Parallel Exhaustive Search vs. Evolutionary Computation in a Large Real World Network Search Space

Garnett Wilson, Simon Harding, Orland Hoeber,
Rodolphe Devillers, and Wolfgang Banzhaf

Memorial University of Newfoundland, Canada (G.W., O.H., R.D., W.B.)

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Switzerland (S.H.)

Issues

- Machine Learning (local optima)
- Exhaustive Search (global optima)
- Execution Performance

Data Set

- We wish to locate anomalies involving
  - catch weight (kg)
  - location
  - time
- annual bottom trawl scientific survey
- Canadian Department of Fisheries and Oceans (DFO)
- Newfoundland and Labrador region
- covers 1,000,000 km^2
- Atlantic cod (*Gadus morhua*) is the focus
- temporal range of 1980-2005
  - includes collapse, moratorium

Data as Large Network: Nodes

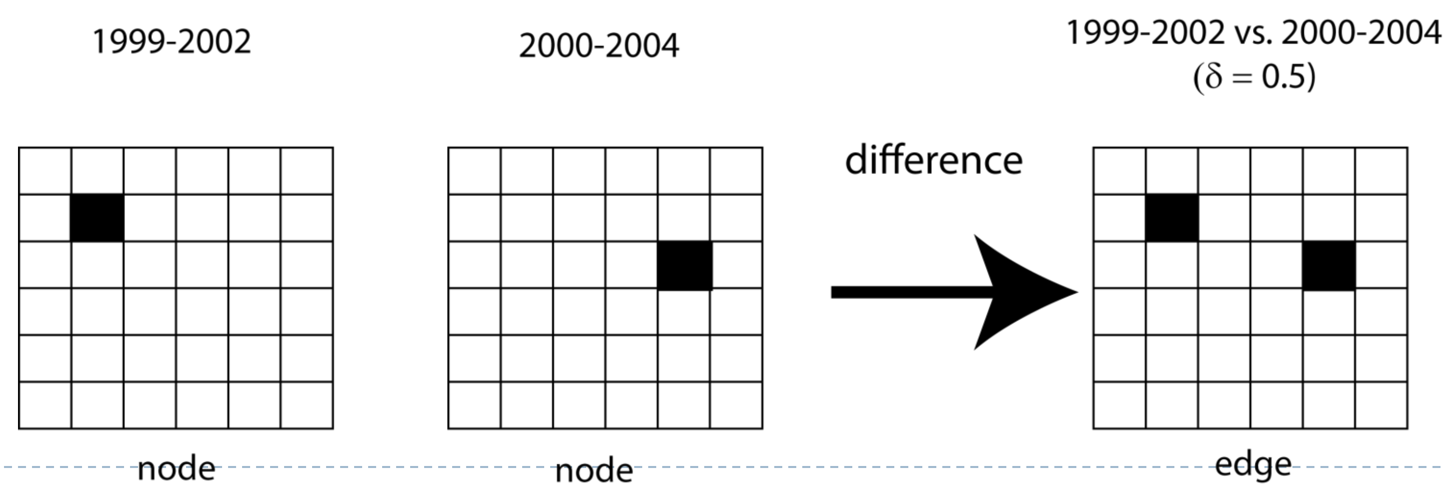
- A node for every combination of
  - location *x,y* in an N x N grid
  - two year time span.
- Time spans:
  - 25 years (1980 to 2005) gives
    26 choose 2 = 325 possibilities.
  - span of one year (e.g. 1996-1996) is also a time span
  - possible time spans is 325 + 26 = 351 in total.
- 30 x 30 grid, so there are
  30^2 x 351 = 315, 900 nodes

Data as Large Network: Edges

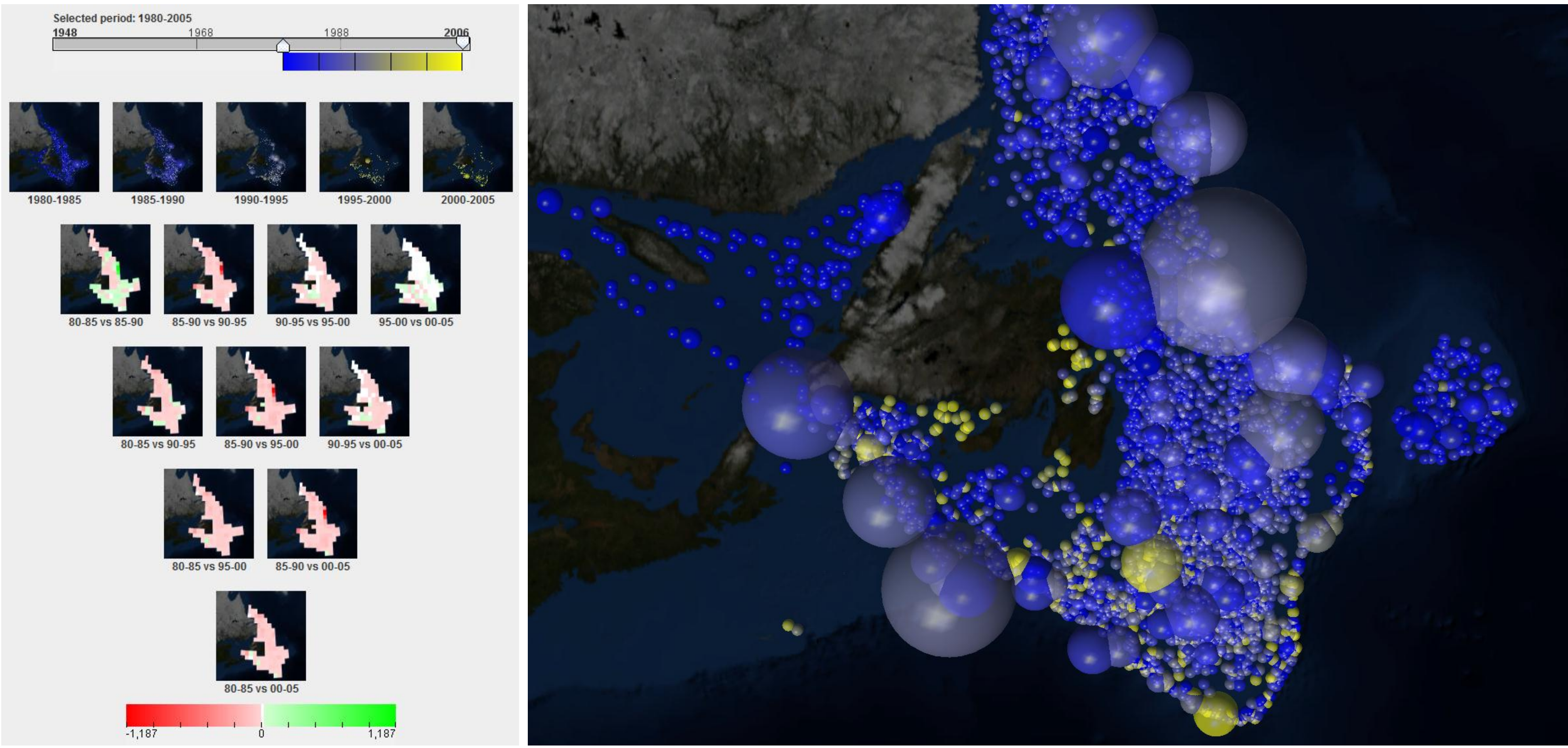
- Edges represent
  - *absolute* difference in catch data
  - between two areas
  - over two time spans.
- Undirected, weighted graph.
- Two time spans can overlap in each edge.
- Both nodes cannot have same time span in one edge (no loops/reflexive ties)
- unique edges ↔ pairings of nodes
- $n\ (n - 1)\ /\ t$ possibilities
  for n nodes and t time spans
  giving 2.8×10^8 edges

Spatiotemporal Visualization of Network Structures

- x,y point in N x N grid for time span
  - node ↔ temporal bin
- difference between time spans
  - edge ↔ difference graphs



Temporal View

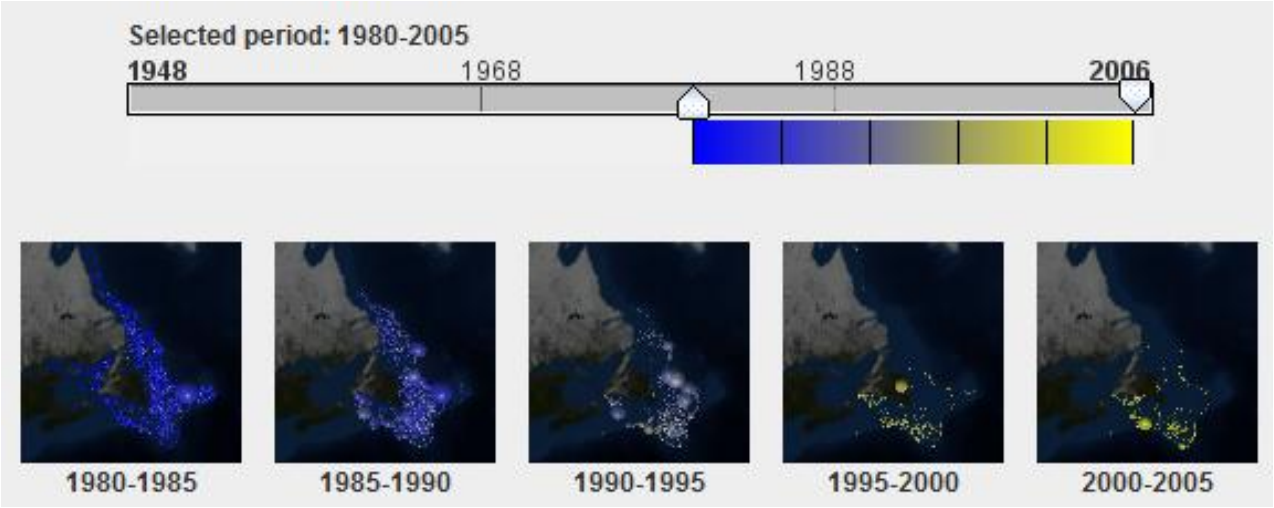


Difference View

Geospatial View

Temporal Binning

- Filering of data temporally
- Equal length temporal bins
  - Specified by user
  - Color encoded
- Data from each bin shown in mini-geospatial views
- Colour scale under timeline as legend



GTDiff

- Visual representation of differences in temporal bins
- Divergent color scale
- Catch has increased (green)
- Catch has decreased (red)

GA Individual and Gene Structure

- composed of 20 gene sequences
- each gene sequence is ordered set of 8 integers
  - corresponds to edge in network
  - first and last 4 integers represent nodes
    - first 2 integers = location
    - last 2 integers = time span
- edge weight = absolute difference in average catch over time span at location in each node

$$\overbrace{\overbrace{\underbrace{x_1, y_1}_{\text{location1}}\ ,\ \underbrace{t_1, t_2}_{\text{timespan1}}}^{\text{node1}}\ \overbrace{\underbrace{x_2, y_2}_{\text{location2}}\ ,\ \underbrace{t_3, t_4}_{\text{timespan2}}}^{\text{node2}}}^{\text{edge}}$$

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

GA Fuzzy Community Algorithm: Fitness Function

- Modularity (*Q*) metric

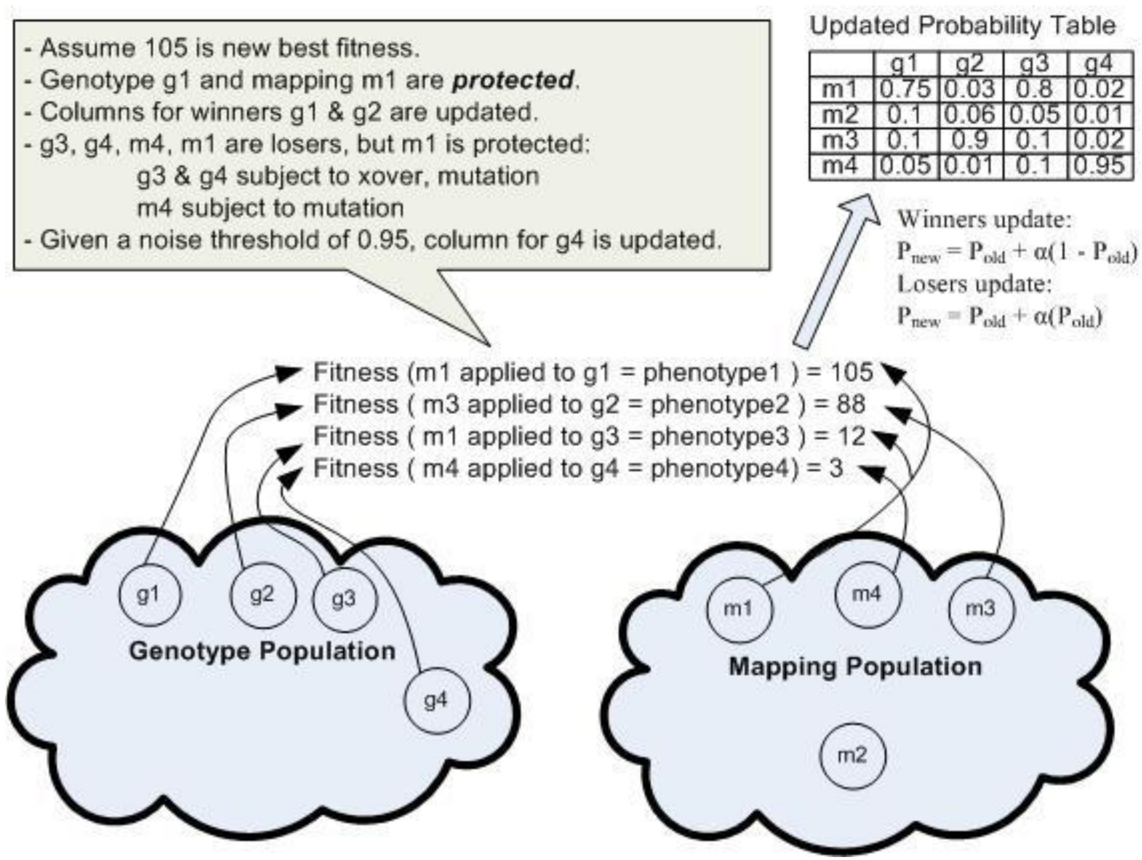
$$Q_w = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta_f(c_i, c_j)$$

  - where A_{ij} is the weight of the connection from i to j
  - k_i of a node i is the sum of the weights of attached edges
  - m is the number of edges in the network
  - δ is the community membership function

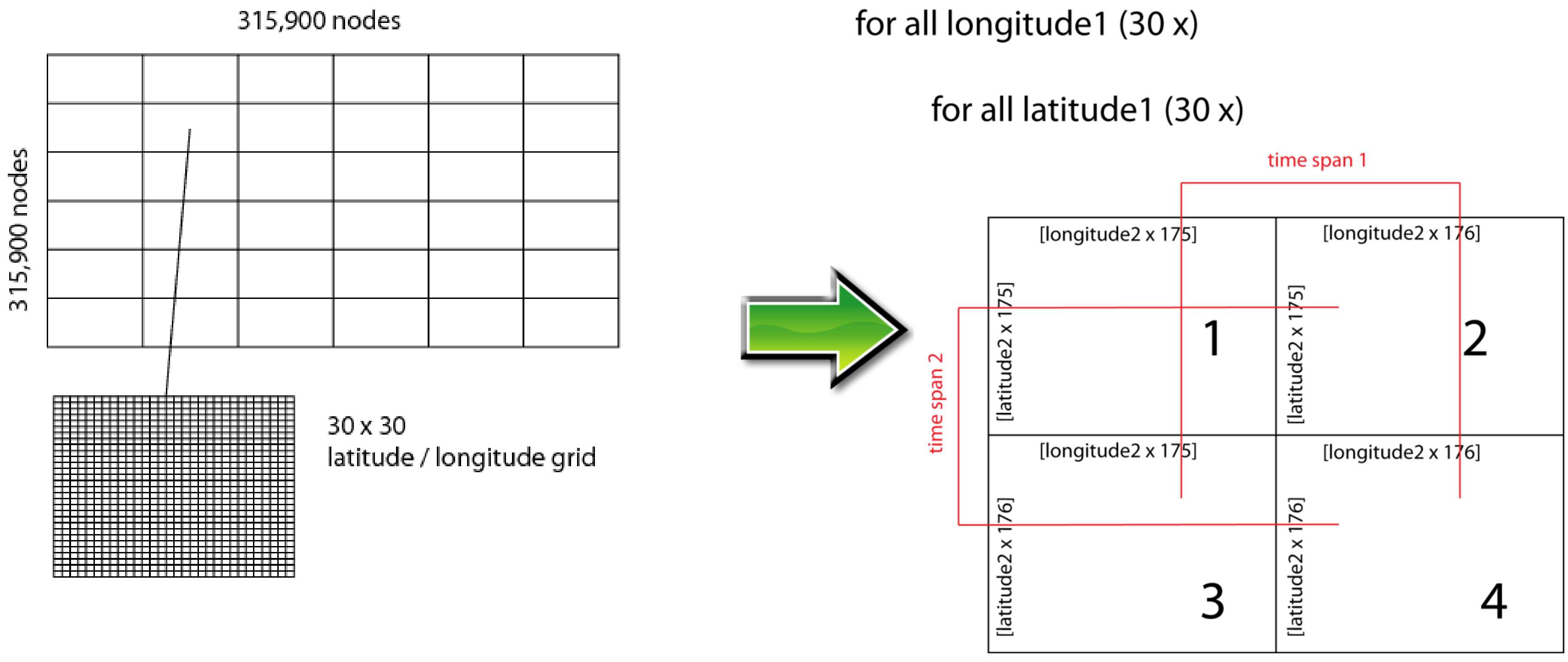
Mapping Individual Structure

Time Span	Mapping
1980, 1981	290
1980, 1982	350
…	…
1996, 1999	290
…	…
2004, 2005	12

Probabilistic Adaptive Mapping Developmental Genetic Programming (PAM DGA)



Parallel Exhaustive Search: Search Space Conception



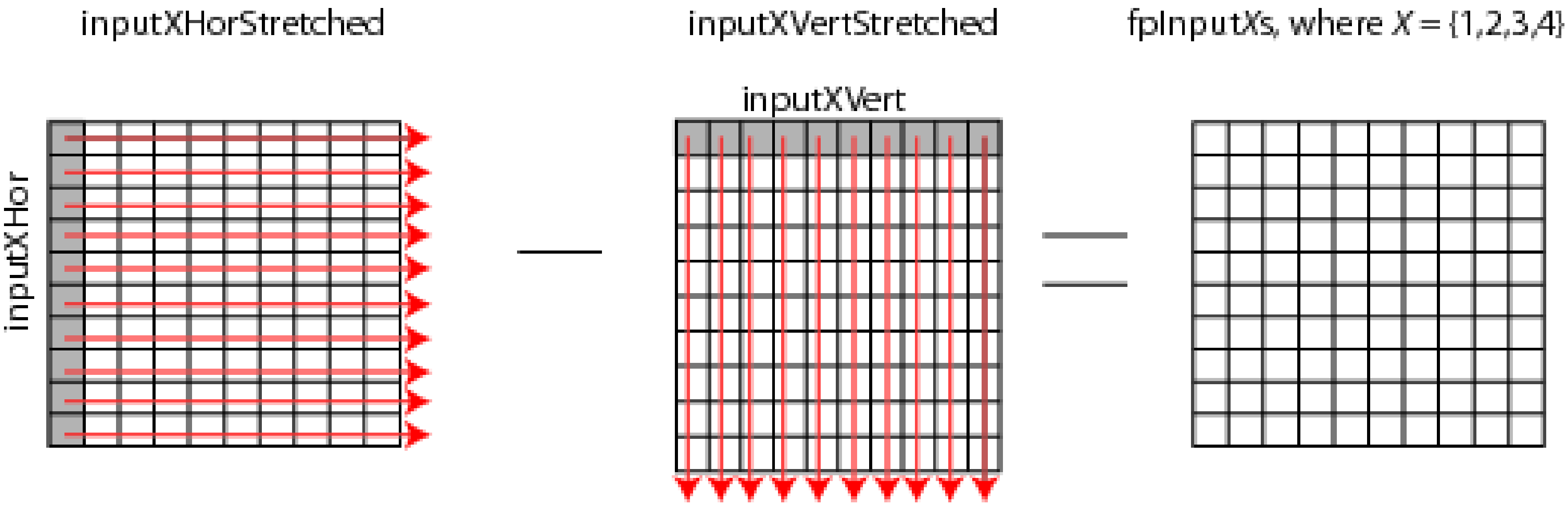
Parallel Exhaustive Search: CPU-side Code

```
int maxLatitude = 30; int maxLongitude = 30;
// input node data in 4 vectors: weightDiffs1_1, 1_2, 2_1, 2_2
for (int longit1 = 0; longit1 < maxLongitude; longit1++)
  for (int latit1 = 0; latit1 < maxLatitude; latit1++)
    if using GPU
      fill weightDiffs1_1 & weightDiffs1_2 vectors for longit2, time span 1
      fill weightDiffs2_1 & weightDiffs2_2 vectors for latit2, time span 2
      on GPU: determine weight difference for all grid points (see Figure 7)
    if using CPU
      fill input1 and input2 arrays for longit2, time span 1
      fill input3 and input4 arrays for latit2, time span 2
      for each input array start a thread (4 threads total)
        in each thread determine vertical, horizontal maxes
find max across arrays 1,2 and 3,4 (rows) and 1,3 and 2,4 (columns)
for maximum values for all rows
  find corresponding longit2, time span 1
for maximum values for all columns
  find corresponding latit2, time span 2
```

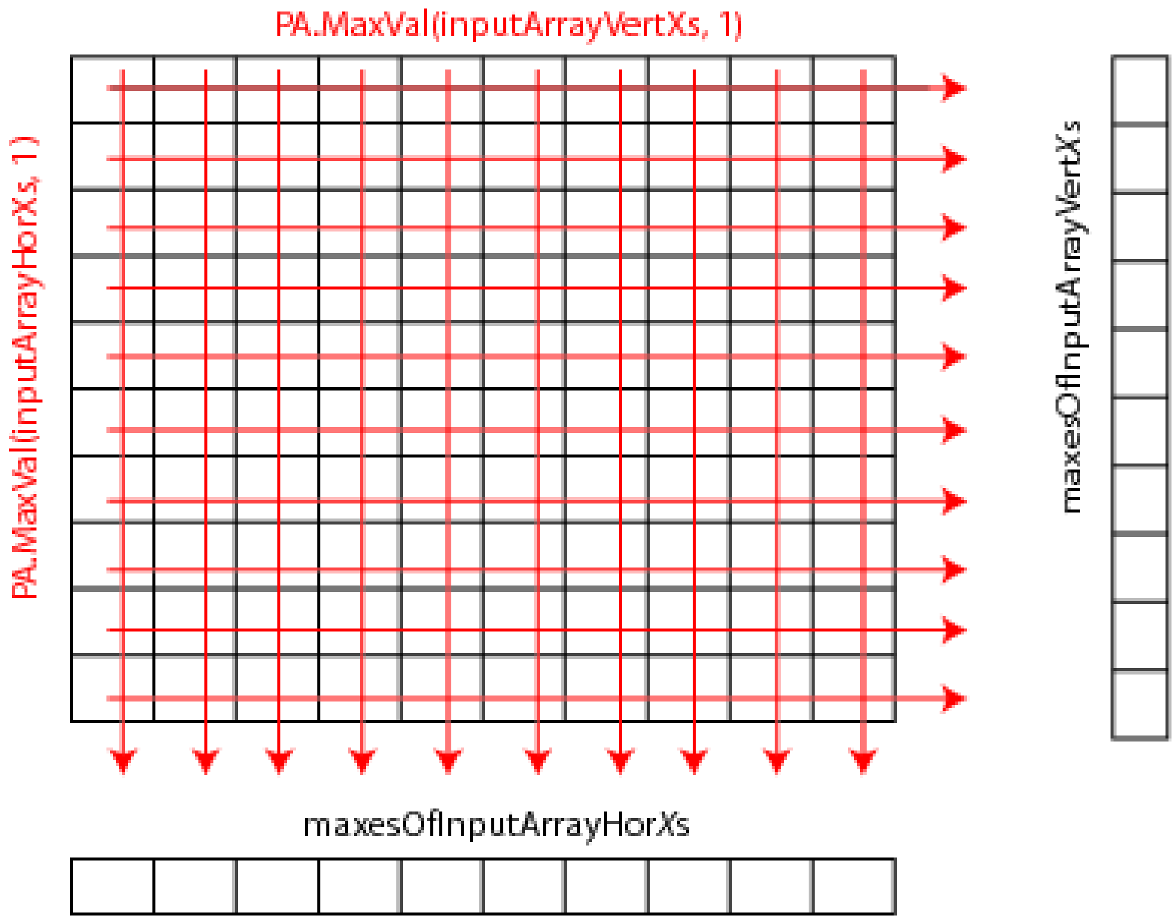
Parallel Exhaustive Search: GPU-side Code

```
1  // replicate weight vectors across rows, columns
2  FPA input1Vert = new FPA(weightDiffs1_1[longit1, latit1]);
3  FPA input2Vert = new FPA(weightDiffs1_2[longit1, latit1]);
4  FPA input3Vert = new FPA(weightDiffs1_1[longit1, latit1]);
5  FPA input4Vert = new FPA(weightDiffs1_2[longit1, latit1]);
6  FPA input1VertStretched = PA.Replicate(input1Vert, dim1);
7  repeat 6 for 3 other grids
8  FPA input1Hor = new FPA(weightDiffs2_1[longit1, latit1]);
9  FPA input2Hor = new FPA(weightDiffs2_1[longit1, latit1]);
10 FPA input3Hor = new FPA(weightDiffs2_2[longit1, latit1]);
11 FPA input4Hor = new FPA(weightDiffs2_2[longit1, latit1]);
12 FPA input1HorStretched = PA.Replicate(PA.Transpose(input1Hor), dim1);
13 repeat 12 for 3 other grids
15 // determine absolute difference between weight differences
16 FPA fpInput1 = PA.Abs(PA.Subtract(input1VertStretched, input1HorStretched));
17 repeat 16 for 3 other grids
19 // determine max in each row and column for [longit1, latit1].
20 FPA fpOutputVert1 = PA.MaxVal(fpInput1, 1);
21 repeat 20 for 3 other grids
22 FPA horFpInput1 = PA.Transpose(fpInput1);
23 repeat 22 for 3 other grids
24 FPA fpOutputHor1 = PA.MaxVal(horFpInput1, 1);
25 repeat 24 for 3 other grids
26 // move result from GPU back to CPU for further processing
27 maxesOfInputArrayVert1s[longit1, latit1] = evalTarget.ToArray1D(fpOutputVert1);
28 repeat 27 for 3 other grids
29 maxesOfInputArrayHor1s[longit1, latit1] = evalTarget.ToArray1D(fpOutputHor1);
30 repeat 29 for 3 other grids
```

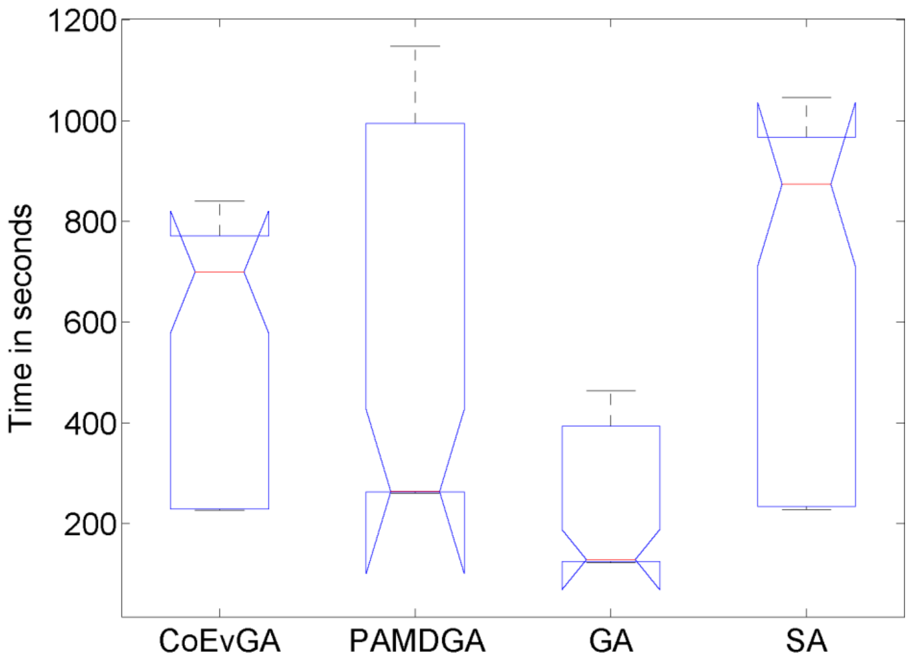
Parallel Exhaustive Search on GPU 1: Replication and Subtraction

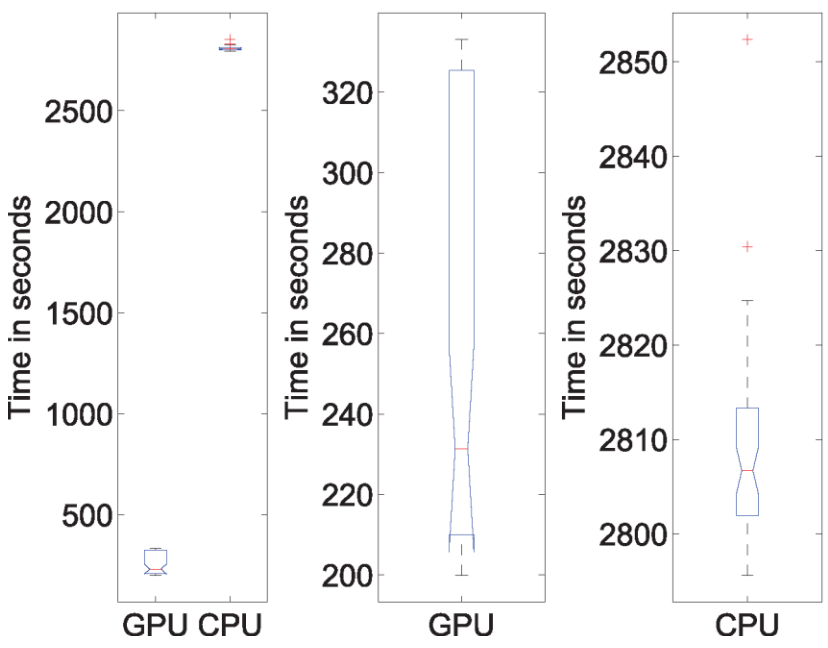


Parallel Exhaustive Search on GPU 2: Maximums across all rows and columns



Performance Results





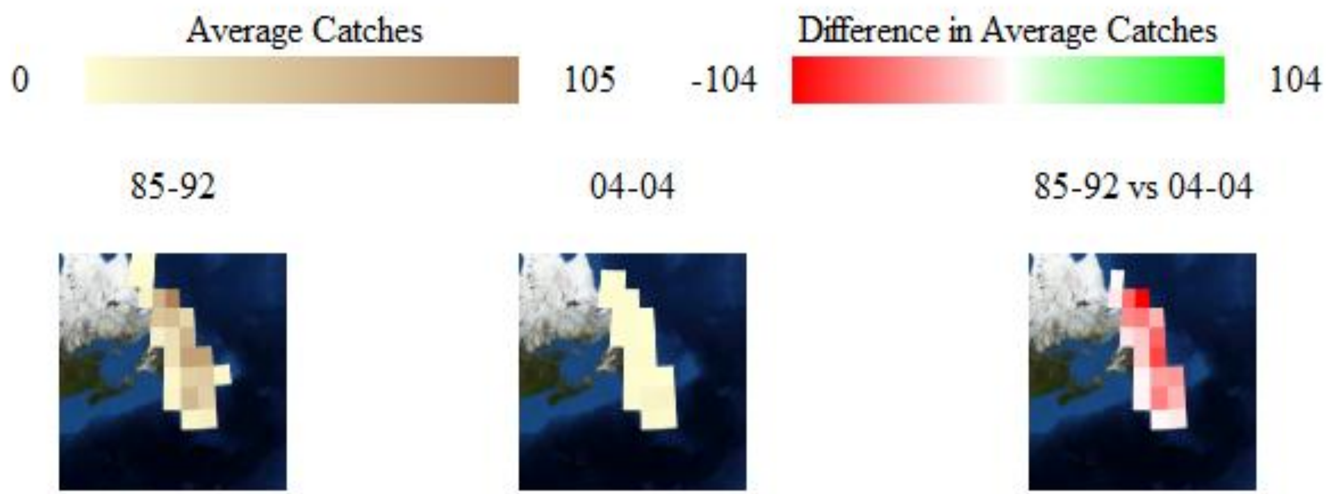
Expert Results: Summary

TABLE I

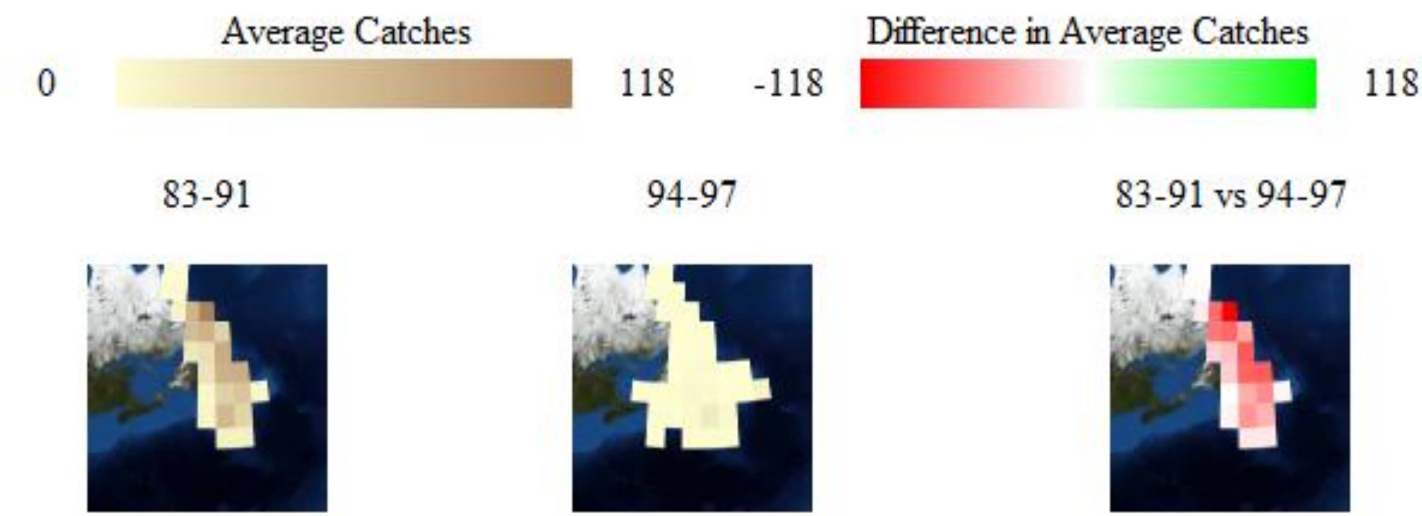
RANKING OF DIFFERENCE GRAPHS

	No	Relevant	Salient	Differences
GA	10	6	1	17
SA	3	7	0	10
CoEvGA	6	6	1	13
PAMDGA	6	7	3	16
Parallel	13	6	1	20

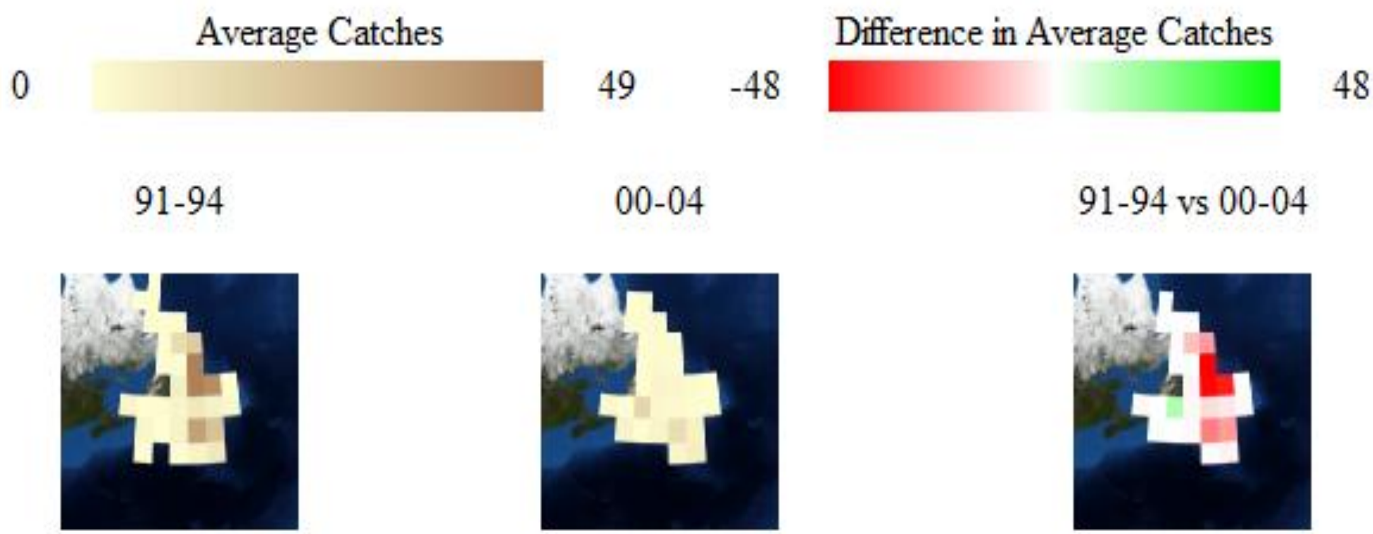
Expert Results:
PAM DGA No.1: GA, Overlap Not Favored



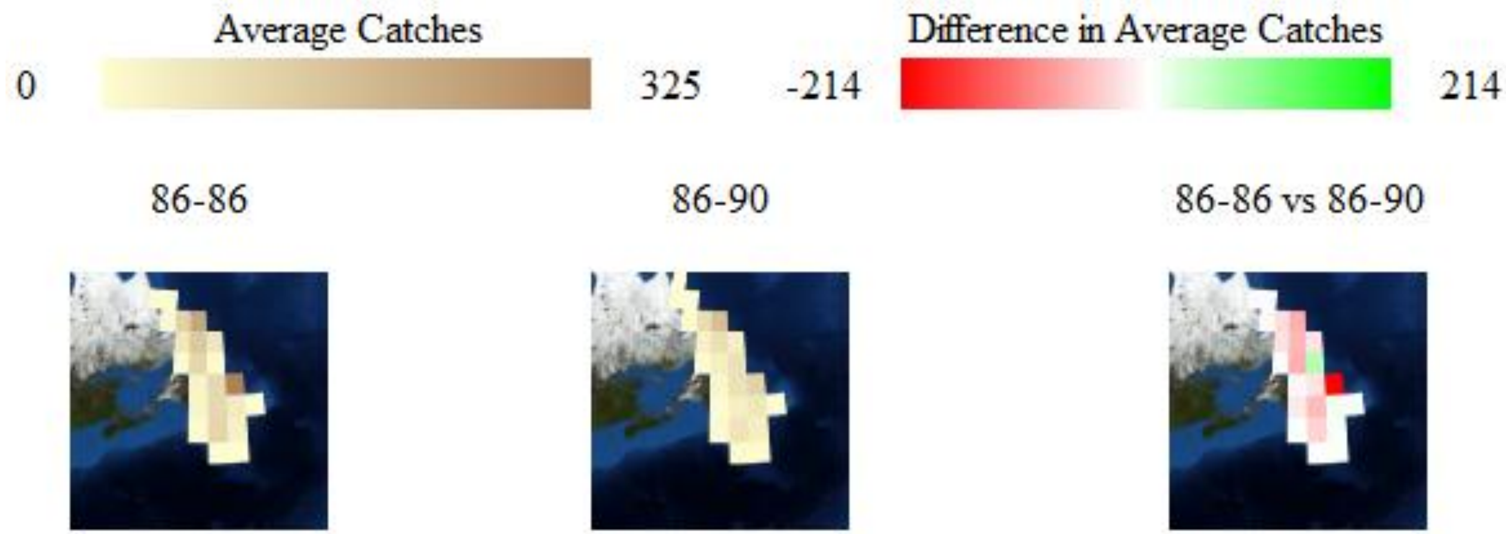
Expert Results:
PAM DGA No.2: GA, Overlap Not Favored



Expert Results:
PAM DGA No.3: GA, Overlap Not Favored



Expert Results: Exhaustive Search



Results Summary

- GPU provides speedup of ~12x that of the CPU
- impressive speedup given GPU literature
  - comparison to multicore CPU implementation
  - well beyond the 2.5x stated by Lee et al. [8]
- fisheries expert found greater value in local optima (EC)
- global optima tended to focus on time periods of
  - abundant catches
  - less interest than EC results