waves are entered as a parameter, then the total herd size of Caribou is divided amongst the waves. The waves are then sent one after another with the nest wave entering the landscape as the last one completes its journey. Each wave consumes a certain proportion of available calories, leaving the remainder for the waves that follow.

Ambush Pseudocode:

for (generations=0; generations < ambushGenerations; generations++)

for (waypoint = waypoints-2; waypoint > 0; waypoint--)

{foundPath = AStar(waypoint, waypoint+1)

foreach(node in foundPath)

{Insert node in resultPath(generations) at index 0.}

} resultingHerd += calculateMigrationScore()

devourVegetation(foundPath)

}

Dendriform: Dendriform is the final algorithm used in the path planner portion on the Deep Dive system. It incorporates A* but also allows for branching during the exploration of the landscape. This means as the line of Caribou is traversing, they can divide on the spot allowing some of the herd to continue their path while the rest look for a separate path. This allows for more complex paths to be generated like the example in Figure 3.

Dendriform Pseudocode:

Calculate optimal A* path

Add starting point and ending point to node list.

While node list has more than two nodes {

checkForNewDivergencePoints

select last two nodes in node list and A*mbush Devour path section.

Remove last node in node list.

If last node in node list is not starting point:

Calculate optimal A* path to ending point.

}

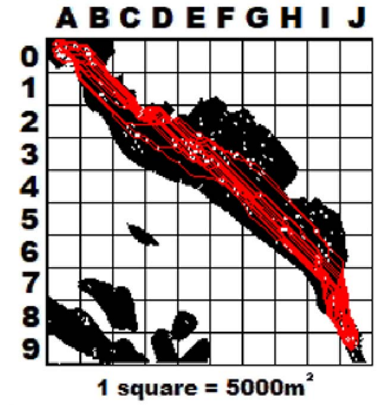

Fig. 3. Sample Run of Dendriform on the landscape.

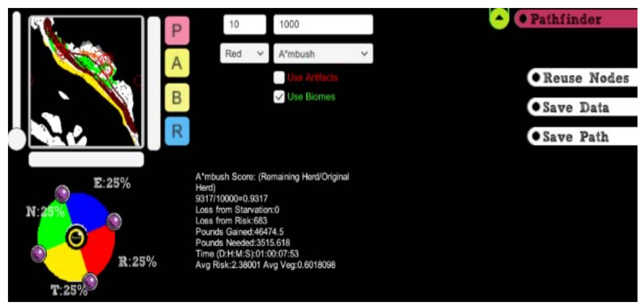

Fig. 4. Current Deep Dive GUI.

The Simulation system then communicates with the simulation GUI in two basic ways. First, the user interface displays a series of tabs through which the user may navigate to a given data set or select an experiment to run as shown in Figure 4. Maps can be viewed in a variety of data styles, such as biome data, topographical data, archaeological points of interest, ruleset hotspots, and so on. Pictured above, the user has selected to run six iterations of the A*mbush pathfinder, each wave being made up of 1000 caribou. The weight priority wheel on the bottom left allows the user to manually set the weights for Effort, Risk, Nutrition, and Time in the performance function. The priority weights control what will be important to the caribou in the current run. The green segment denoted by a "N" is the nutrition this will have caribou prioritize situations which will lead to an increase in calorie or food intake. The blue segment is effort ("E") increasing this priority will cause the caribou to avoid scenario which will lead to excess calories being spent for example going up a steep incline. The red segment is risk ("R") which influences caribou to avoid scenarios which would lead to a higher percentage of death. The last weight denoted by yellow is time, this prioritizes the amount of time it would take to cross the entirety of the portion of the land bridge simulated.

In the next section, the goal will be to simulate the three algorithms with different herd sizes and parameter weights over a subarea restricted by the largest of the two boxes in Figure 1. The model will be used as the basis for expanding the optimization activities from the small area to the entire Land Bridge in Section IV. This will be necessary to address the challenges associated with the change of user groups envisioned here.

III. THE PROTOTYPE HERD MOVEMENT MODEL

The DEEP DIVE system was originally targeted towards the prediction of sites in selected regions on the Land Bridge. Those regions are shown in Figure 1. Now that the focus has expanded to include the entire bridge as part of the educational user experience, the model will need to be expanded to produce optimal paths over the entire Land Bridge.

In the following, the model assumes a Fall migration pathway over the Land Bridge under the control of the four basic parameter components: Risk, Nutrition, Effort, and Time. Figure 5 below shows the effect that effort, time and nutrition can have on Caribou path planning. Part (a) of the figure shows the influence of nutrition on the created generated path. Instead of going straight through, they deviated toward the dark green area, indicating an area of high nutrition or food. In (b) there are two areas of higher elevation that the Caribou intentionally moved around, causing the path to move in a more circuitous way instead of straight through. Part (c) showcases the effect of incorporating both risk and time into the system. The generated path avoids the swamp areas which tend to take more time and are riskier. To better visualize the tradeoffs between the components STRESS charts were produced as a result of simulating the optimal path constructed by each algorithm for herds ranging in size from 50 to 300,000 across the Land Bridge. Herd size is plotted on the x-axis and the herd survival as a percentage is plotted on the y-axis. Figure 6 gives the chart for the Dendriform algorithm.

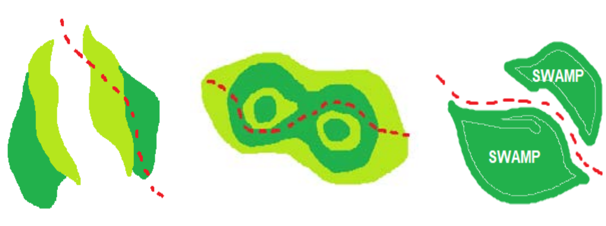

Fig. 5. Example paths showing influence of Nutrition, Effort and Time

In Figure 6 the STRESS curves for Dendriform are given for several different configurations of parameter weights. The Blue curve represents the scenario where the priorities are set to equal across the four priorities. The other four curves represent the scenarios where one of the components is set to 100% at the expense of all of the others. Notice that there are three basic phases. In the first phase, survival rates increase as herd size increases from 50 to around 1000. This reflects the principle of safety in numbers. The scenario that prefers risk over all other factors dominates in that phase. The next phase represents a plateau where the impact of adding new members is offset by an increase in members who are lost due to starvation. In the final phase the nutritional concerns start to dominate with herds above 8000. In that phase, the scenario that focusses only on nutrition is best able to ameliorate the observed reduction in survivability. On the other hand, it is the worst performer in the first phase relative to the others.

The curve that represents an equal weight for each of the components (25%) falls between the risk only and the nutrition only scenarios, This suggests that a truly optimal setting over all four components will have a contribution from each. Figure 7 then plots the STRESS curves for A* against A*Ambush for up to 10 waves for herds of up to 300,000. The same three phases emerge but notice that as the herd is broken up into smaller and smaller waves sizes the slope of the risk dominant portion is reduced. This is because the size of each wave becomes smaller and its contribution to reducing risk is reduced. On the other hand, the stable equilibrium with increased wave number is elongated since less new members are added in when the herd is broken down into smaller phases.

Figure 8 combines the STRESS curves for all three algorithms under the "all things being equal" assumption for the weights. In that Figure Ambush (1 wave) is the same as A*. As before, the more the herd is broken into waves the lower the slope for risk reduction. A* has the highest rate of reduction in risk since all of the individuals in a herd are used at once. Dendriform is a close second since it introduces all of the individuals into the simulation at once although they broke into groups. The plateau phase is the shortest for A* followed by Dendriform. For Ambush the more waves there are, the longer the equilibrium phase and the shorter the nutrition dominant phase.

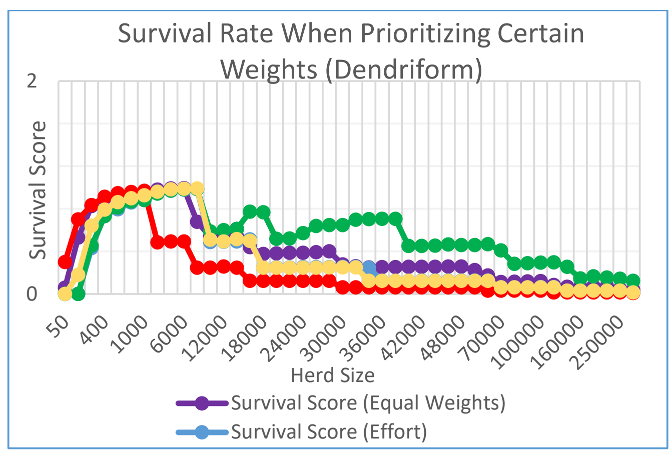

Fig. 6. Survival rates prioritizing different weights using Dendriform across various herd sizes.

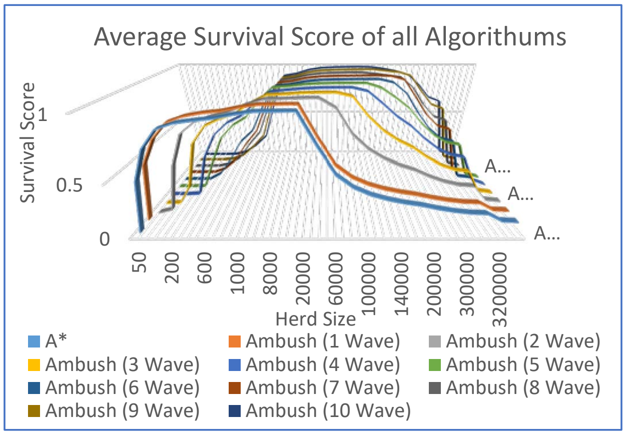

Fig. 7. Survival rates equalizing all weights using the A* and Ambush algorithms across various herd sizes(3D View)

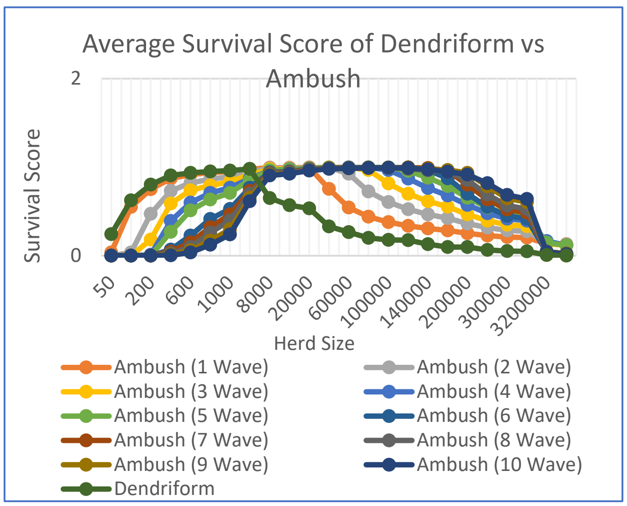

Fig. 8. Survival rates equalizing all weights using the A*, A*mbush and A*Dendriform algorithms across various herd size.

All three of the Algorithms exhibit variations on the three-phase model for the limited combinations of component weights that have been tested. Since some components are more influential in one phase then another, there is room to find a more optimal combination of the parameters. In the next section, a representative herd from each of the three phases is selected and an evolutionary learning algorithm, Cultural Algorithms, is used to produce an optimal solution for each of them.

IV. OPTIMIZING REPRESENTATIVE HERD SIZES

The goal is to identify those cells on the map that are visited by herds of all sizes and all of the algorithms. These locations can then be used as reference points for the user exploration of the Land Bridge. Since the representation consists of over a billion cells, this will make the task of exploration more feasible for the user. Of course, the optimal pathway will be different from one herd size to another. In addition, for any given herd size the optimization landscape can be very rugged. Finding the combination of weights that can produce optimal survivability for large herd sizes is a non-trivial task.

Figure 9 provides a description of the set of all possible combinations of weights for a herd of size 40,000. In the top graph each possible combination of the four weights is generated systematically with the color codes representing the percentage for each configuration. The bottom graph gives the corresponding survivability. It is clear that the topography for the optimal solution is very rugged and the computational cost for the optimal solution will be high for any herd size.

Alternatively, one can employ a machine learning algorithm, the Cultural Algorithm (*CA*) to produce a set of weights that optimize group survivability. The Cultural algorithm is a socially motivated algorithm developed by Reynolds [9], [10]. It's a means to solve problems in a complex system like the ones posed to the Deep Dive's path planner. It is described graphically in Figure 10. The CA is composed of a belief space and population space. Here the population is a set of experiments that employ different values weights for environmental parameters. The knowledge sources are housed

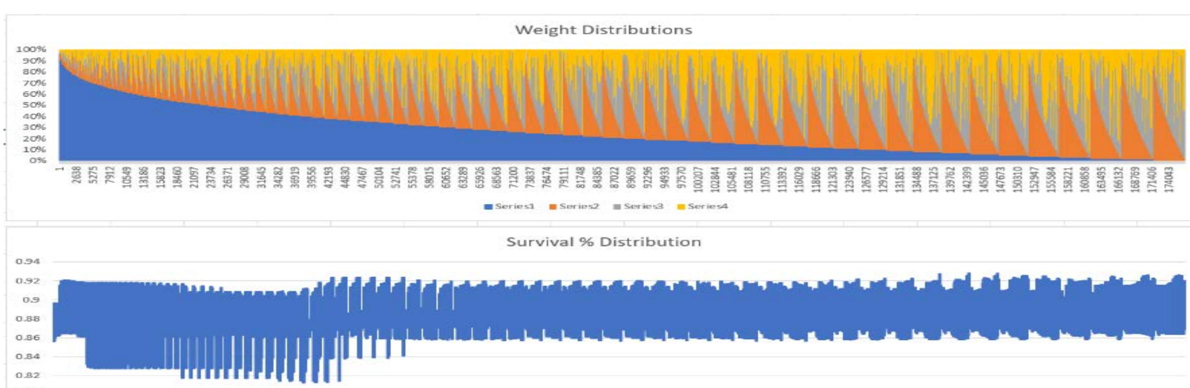

Fig. 9. Survivability as a function of different percentage combination of the 4 environmental parameter class. In the top graph the color codes are: Blue is effort percentage; Red is risk, Green is caloric intake; and Yellow is Time. The second chart gives the survivability of each combination.

in the belief space represent the knowledge of the population.

Individuals from the population are then influenced by belief space knowledge. Their resultant decision is evaluated by their relative fitness. Top performers are accepted into the belief space and used to update the knowledge there. Here the knowledge sources determine the optimal weights priorities of risk, nutrition, time, and effort can have on the system.

In addition, the user can place constraints on the generated pathways in two ways. First, they can require the path generated path to be constrained to pass through a set of manually set waypoints. Those points can be set by clicking on the map in the upper left-hand corner of the screen. Also, users can select rules that constrain the regions within which the paths can be placed. The Rule Selection screen is shown in Figure 11 below.

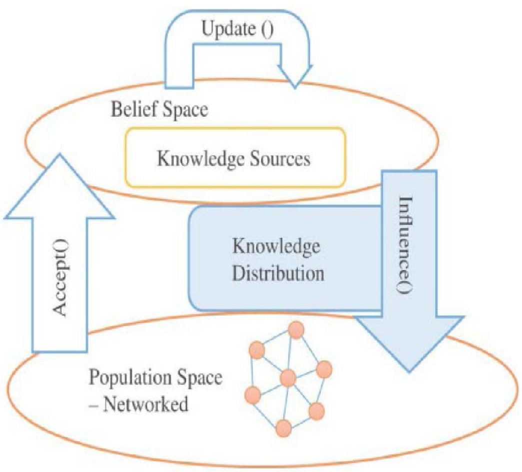

Fig. 10. Cultural Algorithm Representation.

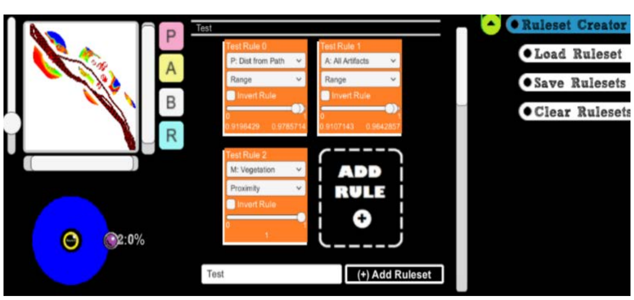

Fig. 11. The GUI screen used to display and modify heuristic constraints on generated paths.

The Cultural Algorithm will be applied to produce the optimum component weights for three representative herd sizes. One herd size will be selected from each of the three phases in the model. The Risk dominant phase is represented by a herd of population size of 8000. The plateau phase is represented by a herd of size 15000, and the Nutrition dominant phase is represented by a herd of size 25000. The caloric content of the Land Bridge cells are represented by biome that represents plant content on the Land Bridge while it was above water between 10,000 and 8,000 B.P.

The biome information used in these tests was the Sonnenberg Version 3 biome map, created from annotated biome map data, early predictions of the nature of areas, and later polygonal region data used to refine and correct the predicted areas. In figure 12, the Sonnenberg data can be seen with each biome represented as a different color, and the presence of water bodies indicated in black. The white color is the plains, a low-hazard, easily-traversable biome. Green represents marshlands, abundant in nutrition but high in risk and requiring more time to move through. Yellow is the sandy beach biome, with relatively low risk but low nutrition and a time penalty for moving in soft sand. Orange represents the rocky biome, with lower vegetation and light risk and time penalties due to the uneven, rocky surface of the landscape. Gray represents the northern cliff biome, which is immediately adjacent to a vertical drop which would likely be fatal to any caribou that fell from it.

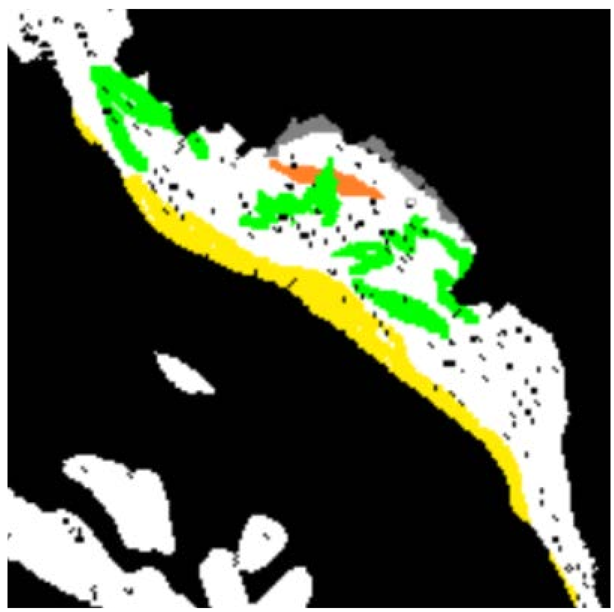

Fig. 12. A visualization of the Sonnenberg v3 Data

The Cultural Algorithm was then applied to produce an optimal configuration of weights for each of the three representative herd sizes when run with Ambush and Dendriform. A* was omitted since it was a special case of Ambush with just one wave.

In addition to the Cultural Algorithm runs, Optimal paths produced by Ambush and Dendriform for scenarios where all weights were set to be equal, and where each component was set to be dominant. The overall set of experiments to be run are given below.

TABLE I. OPTIMAL COMBINATIONS FOR AMBUSH AND DENDRIFORM GENERATED BY THE CULTURAL ALGORITHM

	Ambush 8	Ambush 15	Ambush 25	Dendri 8	Dendri 15	Dendri 25	AVG	STD DEV
Effort	10	19	21	26	43	41	26.66666667	11.84154645
Risk	73	12	7	40	31	13	29.33333333	22.69116323
Nutrition	16	63	66	15	25	35	36.66666667	20.77391527
Time	1	7	5	18	1	10	7	5.859465277

Experiment #1: All things being equal, the four variables were in balance, each contributing 25% to the overall influence.

Experiments # 2,3,4,5: Each variable became a dominant force, occupying 85% of the influence with the other variables contributing only 5%. This was done for each of the four variables.

Experiment #6: Utilizing the Cultural Algorithm to optimize the particular path-finder's given herd-size for survival, the resulting optimal weights were used for the sixth path.

This resulted in a grand total of 36 different paths being created by the various permutations between A*mbush and Dendriform. The datapoints of the map were then tested for how many times paths moved through this data point. The results of the experiments will now be discussed in the following section.

V. EXPANDING THE PROCEDURAL DATA

In this section the results of the Cultural Algorithm optimization are first discussed. Next, the path produced from the 36 separate runs were overlaid on each other across the entire Land Bridge. The paths were generated by three different scenarios: all things being equal; one component dominant over the others; and the optimal paths produced by the Cultural Algorithm. Each scenario was viewed to provide a different perspective on the produced paths. The resultant paths are then described, and the new procedural content translated into the Virtual Reality presentation.

In section III Herd simulations were done using two different scenarios, equal weights, and one dominant. Here, the weights associated with the optimum paths associated with representative herds from each of the migration algorithms are presented. The weights are given in Figure 13 below. Each column is a particular algorithm herd size paring. Ambush 8 stands for a herd of size 8000 controlled by the Ambush algorithm where there were 8 waves of 1000 animals each. Likewise Ambush was run with herd sizes of 15,000 and 25,000 respectively with waves of 1000 apiece. The last three column are runs of Dendriform for herd sizes of 8,000, 15,000, and 25000. In Dendriform herds can break into arbitrary sizes based on the component values. Each row represents the weight of the component associated with optimal path for the herd and algorithm pair.

The first thing to notice is that for Ambush, the was more indicative of a dominant strategy than all things being equal. Recall that Ambush had a slower rate of reduction in with increased herd size since the herd was only added incrementally rather than all at once as with A*. It is important for the algorithm to emphasize Risk based decision in order to compensate especially when the herd size is in the Risk phase. As the herd size increases the emphasis on Risk is reduced. Nutrition on the other hand is the least important for a small herd but becomes increasingly important as the herd size increases. In conjunction with an increase in nutrition there is an increased emphasis on effort with increased herd size since it will take more effort for individuals to find the necessary resources. Time is less of an issue since the waves are synchronized and there is little room for variation in time.

With Dendriform the entire herd is starts at once so even with the smallest herd size, there is less need to be concerned with Risk than with Ambush. The emphasis on Risk subsequently decreases with herd sizes of 15,000 and 20,000. On the other hand, Nutrition is less important for the small herd but increases with increased herd size but is never as important as it is with Ambush. This is due to the fact that herd fission and fusion is based on resources and the efforts to access them. Therefore, effort increases with increased herd size as well. Timing is also more important with Dendriform than Ambush since the sequencing of fission and fusion can impact both Risk and Nutrition. It is particularly important for the smallest herd size since it is important that the sub herds not be too small or too large. Overall Time is more of a factor in Dendriform movements than for Ambush.

In summary the optimum weights generated for Ambush and Dendriform by the Cultural Algorithms were distinctly different from the "all things being equal" setting. However, the weights did reflect the emphasis associated with the phase from which the herd size was selected.

Next the paths generated by all 36 configurations are overlaid on each other and the number of times that a path crossed a cell is counted. Figure 14 gives the collected hit data for the 36 generated paths. The cells are color coded such that the most frequently traversed are bright red and those least traversed are more blue in color. While interesting, our goal is to extract 'hot regions" that can help focus the user's explorations on the land bridge. This particularly important since the overall Land Bridge contains over 3 billion cells but from the graph it is clear that there are regions that are not traversed. Without some guidance the user might wander fruitlessly over the landscape.

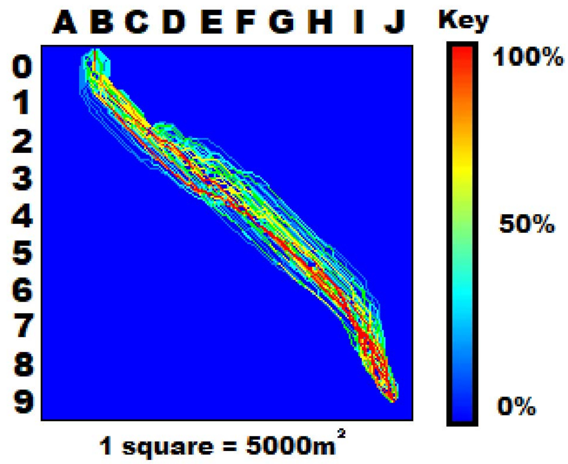

Fig. 14. The collected hit data for 36 different path-runs.

The question then is, are there regions that are traversed by all paths regardless of the optimally of the configuration, the algorithm, or the herd size? Figure 15 gives those cells that are traversed by all paths. What is interesting is that there are some segments that are disjoint from other segments. These are regions that are part of several different pathways, areas that can be reached in several different ways. Those are particularly interested areas to explore.

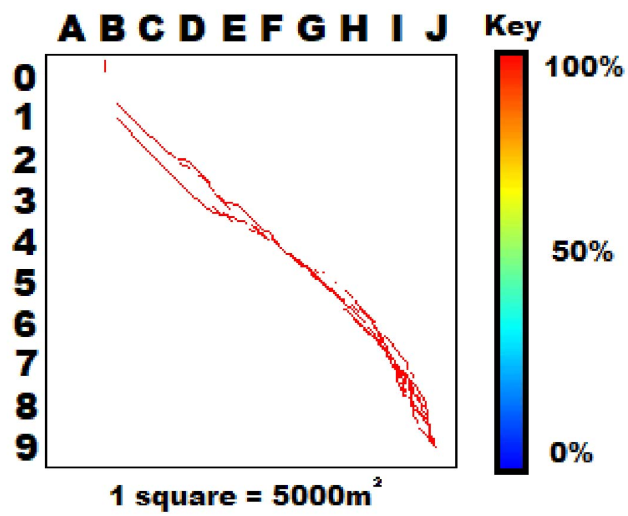

Fig. 15. Squares that are traversed by all paths for all weight configurations, algorithms, or optimality.

The calculated paths can now be overlaid onto the Virtual Reality landscape and used as a vehicle for student users to explore and navigate through the environment. Figure 16 gives an example of such an overlay. The content can be used not only to direct user investigation but as directions for virtual caribou to follow as show in Figure 17.

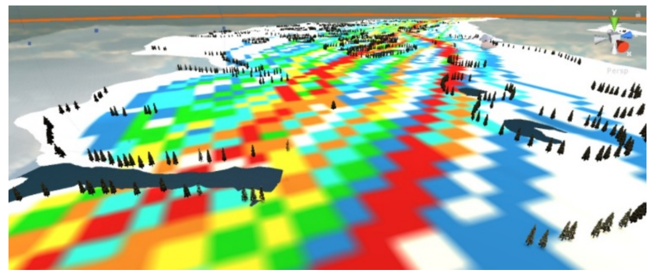

Fig. 16. Generated procedural content overlaid onto the Land Bridge Virtual Reality landscape.

Fig. 17. Caribou group movement along a path.

VI. CONCLUSIONS

Procedural content generation is an important and challenging activity. Yet, even after content is generated changes in the game or system may require corresponding changes in the procedural content. While such adjustments can be done manually the use of machine learning techniques to aid in the automatic adjustment of content is promising. Here the Deep Dive Land Bridge VR system was designed to aid archaeologists in the prediction of ancient sites. Recently a version of the system was repurposed to support STEM related activities by high school students. Initial use in the classroom suggested that certain changes in the procedural content were necessary. Specifically, the users needed more direction in terms of exploring the Land Bridge. The goal of the work described here was to modify the content to support those more directed activities.

In this paper, several algorithmic models of herd movement were described and simulated using various parameter configurations. The result was to produce stress curves that showed how survivability was affected by herd size for each algorithm. From those curves a three-phase model of survivability was produced, and three representative herd sizes were selected for path optimization using an evolutionary algorithm, Cultural Algorithms. The optimal combinations of parameters produced by the Cultural Algorithm was used along with other parameter combinations to produce 36 pathways. The combination of these pathways produced regions of high activity that were then used not only to help students explore the landscape but also to direct corresponding caribou movements in VR.

REFERENCES

1. Reynolds, R. G., O'Shea, J., Che Z., Gawasmeh, Y., Meadows, G., and Fotouhi, F., "The Agile Design of Reality Game AI", in Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies: Intelligent Techniques for Ubiquity and Optimization, IGI Global Press (2011).
2. Fogarty, J., Reynolds, R.G., and Salamayeh, A., and Palazzolo, T., "Serious Game Modeling of Caribou Behavior Across Lake Huron using Cultural Algorithms and Influence Maps", Ed. John O'Shea and E. Sonnenberg, Archaeological, Ethnographic and Paleo-Environmental Perspectives on Caribou Hunting in the Great Lakes: Maps, Directions and Accommodations, University of Michigan Museum of Anthropology (2015).
3. Palazzolo, T., Reynolds, R.G., and Stanley, S., "Exploring Virtual Worlds with Cultural Algorithms: Ancient Alpena-Amberley Land Bridge", in Cultural Algorithms: Tools for the Engineering of Social Intelligence into Complex Systems, pp: 203 – 271, IEEE Wiley Press, Hoboken (2020).
4. O'Shea, J, Lemke, A., Sonnenburg, E., Reynolds, R.G., and Abbot, B., "A 9,000 Year-Old Submerged Caribou Hunting Structure in Lake Huron", Proceedings of the National Academy of Science (2014).
5. Reynolds, R.G., Palazzolo, T., Lemke, A., O'Shea, J., Saad, S., Zhang, C., "Deepdive: The Use of Virtual Worlds to Create Ethnographies of Ancient Civilizations", Human Computer Interaction International Conference, Washington D.C. (2021).
6. The IB Learner Profile, https://www.ibo.org/globalassets/publications/recognition/learnerprofile-en.pdf.
7. McLeod, S., & Shareski, D., Different schools for a different world. Solution Tree Press, Bloomington IN (2017)
8. Culturally Responsive-Sustaining Education Framework, http://www.nysed.gov/crs/framework.
9. Reynolds, R.G., Culture on the edge of chaos: Cultural algorithms and the foundations of social intelligence, 1st ed. Springer International Publishing Basel, Switzerland (2018).
10. Reynolds, R.G., Cultural Algorithms: Tools for the Engineering of Social Intelligence into Complex Cultural Systems, Wiley-IEEE Press, Hoboken (2021)
11. Koenig, S., Fast Replanning for Navigation in Unknown Terrain, in IEEE Transactions on Robotics, vol. 21, no. 3, pp. 354-363, (2005)
12. Millington, I., AI for Games. Third edition. CRC Press, Boca Raton, FL (2019)