A Comparison of Incremental Community Assembly with Evolutionary Community Selection

Daniel Ashlock and Meghan Timmins

Abstract—Given a set of potential species and a replicator dynamic model of their interaction, the community assembly problem seeks the maximal set of species that can co-exist indefinitely without extinction. In this study we compare a standard model, which assembles a community one species at a time, with an evolutionary algorithm that selects sets of species directly. The comparison is performed using a standard competition model. The system is tested with three different available species pools of one hundred species. The diversity of communities located with the evolutionary algorithm substantially exceeds that of those located by serial addition of single species. In agreement with past research, the serial species addition algorithm located communities that, while not the largest, were highly resistant to invasion by a single additional species. A comparison of the diversity between the communities located by the two algorithms demonstrated that the evolutionary algorithm located a very much larger variety of community types. For all three species pools, the communities found in different runs of the serial species addition algorithm shared large common cores of species.

I. INTRODUCTION

This study compares two algorithms for locating large stable communities within a space of ecologies within a pre-defined set of possible species. We begin with definitions needed to pose the problems being solved and define the search algorithms that are the focus of the study.

Definition 1: Given a collection of species with interactions described by a competition matrix that drives a replicator dynamic model of species density over time, a community is stable if no members of the community go extinct.

Definition 2: The community assembly problem consists of the location of the largest possible stable community within the type of situation described in Definition 1.

Characterizing ecological communities as dynamical systems is a long-standing technique for the mathematical modeling of ecosystems [18]. Fukami [7] considered the impact of the size of the species pool on community assembly and found that resistance to invasion is more common when the available species pool is small. The strength of pairwise species interactions is considered in [13] where the authors find that a community that is resistant to invasion typically has weaker interspecific interactions; strong interactions open the door to invasion.

In [16] the authors introduce the serial assembly method for the community assembly problem. They find that some stable communities cannot be reassembled from their member species - the only way to arrive at these communities is via an intermediate community, some of the members of which become extinct as the community assembly proceeds. They also find that invasion resistance remains low during community assembly until the process lands at a state the authors term an “endpoint”. The endpoint state is one that exhibits a high resistance to invasion. This study confirms this phenomenon in all three artificial ecologies examined.

Both [20], [22] studied species diversity and ecosystem productivity of grasslands. Both studies found that diverse ecosystems, those with large stable communities, had higher biomass productivity than monocultures. Studying the ability of grasslands to survive invasion and disasters, [20] found that less diverse communities were better able to survive disruptions. These results are both potentially important for modeling and planning grassland restoration projects.

The author of [9] notes that most models for community interactions are based on resource competition or predation. These models strongly predict that the number of species should not exceed the number of critical limiting resource factors in the ecology. A model is presented incorporating positive interactions. This model demonstrates that, when such interactions are included in a model, the number of species in a stable community expands well beyond the bound suggested by the number of critical resources.

In [17] they authors propose mechanisms for understanding the inverse relationship between ecosystem diversity and stability. This work supports the conclusion from [13] that stronger interactions can be a destabilizing factor. Other mechanisms that help diversity to decrease the stability of a community are differential reaction times to disturbances and differential degrees of response to perturbations by different species.

An important factor in the stability and diversity of ecosystems is that of the spatial distribution of species. The ability of invasive species to colonize dry grasslands, for example, depends on the spatial distribution of the species in the original ecological community [24]. Disturbance also plays an important role in maintaining the biodiversity of a community. Gopher mounds, for example, clear all plants from small patches of land. The differential rate of recolonization of the mounds enhances the species diversity of grassland communities [23].
The remainder of this study is structured as follows. Section II provides background on the community assembly problem and specifies the replicator dynamic model used in this study. Section III gives the design of experiments. Section IV gives and discusses results while Section V draws conclusions and outlines potential next steps.

II. BACKGROUND

The competition model used to check the stability of a community of species is a difference equation replicator dynamics model based on a competition matrix $M$. The entry $M_{i,j}$ specifies the degree to which the presence of species $i$ is helped or hurt by members of species $j$. Positive numbers indicate help, negative numbers indicate harm. The matrix is initialized to have all species in a putative community equally common and then the replicator dynamics are run for 2000 time steps and surviving species are counted. The number of surviving species is the fitness value for a putative community. The updating equation for species $s_i$ at time $n+1$ is a logistic model with competition given by Equation 1.

$$ s_i(n+1) = M_{i,i} \cdot s_i(n)(C - s_i(n)) + \sum_{i \neq j} M_{i,j} \cdot s_j(n) $$

(1)

After updating, the values obtained from the difference equation for each species are normalized. The individual species values are divided by the total to form a species density profile. Any species whose density drops below a critical threshold of $\tau = 0.001$ is removed. The model selects putative communities from a set of $N = 100$ potential species ($M$ is a $100 \times 100$ matrix) making the removal threshold $\tau$ relatively soft. The carrying capacity is set to $C = 0.99$.

The term $s_i(n)(C - s_i(n))$ in Equation 1 represents logistic growth [12] of a species existing in isolation. This model of growth makes a population self-limiting. The number of individuals in the next generation is proportional to the product of the number in the current generation and the carrying capacity minus the number of individuals in the current generation. This forces large populations (those near the carrying capacity) to have low growth rates before interactions with other species are considered. The carrying capacity parameter $C$ is set near one making it possible for a single species to dominate an ecology. Reducing the carrying capacity parameter would make it more likely that larger communities would remain stable. Examples of the way density can vary during addition of single species in a reduced model are shown in Figure 1.

A. Selection of the Competition Matrix

Discussion of the properties of competition matrices may be found in [18], [10]. The range of values in the matrix represent the interaction strength between species and, along the diagonal, the growth rate for small populations (the choice of logistic growth forces large populations to grow slowly due to self-competition). In [3] the effect of growth rates on the community assembly problem are studied.

The entries of the matrix $M$ are generate using a log-normal distribution. If $N(\sigma, \mu)$ is a normal distribution with standard deviation $\sigma$ and mean $\mu$ then diagonal entries of $M$ are generated with the distribution

$$ M_{i,i} \leftarrow e^{N(0.2, 0.0)} $$

(2)

while off diagonal entries are generated with the distribution

$$ M_{i,j} \leftarrow e^{N(0.2, 0.0)} - 1.0 $$

(3)

except that the log normal numbers are reduced to 2.0 if they are larger than 2.0. This truncation was imposed to prevent rare, very large values in the competition matrix.

Serial Species Addition Algorithm

A flowchart for the SSAA is shown in Figure 2.

B. Algorithms

This study compares two algorithms, a serial species addition algorithm (SSAA) [3] and the community selection evolutionary algorithm (CSEA), which is introduced in this study. Three sets of 100 virtual species are used to compare the algorithms, each generated randomly as outlined in Section II-A. Algorithm 1 summarizes the SSAA; a flowchart is given in Figure 2. It is a form of random walk in the space of possible communities, except that when a species is added, extinction of the new species or other community members may take place.
Fig. 1. Shown are exemplary species density tacks during 500 time steps of the replicator dynamics model for a ten-species competition matrix. The upper example shows complex dynamics while the lower example rapidly reaches a stable state. Reduced models are depicted because depictions of the full model lack visual clarity.

**Algorithm 1: Serial Species Addition Algorithm**

**Input:** A competition matrix $M$  
**Output:** A largest stable community  
**Details:**  
Initialize a community with one species  
Repeat 1000 times  
   Add a species not in the community  
   Initialize present species to uniform density  
   Update replicator dynamics 2000 times  
   Update community to contain surviving species  
   If community size is new maximum, save it  
End Repeat  
Report largest community located  

Except for its fitness function, the evolutionary algorithm is a standard binary gene algorithm. The binary genes are length 100 with a one denoting the presence of one of the virtual species and a zero denoting its absence. Fitness of a gene, and the community it defines, is the number of species remaining after 2000 updatings of the replicator dynamic model. As in the SSAA, the replicator dynamics are initialized with all present species at the same density. The algorithm is steady state [21] using size seven tournament selection. This model of evolution selects and sorts seven population members by fitness. The two most fit members of the tournament are copied over the two least fit. The copies then undergo two-point crossover and mutation. Mutation flips 1-5 bits in each of the children, with the number of bits flipped selected uniformly at random. A population of ten putative communities is used. These values were selected by preliminary experimentation.

**III. Design of Experiments**

Three different competition matrices were used to compare the SSAA and the CSEA. Preliminary testing of the SSAA showed that it commonly settled to a single value after about 500 updatings (additions of a single species). Based on this, 100 instances of the SSAA were run for 2000 updatings on each of the competition matrices. The largest stable community found in each walk was saved. The evolutionary algorithm was run 10 times for each competition matrix and the best community found was saved. The evolutionary algorithm was run for 5,000 timesteps, each consisting of a single instance of tournament selection, with summary statistics saved every 50 mating events. The statistics saved were population mean fitness, the sample standard deviation of fitness, and the best fitness. The length of time to run the evolutionary algorithm was chosen so that, given that each tournament selection involves two fitness evaluations,
both algorithms were permitted 100,000 fitness evaluations per virtual ecology.

A. Evaluation Tools

A common problem in evolutionary computation is that of figuring out if the algorithm has been run long enough. In this study each algorithm was permitted 100,000 fitness evaluations to permit a fair comparison. Given that the algorithms are very different, it would be nice to have an assessment that tells us if running the algorithm for some additional amount of time might have helped. The time of last innovation assessment can provide some perspective on this issue.

For all of the experiments we examine the fitness tracks of the experiments and compute the last time the current best fitness (CSEA) or current fitness (SSAA) changes. This time of last innovation is then divided by the time the algorithm ran to obtain a fractional time of last innovation. Figure 5 shows the results of this assessment. Time of last innovation serves as a much more compact way of displaying information commonly used to assess if an algorithm was run long enough. It replaces plots of fitness-versus-time which are examined to see if fitness increase has stopped.

IV. RESULTS AND DISCUSSION

Figure 4 shows the distribution of largest community sizes located with each algorithm for all three competition matrices used. For the first competition matrix, with a single exception, the values located by the CSEA algorithm were superior to every value located by the SSAA algorithm. For the second competition matrix the lowest value for the CSEA is 75, the highest value for the SSAA is 75, and each occurs only once. For the third competition matrix there is some overlap but the results for the CSEA are much better than those for the SSAA. Given that the number of fitness evaluations used were kept the same for both algorithms, this is strong evidence that the CSEA is better at locating large, stable communities.

The number of fitness evaluations allocated were set by characterizing the behavior of the SSAA. Figure 3 shows fitness tracks for exemplary runs of each algorithm for each competition matrix. The SSAA is a non-elitist sampling algorithm. The run of the SSAA using the first competition matrix shows the algorithm obtains a fitness higher than its final fitness early in the run and then falls back. The evolutionary algorithm is elitist, and so the best fitness must be non-decreasing. This would seem to make it easier for the CSEA to become trapped in a local optima - but there is another factor. Crossover and up to five mutations permit the evolutionary algorithm to examine a much larger number of designs derived from the current best design in comparison to the SSAA.

The number of fitness evaluations that the algorithms were permitted to use was set by examining the behavior of the SSAA. The three exemplary runs of the SSAA show that it arrives in a stable state well before the end of the serial assembly run. This, in turn, means that the evolutionary algorithm may not be “done” at 5,000 fitness evaluations. Figure 5 gives the fractional time of last innovation of both algorithms on all three competition matrices. This metric demonstrates that the CSEA was still finding new, better results long after the SSAA had hit a stable final value. It is very likely that running the evolutionary algorithm for a longer time may yield better results.

A. The Character of the Communities Located

The information on time of last innovation in Figure 5, while suggesting that the CSEA could benefit from more time, also suggests that while the communities located by the SSAA algorithm are not the largest they are quite stable and have a very high resistance to invasion. If there is a hyper-stable community that is easy to locate with the SSAA algorithm this, in turn, raises the question of how often a particular community is located. Since the competition matrices are themselves randomly generated objects the actual identity of a stable community is not particularly important, rather it would be interesting to know the degree to which the same or similar communities are being located. The vertical span on the box plots for SSAA results in Figure 4 is quite small, suggesting that communities of a relatively uniform size are being located.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>SSAA</th>
<th>CSEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.38 ± 0.13</td>
<td>21.0 ± 1.50</td>
</tr>
<tr>
<td>2</td>
<td>1.90 ± 0.12</td>
<td>20.4 ± 0.91</td>
</tr>
<tr>
<td>3</td>
<td>3.40 ± 0.10</td>
<td>26.2 ± 1.47</td>
</tr>
</tbody>
</table>

**Table 1**

Shown are 95% confidence intervals on the average Hamming distance between the largest stable communities.

A simple scalar measure of difference between the communities located is the Hamming distance. The Hamming distance between two communities is the number of species that appear in one community but not the other, in both directions. If we think of a community as a binary string that is one when a species is present and zero when it is not then this is the classical Hamming distance on binary strings. Table 1 summarizes the distribution of distances between pairs of communities located with each algorithm for each competition matrix. The communities located by the CSEA are far more diverse than the ones located by the SSAA. The average distance for the SSAA using the second competition matrix is about two - with a mode community size of 69. This means that very similar communities, with minor variations, are being located over and over.

B. Biological Justification of the Algorithms

The usual reasoning behind the use of the SSAA algorithm is that new species arrive in an ecology individually. The rhythm of adding a new species, permitting the ecology to come to equilibrium and then repeating captures the essence of the algorithm. The CSEA, on the other hand,
Fig. 3. Shown are exemplary fitness tracks from the SSAA and CSEA experiments, one for each algorithm and competition matrix.
simply directly evolves selections of species from which communities are formed by running the replicator dynamic algorithm. It contains no model of how the community forms and so may be locating large stable communities that might not occur in practice.

While invasions by single species is a common way for new species to arrive, floods, fires and other natural disasters can cause forced simultaneous migration of multiple species. The importation of fire ants into the southern United States brought not only the ants but the soil they lived in as the contents of large concrete pipes imported from Argentina [5]. Presumably a number of other species were present in this soil. These examples show that single species addition is not the only mechanism of community assembly, making the CSEA a reasonable way to search for maximal stable communities.

V. CONCLUSIONS AND NEXT STEPS

This study serves as proof-of-concept for using an evolutionary algorithm to attack the community assembly problem. As far as the authors can tell this is the first used of an evolutionary algorithm on the community assembly problem. The CSEA found substantially larger communities than the SSAA for all three competition matrices tested. The degree of advantage of the CSEA over the SSAA showed some dependence on the choice of competition matrix. This suggests that many more matrices should be examined to get a better notion of the degree and range of the superiority of the CSEA.

The CSEA is one type of evolutionary algorithm, a fairly generic one that is easy to implement. There are a large number of other algorithms that are worth testing on the community assembly problem. It may also be worth publishing a collection of matrices that would serve as an additional test problem for evolutionary algorithms that use binary string representations. The work in this study shows that the community assembly problem is non-trivial with a fitness landscape possessing rich structure.

A. Stability and Diversity of Communities

The diversity data summarized in Table I shows that the CSEA locates a far more diverse collection of communities than the SSAA. The exemplary fitness tracks for the SSAA in Figure 3 and the tightness of the distribution of maximum fitness values for the SSAA in Figure 4 show that the SSAA is locating very similar communities over and over. The CSEA is not doing this. The search operators of the CSEA can change the community being examined a great deal more than the SSAA's addition of a single new species. This suggests that the communities located by the SSAA have some interesting properties:
1) They are very stable against invasion by a single species, confirming one of the results in [16].
2) Rephrasing the above, relative to the SSAA, these communities are local optima of the search space that are very hard to break out of.
3) As local optima these communities must have a very large basin of attraction, as the algorithm seems not to wander past them to the much better optima found by the CSEA. Remember that both algorithms are using the same figure of merit and are searching the same space.
4) The fact that the CSEA did not get stuck in these local optima (this is clear from the size of the optima located) suggests that they are not local optima of the evolutionary search.

The remarkable stability against invasion by a single additional species suggests three avenues for additional research. First, the SSAA could be modified to add pairs, small groups, or a reasonable distribution of numbers of additional species instead of just one. This may remove the “endpoint” state or it may lead to finding even more robust endpoint states. Second, it might be interesting to check the communities located by the CSEA for their stability against invasion by a single (or multiple) species. We hypothesize that the larger stable communities located with the CSEA will be less resistant to invasion than those located with the SSAA.

Finally, the size of the available species pool in this study was 100 for all three ecologies examined. In [16] the authors found that the stability of assembled communities increased with the size of the available species pool. Testing for this with the SSAA and CSEA, as well as their generalizations and modifications, is a natural topic for future investigation. In general, stability against invasion is a second potential criterion for optimization of communities.

B. Explaining Observed Diversity

Classical community interaction theory predicts that the number of critical resource limits in the environment should place an upper bound on the number of species in a stable community. This prediction is in complete disagreement with observations in the field and in [17] the authors show that including positive interactions between species in the model can mitigate the inaccuracy of classical theory. While far more than the three exemplary virtual ecosystems used in this study would be required, the CSEA could be used to perform a computational test of the positive interaction effect in a broad sampling of virtual ecosystems.

C. Open-ended Species Pools

It might also be interesting to study communities where the competition is not managed with replicator dynamics. Populations of game playing agents are an active topic of current research [6], [11], [8], [1] with reproduction managed
with an evolutionary algorithm. If the fitness of the agents is estimated by tournaments in which the agents compete by playing a game, then there is an implicit and intrinsically meaningful competition matrix. The distribution of the sizes of stable communities of game playing agents might be a good way to characterize the balance of a game. Unbalanced games would have a smaller number of stable communities and those communities would have fewer members.

Leaving game theory, there is also an interesting point in theoretical biology to be investigated. Several of the studies cited in the introduction note that the size of the available species pool has a strong influence on the outcome of the species assembly problem. In [4] the authors note that simulations in theoretical biology can benefit substantially from having a phenotype whose quality drives evolution. Many encodings for game playing agents such as finite state machines or the various representations of genetic programming [14], [15], [19] are potentially open-ended, providing an arbitrarily large species pool. Performing community assembly on game playing agents encoded in an open ended fashion would permit examination of the extreme case of the community assembly problem relative to the size of the species pool.

D. Other Competition Models

The logistic replicator dynamics used in this study were chosen because the reproduction of individual species is self-limiting, making the model rich in stable communities. There are a vast array of other replicator dynamics models in the literature. This study compared a standard and EA technique for locating large stable communities for one replicator dynamic model. The comparison, or just the apparently superior evolutionary technique, could be used with any replicator dynamic model.

Another very important factor required for accurate modeling of ecological communities is spatial structure. The work in [24], [23] demonstrates that the size of a stable ecological community is substantially enhanced by incorporating spatial structure into models. Grid based models like that in [2] could be used to replace the replicator dynamics component of the system used in this study as a way of incorporating spatial structure into the model.

E. Optimizing and generalizing the CSEA

A modest amount of ad-hoc parameter setting for the CSEA was performed while debugging the algorithm. Since substantially superior results were obtained with the evolutionary algorithm, additional parameter tuning was not required for this proof-of-concept study. For more challenging replicator dynamics models or to obtain more nearly optimal results, tuning the CSEA and testing other EA-techniques on the community assembly problem are an early priority for additional research.

REFERENCES