Large Network Analysis for Fisheries Management using Coevolutionary Genetic Algorithms

Garnett Wilson
Department of Computer
Science
Memorial University of
Newfoundland
St. John's, NL, Canada
gwilson@mun.ca

Simon Harding
Department of Computer
Science
Memorial University of
Newfoundland
St. John's, NL, Canada
simonh@mun.ca

Orland Hoeber
Department of Computer
Science
Memorial University of
Newfoundland
St. John's, NL, Canada
hoeber@mun.ca

Rodolphe Devillers
Department of Geography
Memorial University of
Newfoundland
St. John's, NL, Canada
rdeville@mun.ca

Wolfgang Banzhaf
Department of Computer
Science
Memorial University of
Newfoundland
St. John's, NL, Canada
banzhaf@mun.ca

ABSTRACT

Traditionally, a genetic algorithm is used to analyze networks by maximizing the modularity (Q) measure to create a favorable community. A coevolutionary algorithm is used here to not only find the appropriate community division for a network, but to find interesting networks containing substantial changes in data within a very large network space. The network is one of the largest, if not the largest, analyzed by evolutionary computation techniques to date and is created using a real world data set consisting of fisheries catch data in the north Atlantic Ocean off the coast of Canada. This work examines the quantitative performance of two types of coevolutionary algorithms against both a standard GA that uses a natural (but not necessarily optimal) division of the data set into communities, and simulated annealing. The goal for all search algorithms was to automatically find anomalies (differences in catch) within the data. To measure practical usefulness of the system, a fisheries expert analyzed the best networks located by the search algorithms using an existing visualization software prototype. The expert indicated that a refined version of coevolutionary GA known as PAMDGA was found to most reliably locate subnetworks containing catch differences of biological relevance.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Performance, Experimentation

Keywords

coevolution; genetic algorithm; Q modularity; spatiotemporal visualization; fisheries

1. INTRODUCTION

In most works that employ the use of genetic algorithms (GA) to analyze the structure of a network, researchers use community membership as a fitness function. In particular, the community membership function that is optimized by the GA is the modularity metric, also called the "Q metric." The Q metric rewards networks of densely connected nodes within communities and sparse connections between communities. The goal of evolution is to determine the community structure of a network with an optimal Q value. The fitness function typically only optimizes community structure; it does not look for anomalies within a network.

To the authors' knowledge, we present the first instance of a coevolutionary GA used for the analysis of community structure of a network in this work. Two coevolutionary GA-based algorithms are examined, where both algorithms determine not only community structure (as in other literature), but also simultaneously determine solutions. Using actual data obtained from the Department of Fisheries and Oceans in Atlantic Canada, a software tool called GTdiff [3] was developed that could analyze differences in bottom trawl survey data for the northern cod (*Gadus morhua*) in the form of a very large network [8]. A network is of interest due to one or more anomalies (or large changes in average catch data over time and geographical location). These large differences in catch between time and space are of interest to fisheries observers as they could indicate overfishing or biological phenomena that should be more closely examined, and as such they are considered good solutions to be located by the GA. In particular, the network nodes each correspond to an x, y point on an N x N grid for a particular span of

years. The edges in the GA solution network correspond to differences between the time spans, which are visualized in GTdiff as difference graphs that highlight changes between time spans (additional details and figure in Section 3). The GA algorithms described in this work represent an attempt to aid fisheries scientists using the visualization tool through partial automation of the search for anomalies in substantial amounts of catch data.

The size of this network is greater than any networks examined in the evolutionary computation literature that is known to the authors. The use of this large real world network poses a new challenge in that a known community structure does not exist in the network. A search thus can be conducted to establish a community structure. Given the context of a community structure for the network provided by one population, a second population of potentially anomalous networks is also evaluated using coevolution based on a community-based fitness metric.

Section 2 describes previous work in the evolutionary computation literature related to establishing community structure in networks. Section 3 describes the spatiotemporal fisheries catch data set used in this work, including its representation as a large network and the Q-based fitness function used by the GA for evaluation of its subnetworks. Section 4 describes the four search algorithms that are compared, namely genetic algorithm (GA), simulated annealing (SA), coevolutionary GA, and a refined coevolutionary GA called "Probabilistic Adaptive Mapping Development Genetic Algorithm (PAMDGA)". Section 5 describes quantitative performance analysis of the algorithms, with expert analysis of the networks located by the search algorithms following in Section 6. Section 7 provides conclusions.

2. PREVIOUS WORK

Genetic algorithms are often chosen as a search algorithm for optimizing the modularity metric Q of a network. The individuals evolved by the GA are typically a particular division of a network into communities, which can be considered a mapping of the network's nodes into communities. Tasgin et al. [7] applied a genetic algorithm to individuals that were a mapping of each node to a community, where the fitness function was the straightforward optimization of the Q metric. Tasgin et al. use the GA to decide the community structure of the network for two smaller (karate club and football matches) and one larger (email dataset) network that are often used as benchmarks in the literature. Their results indicate that the GA search is scalable for potentially large networks. Gog et al. [1] also use a GA algorithm to evolve individuals that are mappings of network nodes to communities. The GA is enhanced slightly in that the GA individuals are aware of the current optimum solution of the search and a best ancestor. Nicosia et al. [6] use a GA with overlapping community structure in the modularity-based fitness function (as in this work). Genotypes consisted of a mapping of nodes to communities for the network, with each element of the mapping representing the strength of community membership for each node. The authors examined evolved networks for two benchmarks and two newly introduced networks of varying size.

In the existing literature, if there is an ideal or known community structure for a network under investigation, authors typically attempt to show that an evolution-based algorithm is able to discover a network reasonably close to the known structure. If there is not a known community structure for the network (as in this work), authors will typically try to show that their solution can outperform other machine learning techniques in its ability to maximize the Q metric. In previous work [8], the authors presented a standard GA to search for anomalous networks given a natural, assumed community structure based on a year-based temporal ordering of nodes in the fisheries catch data network. While the assumption of temporally ordered years as a community structure is a natural one, there is the opportunity to expand the power of the search for anomalies by also allowing a search through the community space in addition to a search for a solution. In this work two types of coevolutionary GA are applied to the fisheries data set to provide improved networks that better represent anomalies that would be interesting to a domain expert. In order to search for both solution and appropriate underlying community structure simultaneously, we present two algorithms that coevolve two populations: one population of network solutions and another population of mappings that indicate node community membership.
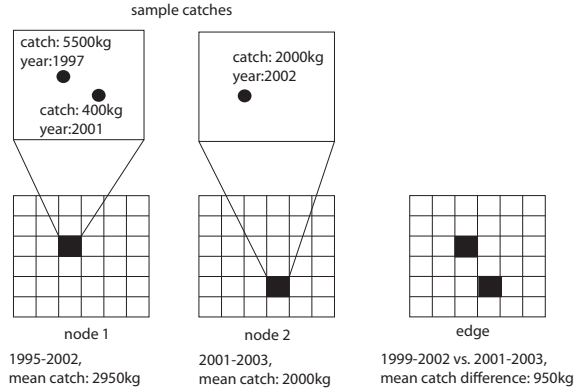
3. SPATIOTEMPORAL CATCH DATA

3.1 Dataset as a Very Large Network

The very large network used in this work is based on a spatiotemporal data set of annual bottom trawl survey catch data for the Atlantic cod (*Gadus morhua*) conducted by the Canadian Department of Fisheries and Oceans (DFO) for the Newfoundland and Labrador, Canada region. To the authors' knowledge, this data set as described herein represents the largest network data set analyzed in the evolutionary computation literature to date. Nodes represent catch amounts at a particular geographical location and an associated time span, where the mean catch over all data points in the given span of years is calculated to determine the level of catch for the node (see leftmost two girds of Figure 1). Edges in the network represent differences in the mean catch between locations over two time spans, and are shown in the rightmost grid of Figure 1. Edges in the network using two nodes with the same time span are excluded, since they do not reflect any difference because the entire geographical area is viewed at once in the visualization tool (application of the tool is discussed in Section 5). In virtue of not allowing the same time span (regardless of locations) in an edge, loops (reflexive ties) are prohibited in the large graph.

The data collected covers an area of 1,000,000 km^2 and a temporal range of 1980 to 2005. The data set produces a very large network to be evaluated: The search space involves a node for every pair of locations x, y in an N x N grid and two year time span. The number of unique, unordered two year time spans for the 26 year period we examine (1980 to 2005, inclusive) is $\binom{26}{2}$, or 325 possibilities. The span of one year (e.g. 1996 to 1996) is also considered a possible time span of interest, so the number of possible time spans is thus a total of $325 + 26 = 351$. The area covered by the data set is divided as a 30 x 30 grid (selected as an appropriate resolution for viewing changes in preliminary experiments with an expert). Given the number of possible time spans, there will thus be 30^2 x $351 = 315,900$ possible nodes to consider.

Nodes are average catch data over a particular area during a time span. We wish to consider the difference between nodes as absolute differences in those mean catches.

sample catches

catch: 5500kg
year:1997

catch: 400kg
year:2001

catch: 2000kg
year:2002

node 1
1995-2002,
mean catch: 2950kg

node 2
2001-2003,
mean catch: 2000kg

edge
1999-2002 vs. 2001-2003,
mean catch difference: 950kg

Figure 1: Relationship between network structure and spatiotemporal visualization. Catch amounts are hypothetical for purposes of illustration.

Thus, the network to be considered consists of an undirected, weighted graph. The number of all unique edges existing in this search space is the number of possible pairings of nodes, with no time span compared to itself, giving $n(n-1)/2t$ possibilities for n nodes and t time spans, or approximately 1.5 x 10^8 edges.

3.2 The Q Metric

The *modularity* (or Q) metric is applied in this work to a network where a strength is associated with the connection between nodes. Newman adapted the modularity metric for weighted networks in [5], which is used here and is defined as

$$Q_w = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta_f(c_i, c_j) \qquad (1)$$

where A_{ij} is the weight of the edge from i to j, k_i of a node i in a weighted network is the sum of the weights of the edges connected to it ($k_i = \sum_j A_{ij}$), and $m = \frac{1}{2} \sum_{ij} A_{ij}$ is the total weight of the edges in the network. Q_w has an absolute value between 0 and 1, where a value of over 0.3 is typically considered to indicate favorable community division [5].

The typical community membership function is denoted as $\delta_f(c_i, c_j)$, where c_i is the community to which a node i is assigned. In the traditional community membership function, a node cannot be a member of more than one community (communities cannot overlap). In this work, the members of a community are time spans of two years. The Q metric is thus modified to allow for the more natural choice of overlapping community membership, detailed in [2]. We calculate overlapping community membership as the degree of overlap between time spans (communities) within each of two nodes:

$$\delta_f(c_i, c_j) = \frac{|Y_i \cap Y_j|}{|Y_i \cup Y_j|} \qquad (2)$$

where Y_i is the enumeration of all years (inclusive) for the node i time span. Similarly, Y_j is the enumeration of years for the node j time span. The function δ_f yields a decimal value between 0 and 1, conforming to the traditional range of possible values for Q_w. In practice, the δ_f function provides large values of Q_w for networks with edges with large differences within overlapping time frames. For instance, the user may see that two years out of a five year span reflect abnormally large (or low) catches on average.
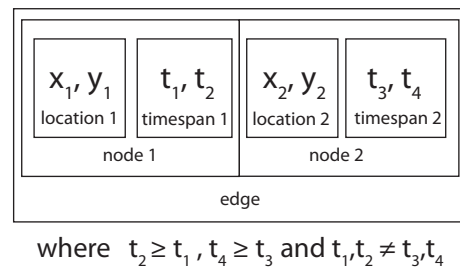
In addition to traditional community overlap, we examine a variant of (1) where δ_f is replaced by $1 - \delta_f$ to emphasize non-overlapping time spans. In this instance, large differences in catch for non-overlapping time spans will be highlighted by the GA search. As an example, a three year span containing high (low) catch compared to a six year span starting after the end of the three year span could be identified.

In the case of either variant of (1), there can be a trade-off between maximizing overlap (or reducing overlap) of particular time spans and the largeness of the catch differences within those time spans. Also, one large difference between time spans (edge weight) can sometimes come at the expense of other lower differences (edge weights) in the same network. These trade-offs are effected by the differences detected by search in the network, the number of time spans (communities) in the network, and the associated lengths of those time spans. All of these aspects will be examined in greater detail as they relate to individual algorithms in Section 5. In particular, we compare the results of a GA, two coevolutionary GAs, and simulated annealing. Each of these algorithms is considered using each of the two Q_w variants.

4. SEARCH ALGORITHMS

4.1 GA Algorithm

Each GA individual genotype is a chromosome of 20 gene sequences, where each sequence is an ordered set of 8 integers (genes) that correspond to an edge in the network. Each individual is thus a list of edges, or a network that should be of interest to the user. The first 4 integers correspond to the first node in the edge, while the last 4 integers correspond to the second node of the edge. In each set of 4 integers identifying a node, the first two integers provide the x and y coordinates in the N x N grid and the second two integers provide the two years of a time span in the data set and are always ordered. There is an added restriction on the 8 integers of the edge that the time spans (integers 3,4 and 7,8) cannot be the same. The absolute difference between the average catch over all years for the time span and location pair in each node is considered to be the weight of the edge. The gene sequence representing an edge is shown in Figure 2.

x_1, y_1	t_1, t_2	x_2, y_2	t_3, t_4
location 1	timespan 1	location 2	timespan 2
node 1		node 2	
edge			

where $t_2 \geq t_1$, $t_4 \geq t_3$ and $t_1, t_2 \neq t_3, t_4$

Figure 2: Gene structure representing two nodes and their joining edge.

The GA involves a steady state tournament of 100,000 rounds using a population of 10. The small population size has been found to improve evolutionary search and will guide the construction of interesting networks rather than relying on the discovery of randomly generated material in a larger initial population that must be found using extensive search. The algorithm for the GA tournament is

```
for each tournament round
  randomly select 4 genotypes
  determine fitness (Q value) of 4 genotypes
  keep best two genotypes as parents
  replace worst two genotypes with parents
  mutation on children
  crossover on children
```

Four individuals are selected for evaluation for each tournament round. The top two individuals become "parents" and are left unchanged (in virtue of this aspect of selection, the best individuals are left unchanged). The last two individuals are replaced by copies of the parents and thus become "children." The children then have mutation and crossover applied to them. The mutation operator is applied to the children always, and the operation involves each edge-based sequence of 8 genes having a 50% (rate of 0.5) chance of being mutated. In practice, on average half of the nodes and edges in the childrens' network are replaced with newly generated edges to more efficiently explore the search space. Standard two-point crossover is used to exchange sections of edge-based gene sequences, also at rate of 0.5. The fitness function is (1), either with a preference for overlap (δ_f) or no overlap ($1 - \delta_f$). The mutation and crossover rates were found to provide fast real-time search progress in our experiments when the best individuals in each round were retained.

4.2   Coevolutionary GA Algorithm

While the standard GA algorithm provides a means of searching for an anomalous subnetwork involving large catch differences from a larger network, there is no search occurring to attempt to refine a community structure for the large network. Most GAs only perform the search for an optimal community structure for a single network. We now present the first usage of a coevolutionary GA to search for both an optimized solution (subnetwork) and an appropriate community structure for the general problem space.

The coevolutionary GA maintains and performs a search on two populations: a genotype and a mapping population. The genotype structure is as described for the general GA, but there is also the evolution of the underlying mapping structure as a separate population. That is, the second population consists of potential mappings of all time spans of two years to a community. As described in Section 3, there are 351 time spans that are each considered a community in the standard GA. A mapping individual consists of all ordered year pairings that constitute a time span. Upon initialization, each ordered pairing of years are given a randomly assigned community number from 1...351. As such, the mapping is redundant because more than one ordered pair of years (time span) can be a member of the same community and there will be 351 or less communities. By allowing the mapping of time span to community to be redundant, the GA search on the mapping population will emphasize particular sets of time spans (not necessarily se-

quential). That is, instead of each time span being its own community by definition (as in the standard GA), a number of collectively interesting time spans can be grouped in a community by the coevolutionary search in the mapping population. The coevolutionary GA operates as follows

```
for each tournament round
  if round % 2 == 0
    randomly select 4 genotypes
    determine genotype, best mapping fitnesses
    keep best two genotypes as parents
    replace worst two genotypes with parent copy
    mutation on children genotypes
    crossover on children genotypes
  if round % 2 != 0
    randomly select 4 mappings
    determine mapping, best genotype fitnesses
    keep best two mappings as parents
    replace worst two mappings with parent copy
    mutation on children mappings
```

The coevolutionary GA operates in somewhat the same manner as the standard GA, except there are alternating rounds for evaluation of genotype individuals and mapping individuals. In the genotype evaluation rounds (even numbered rounds), each of the four genotype individuals chosen are paired with the current best mapping individual (chosen randomly in the initial round) for fitness evaluation. No genotype individual can be evaluated in the absence of a mapping individual in the coevolutionary algorithm: there is no default association of community with each time span. The two losing genotype individuals are subjected to mutation and crossover based on the established thresholds. In the mapping evaluation rounds, 4 mapping individuals are chosen and are evaluated by pairing with the current best genotype individual. As before, the best two mapping individuals are retained as parents and the worst two mappings become copies of the parents and are potentially subjected to the operation of mutation. The rate of mutation is kept low (rate of 0.1) so that there is a more consistent context against which the genotypes will evolve (found to be beneficial in initial experiments). Since the mappings are an ordered list of year time spans, crossover is not appropriate.

4.3   PAMDGA Algorithm

The PAMDGA (Probabilistic Adaptive Mapping Developmental Genetic Algorithm) algorithm is a genetic algorithm version of the PAMDGP (Probabilistic Adaptive Mapping Developmental Genetic Programming) algorithm introduced in [9]. The algorithm is designed to overcome known difficulties of more standard coevolutionary algorithms, in particular loss of context and fitness spiking due to a sudden change in one of the current best individuals grouped for coevolution (known as the "Red Queen Effect"), lack of exploration of the search space, and general lack of fitness-based performance. These drawbacks are mitigated or eliminated through the use of slight elitism (not allowing the best genotype or mapping in the best pairing to be altered by mutation or crossover), and exploration of the search space using a selection table that enables any grouping to be selected at any time (see [9] for additional background).

The algorithm begins with initialization of genotype and mapping populations of size g and m, respectively. A probability table of size g x m is then created with cells initial-

ized to $1/m$. For each round of a steady state tournament, 4 cells of the probability table are selected using roulette selection on the m axis. The selected cells correspond to four genotype/mapping pairings that are selected where the genotypes must be unique but the mappings can be chosen more than once. The pairings are evaluated for fitness, and the best two pairings are considered parents and are left unaltered. The best two pairings are also checked against the current best genotype/mapping pairing found so far in the tournament to determine if they will be identified as the new best. Once the current best genotype/mapping pairing is identified, the table cell corresponding to the two best genotype/mapping pairings is updated according to

$$P(g,m)_{new} = P(g,m)_{old} + \alpha(1 - P(g,m)_{old}) \qquad (3)$$

and the other combinations in the same column are updated according to

$$P(g,m)_{new} = P(g,m)_{old} + \alpha(P(g,m)_{old}) \qquad (4)$$

where g is the index of the genotype, m is the index of the mapping, α is the learning rate (corresponding to the emphasis of current table values over previous values), and $P(g,m)$ is the probability in cell $[g, m]$ of the table. Updates by equations 3 and 4 result in all values in a column always having a sum of unity. A threshold value of γ is used to prevent premature convergence on a sub-optimal solution: Following the table update, if any cell in the probability table column corresponding to the winning genotypes exceed γ, all values in that column are then reset to $1/m$ so they sum to unity. The effect of noise addition and normalization is to effectively reset the chances of selection of all mappings with respect to the genotype handled by that table column. The last two ranked pairings are considered the children and are subject to genetic operations based on their respective associated thresholds. However, if either the genotype or mapping of the losing pairings is identified as the current best genotype or mapping found so far in the tournament, they are protected from both mutation and crossover.

```
initialize genoPopSize x mapPopSize probTable
for each tournament round
  use roulette selection on probTable mapping rows
    choose 4 geno/mapping pairs (unique geno)
  rank 4 geno/mapping pairs
  verify or replace bestGeno/bestMapping pair
  update probability table by Eq. (1) & (2)
  if (cell in bestGeno column >= gamma)
    set column values to 1 / mapPopSize
  leave best 2 geno/mapping pairs as parents
  for worst 2 geno/mapping pairs
   if (geno != bestGeno)
      replace genotype with parent
      mutation of genotypes
      crossover of genotypes
   if (mapping != bestMapping)
      replace mapping with parent
      mutation of mapping
```

For the purposes of this work, the number of tournaments was kept the same as GA and coevolutionary GA (100,000 rounds). The rate of mutation and crossover for genotypes, and mutation for mappings is the same as the coevolutionary GA described in the previous section. The α learning

rate is set at 0.1 (found to be a good value in preliminary experiments), and γ is set at 0.9 to indicate that for a particular genotype column in the probability table no particular mapping should have a chance to be selected that exceeds 90%.

4.4 Simulated Annealing

Simulated Annealing (SA) is often considered to be the prominent search algorithm for determining large Q divisions of a network [4]. As such, we use it as a benchmark algorithm against which to compare the performance of the standard GA and the coevolutionary GA. Each GA algorithm is run for 100,000 rounds with 4 individuals evaluated per round, so 400,000 individual evaluations are conducted. For equivalent computational effort, the SA is run for 400,000 evaluations.

The SA algorithm tracks both the best state located so far in the search and the current state in the search. For each cycle of the SA, the current state can be replaced with the new candidate state with probability $e^{\Delta E/T}$ where ΔE is the change in value of Q and T is the current temperature of the system. The temperature of the system is reduced with each completed cycle. The SA then updates the best state located so far with the the new current state if it has a higher Q (energy) than the current best state.

5. PERFORMANCE RESULTS

All algorithms were examined with respect to four metrics: fitness achieved (highest Q value of network located), number of timespans in the best network, maximum difference in the best community generated, and mean length of time spans. The results are shown in the boxplots of Figures 3 to 7 below for 50 trials (of 400,000 evaluations each). The bottom, middle, and top lines of boxes indicate lower quartile, median, and upper quartile values, respectively. If there is no overlap in the notches of two boxes around the medians, the medians of the two data sets are different at the 0.95 confidence interval. Whiskers extend from the boxes to 1.5 times the interquartile range. The symbol '+' corresponds to points from 1.5 to 3 times the interquartile range, and the symbol 'o' corresponds to points outside 3 times the interquartile range.

In Figure 3 it is evident that for the overlap-favored fitness function, the coevolutionary algorithm provided the highest Q values. A more precise plot of its Q values is given in Figure 4. The tendency for the coevolutionary GA to switch mappings (to jump from one overlapping time span pair to another) likely yielded these results. However, our domain expert (fourth author) found the non-overlapping fitness function to provide the most worthwhile information (discussed in greater detail in the next section). For the non-overlap favored fitness type, PAMDGA and GA provided significantly higher Q values. However, we can see from a higher precision view of the best performing algorithms in 4 that there is no difference between PAMDGA and GA at 95% confidence with respect to the median.

As mentioned in Section 3.2, the Q-value of the network is a result of the length of time spans (related to community membership and overlap), number of time spans considered (related to number of communities), and the differences in catch between times (value of network edges). Figures 5 to 7 consider these aspects of the Q modularity in turn. In Figure 5 it is evident that for the overlap-favoring fit-
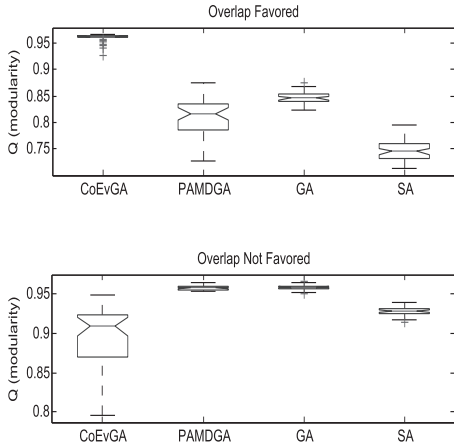
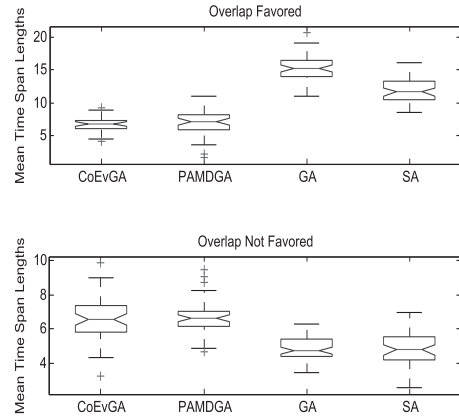**Figure 3: Modularity (Q) of the best networks for given fitness function over 50 trials.**
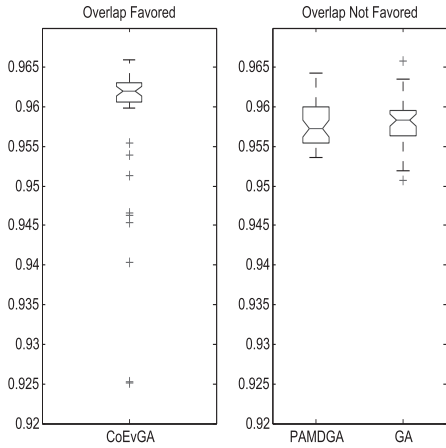
Figure 4: Modularity (Q) of the best networks for given fitness function over 50 trials at greater precision for best algorithms.
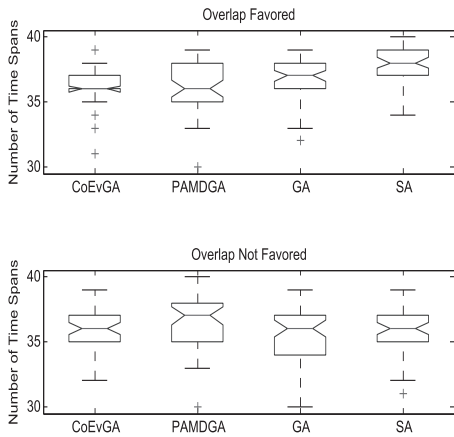

Figure 5: Number of time spans within the best networks for given fitness function over 50 trials.

ness function, the standard coevolution and PAMDGA algorithms produce the lowest number of time spans but are not different with any statistical significance. Lower number of time spans given equal network sizes will yield higher modularity, so lower number of time spans is better. For the non-overlap favored fitness, there is very little difference in the number of time spans in the best networks produced by every algorithm (notches overlap).

Closely related to the number of time spans represented by the networks is the length of the time spans used, as seen in Figure 6. For the overlap favored fitness function, the coevolutionary GA and PAMDGA had the smallest time span lengths being compared when overlapped and GA had the largest time spans. Conversely, given the non-overlap favored fitness function, the coevolutionary GA and PAMDGA produced the largest time span lengths. This is significant because the coevolutionary algorithms aim, in their evolution of the mapping population, to emphasize particular time spans. In particular, using a fitness function rewarding overlapping time spans, shorter time spans were em-

Figure 6: Average length of time spans within nodes of the best networks for given fitness function over 50 trials.
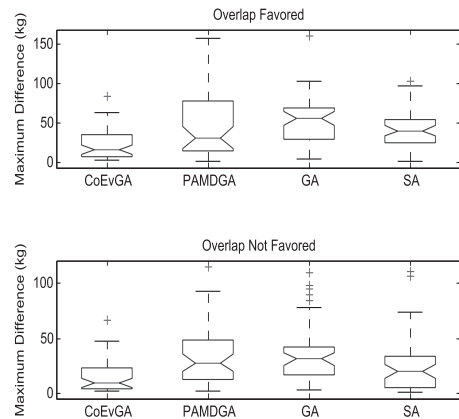
Figure 7: Maximum difference within the best networks for given fitness function over 50 trials.

phasized in coevolution. For a fitness function emphasizing non-overlapping time spans, coevolution emphasized longer time spans.

In Figure 7, examining overall spread of the maximum catch difference in each network, we can see that for the overlap-favored fitness function the coevolutionary GA actually identified the lowest average differences throughout the networks. Coupled with the fact that it produced the highest modularity networks (Figure 3), we can conclude that coevolutionary GA traded off large differences for higher overlap in years. Paired with shorter time spans considered (recall Figure 6), this trade off would naturally not produce the most interesting networks for fisheries experts. For the non-overlap favored network, in terms of overall spread of the data, PAMDGA consistently produced large maximum average differences within its network when noting reach of whisker and outliers (but its results in this respect are not statistically different than GA around the median and are generally comparable to GA). However, for non-overlapping time spans, PAMDGA dominates the other algorithms when considering both longer time span lengths (Figure 6) and large average differences (Figure 7) together. By maximizing both the time span length and catch difference metrics, PAMDGA produces high modularity networks (Figure 3) that were chosen as most useful by a fisheries expert for the geographical area under study.

6. EXPERT EXAMINATION OF VISUALIZATIONS

A fisheries expert, the fourth author, examined the best networks located by all algorithms (GA, SA, Coevolutionary GA, and PAMDGA). He was asked to rate each network edge of the best networks. Network edges were visualized using GTdiff as two temporal bins and one corresponding difference graph. The first two grids are the temporal bins in GTdiff, and display average catch in kilograms in each spatial grid element. The two temporal bins are ordered sequentially based on the last year of the time spans; if the last year is the same, the temporal bins are ordered by the first year. The colour scale spans from light yellow (lowest average catch) to brown (largest average catch). The third grid is the difference graph, where the difference in average catch between the two time spans is displayed as a positive (green) or negative (red) change. No change in catch is represented by white, and the degree of saturation of green and red is used to represent positive and negative differences, respectively. The visualization tool was used to change the resolution of the final networks to 10 x 10. The resolution does not change the content of the networks and allowed for viewing of trends by the expert and presentation in publication rather than examination on the high resolution display used in the study.

During the time period examined (1980 to 2005), there were known anomalies (large differences in catch over time): Fisheries scientists reported that the population levels of cod dropped suddenly in the early 1990s, which prompted a moratorium on cod fisheries starting in 1992. In addition, other smaller changes known to fisheries scientists occurred during particular years. The three options for the rating of each difference graph by the expert were: No (meaning no difference relevant to fisheries scientists appeared), Relevant (a difference relevant to fisheries scientists appeared),

or Salient (a special case of Relevant indicating that an important biological shift was identified). The ratings of each of the difference graphs (individual edges) in the final best network of each algorithm out of 50 trials are summarized in Table 1 as the number of responses for each rating and total differences rated.

Our expert found that for almost all results for every algorithm using the overlap favored fitness function, differences were not emphasized appropriately. In particular, there were many instances of small changes in catch between two time spans with a very large degree of overlap in the final networks. Despite this, the expert indicated that PAMDGA provided the largest number of relevant differences, with both coevolutionary algorithms significantly outperforming the standard GA and SA algorithms.

The expert found that for non-overlap favored fitness, SA had the largest proportion of relevant differences. However, there were only 10 differences in the best SA network. The expert also indicated that none of the differences located in the SA were of particular interest; they simply presented known overall trends with no particular anomalous changes identified. In contrast, PAMDGA presented the highest proportion of salient anomalies in the data. PAMDGA gleaned from the dataset a remarkably higher percentage of salient differences than any other algorithm (3/16, or 18.75%). We present all three difference graphs for PAMDGA that represented salient differences for the expert in Figure 8.

The top difference graph in Figure 8 clearly shows the decline of cod stocks from the decade leading up to the moratorium of 1992 to 1993 on northern cod compared to a sample later year in the future. In the middle graph of Figure 8, the darker red across the region indicates that cod stocks dropped considerably after the years 1983 to 1991, which were prior to the moratorium on cod (1992 to 1993). This difference graph nicely contrasts those previous years of abundance with post moratorium levels of 1994 to 1997. In the bottom difference graph of Figure 8, years mostly covering the time period of the moratorium (1991 to 1994) are compared to years following the moratorium by several years (2000 to 2004). The interesting aspect of these data for the expert was the red in the difference graph depicting the decline in cod stocks post-moratorium in the northeast of the grid with the exception of a place southwest of Newfoundland (light green portion of grid). This area actually corresponds to known areas where levels of cod during the moratorium were lower than the current levels.

Table 1: Ranking of Difference Graphs

Overlap Favored				
	No	Relevant	Salient	Differences
GA	5	2	0	7
SA	6	1	0	7
CoEvGA	6	11	0	17
PAMDGA	2	5	0	7
Overlap Not Favored				
	No	Relevant	Salient	Differences
GA	10	6	1	17
SA	3	7	0	10
CoEvGA	6	6	1	13
PAMDGA	6	7	3	16

7. CONCLUSIONS

This work described the application of coevolutionary GA algorithms to the simultaneous discovery of problem solution (interesting network containing anomalous large changes in catch data) and underlying community structure. A large real-world network based on geospatial fisheries catch data over a 25 year span in Atlantic Canada was used as the basis for performance comparisons. The fitness function for the analysis of the networks used modularity (the Q metric) with a community membership function that either did or did not favor overlapping of communities to emphasize different relationships between time periods. Four search algorithms were compared: GA, SA, standard coevolutionary GA, and PAM DGA (refined coevolutionary GA). The best networks found by all algorithms were examined using a prototype visualization tool designed for fisheries scientists.

Results indicated that the coevolutionary algorithm produced superior Q-based fitness results for overlap-favored fitness. However, the fisheries expert indicated that non-overlap favoring fitness provided more interesting final networks. With respect to the non-overlap favored fitness function, PAMDGA provided high fitness networks that combined both large differences and extended time spans in its chosen networks to a greater degree than the other algorithms. Upon examination of the final networks by the expert, he found that PAMDGA located the largest number of interesting known trends in the catch data.
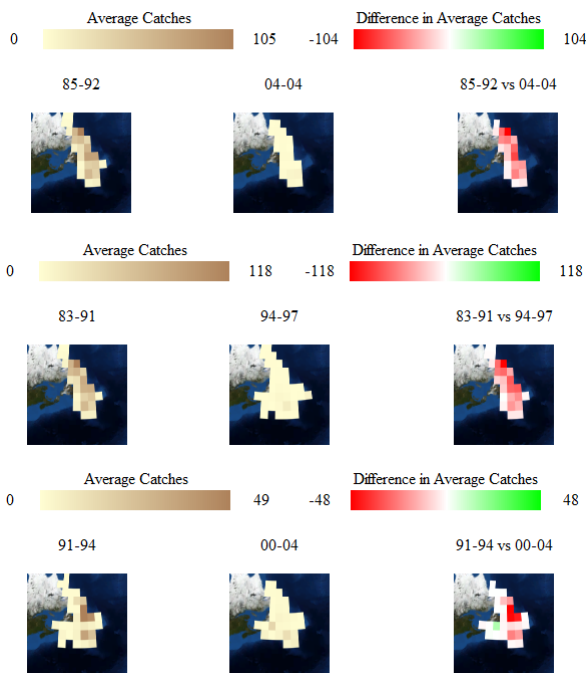
Figure 8: Salient difference graphs selected by expert from the highest Q, no overlap favored network produced by PAMDGA. Catches are shown in thousands of kg.

9. REFERENCES

[1] A. Gog, D. Dumitrescu, and B. Hirsbrunner. Community detection in complex networks using collaborative evolutionary algorithms. In *ECAL'07: Proceedings of the 9th European Conference on Advances in Artificial Life*, pages 886–894, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] S. Gregory. Fuzzy overlapping communities in networks. Technical Report CSTR-10-008, University of Bristol, October 2010.

[3] O. Hoeber, G. Wilson, S. Harding, R. Enguehard, and R. Devillers. Exploring geo-temporal differences using gtdiff. In *2011 IEEE Pacific Visualization Symposium (PacificVis)*, pages 139 –146, New York, USA, 2011. IEEE Press.

[4] X. Liu, D. Li, S. Wang, and Z. Tao. Effective algorithm for detecting community structure in complex networks based on ga and clustering. In *ICCS '07: Proceedings of the 7th International Conference on Computational Science, Part II*, pages 657–664, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70(5):056131, Nov 2004.

[6] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03024, 2009.

[7] M. Tasgin and H. Bingol. Community detection in complex networks using genetic algorithm. In *ECCS '06: Proc. of the European Conference on Complex Systems*, Apr. 2006.

[8] G. Wilson, S. Harding, O. Hoeber, R. Devillers, and W. Banzhaf. Detecting anomalies in spatiotemporal data using genetic algorithms with fuzzy community membership. In *ISDA 2010: 10th International Conference on Intelligent Systems Design and Applications*, pages 97–102, Chennai, India, 2010. Research Publishing Services.

[9] G. Wilson and M. Heywood. Introducing probabilistic adaptive mapping developmental genetic programming with redundant mappings. *Genetic Programming and Evolvable Machines*, 8:187–220, June 2007.