4M17 Exercise III Optimizing the bird Function

eng-216er

January 12, 2015

Summary: Summary goes here.

Contents

List of Figures

List of Tables

1 Rationale behind the use of JavaScript

JavaScript is unique in that programs written in it can be embedded in a html document, and executed in a web browser. No other language can be used for client side web programming without either using a browser extension (Java, Flash) or compiling into JavaScript (CoffeeScript).

This provided the motivation for me to implement the optimisation algorithms in JavaScript. In small part, this was because of the possibility of creating a simple html based UI for controlling the optimization parameters and inspecting the results.

Largely however, I was drawn to using JavaScript because being able to solve optimization problems in a browser could potentially be useful within several web programming contexts. For instance, the development of WebGL allows for hardware accelerated graphics problems to be developed for the web. Optimization can be used to solve useful problems in graphics programming. An example is computing the best possible conformal mapping between texture co-ordinates, and coordinates that make up a mesh of a surface. This cam be used to apply a texture to a 3D surface, while minimising the effect of distortion on the surface.

There are currently very few JavaScript optimization libraries. Although the software provided in this report does very little to rectify that, it does provide a starting point for more complex software.

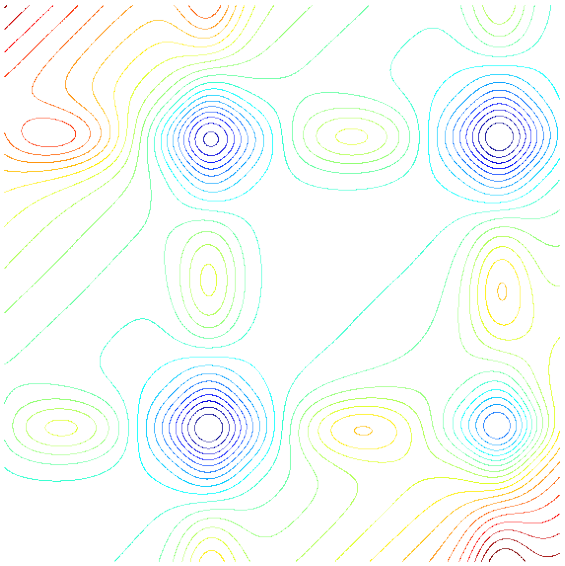
A Listings

A.1 PlotImage.m

```
function [  ] = PlotImage(  )
n = 1000;
range = linspace(-6,6,n);
y = Bird( ones( n,1 ) * range , range' * ones( 1,n )  );

contour( range, range, y, 23);
axis equal
axis off
end
```

A.2 Bird.png



A.3 index.html

```
<!DOCTYPE html>

<html>
<head>
  <title>Optimization Ex3</title>
  <meta charset="utf-8">
  <script type="text/javascript" src="minmax.js">
</script>
  <script type="text/javascript" src="ex3.js">
</script>
  <script type="text/javascript" src="genetic_algorithm.js">
</script>
  <script type="text/javascript" src="tabu.js">
</script>
  <style type="text/css">
```

```
body{
    font-family: monospace;
  }
  h1 {
    text-align: center;
    font-size: 16pt;
  }
  .parent {
    max-width: 500px;
    max-height: 500px;
    margin: 1em auto;
    border: 1px solid black;
  }
  .buttons {
    margin: 1em auto;
    text-align: center;
  }
  .animParent{
    margin: 0.5em auto;
    padding: 0;
    height: 4em;
    overflow: hidden;
    text-align: center;
  }
  #loadingAnim{
    margin: 0 auto;
    display: none;
    background-color: black;
    border: 1px solid black;
    -webkit-animation: loader 2s ease infinite;
    animation: loader 2s ease infinite;
  }
  @-webkit-keyframes loader {
    from {
      opacity: 1.0;
      margin-top: 1em;
      width: 0em;
      height: 0em;
      border-bottom-right-radius: 0em;
      border-bottom-left-radius: 0em;
      border-top-right-radius: 0em;
      border-top-left-radius: 0em;
    }
    to {
      opacity: 0;
      margin-top: 0em;
      width: 4em;
      height: 4em;
      border-bottom-right-radius: 2em;
      border-bottom-left-radius: 2em;
      border-top-right-radius: 2em;
      border-top-left-radius: 2em;
    }
  }
  @keyframes loader {
    from {
      opacity: 1.0;
      margin-top: 2em;
      width: 0em;
      height: 0em;
      border-bottom-right-radius: 0em;
      border-bottom-left-radius: 0em;
```

```
78          border-top-right-radius: 0em;
79          border-top-left-radius: 0em;
80        }
81        to {
82          opacity: 0;
83          margin-top: 0em;
84          width: 4em;
85          height: 4em;
86          border-bottom-right-radius: 2em;
87          border-bottom-left-radius: 2em;
88          border-top-right-radius: 2em;
89          border-top-left-radius: 2em;
90        }
91      }
92      #controls {
93        text-align: left;
94        margin: 0 auto;
95        width: 30em;
96      }
97    </style>
98  </head>
99
100 <body>
101   <h1>Bird Function Optimizer</h1>
102
103   <div class="parent">
104     <canvas id="c" width="500" height="500"></canvas>
105   </div>
106
107   <div class="buttons">
108     <button class="algorithm" onclick=
109     "genetic_algorithm(␣bird_function␣)">Genetic Algorithm</button>
110     <button class="algorithm" onclick=
111     "tabu_search(␣bird_function␣)">Tabu Search</button>
112
113     <div class="animParent">
114       <div id="loadingAnim">
115          
116       </div>
117     </div>
118
119     <div id="urlsection"></div>
120
121   </div>
122
123   <div id="controls">
124     <h3>Settings:</h3>
125
126     <span>Pause Between Iterations</span><br>
127     <input type="range" class="range" min="0" max="1000" step="1"
128     value="500" id="pause" onchange="updateSlider('pause')">
129     <span id="pausespan"></span> ms<br>
130
131     <h3>GA Specific Settings:</h3>
132
133     <span>Population Size</span><br>
134     <input type="range" class="range" min="10" max="50" step="5"
135     value="25" id="gapopulation" onchange=
136     "updateSlider('gapopulation')"> <span id=
137     "gapopulationspan"></span><br>
138
139     <span>Parent Count</span><br>
```

```
    <input type="range" class="range" min="5" max="25" step="1"
    value="10" id="gaparents" onchange="updateSlider('gaparents')">
    <span id="gaparentsspan"></span><br>

    <span>Selection Strategy</span><br>
    <select id="gastratergy">
      <option value="tournament">
        Tournament Select
      </option>
      <option value="frequency">
        Frequency Dependant Select
      </option>
    </select><br>

    <span>Mutation Rate</span><br>
    <input type="range" class="range" min="0" max="0.5" step="0.02"
    value="0.02" id="gamutate" onchange="updateSlider('gamutate')">
    <span id="gamutatespan"></span><br>

    <h3>Tabu Specific Settings</h3>

    <span>Short Term Memory</span><br>
    <input type="range" class="range" min="1" max="20" step="1"
    value="7" id="tabushort" onchange="updateSlider('tabushort')">
    <span id="tabushortspan"></span><br>

    <span>Medium Term Memory</span><br>
    <input type="range" class="range" min="1" max="20" step="1"
    value="4" id="tabumedium" onchange=
    "updateSlider('tabumedium')"> <span id=
    "tabumediumspan"></span><br>

    <span>Long Term Memory Grid Length</span><br>
    <input type="range" class="range" min="1" max="20" step="1"
    value="3" id="tabulong" onchange="updateSlider('tabulong')">
    <span id="tabulongspan"></span> per side<br>

    <span>Intensification Step</span><br>
    <input type="range" class="range" min="1" max="40" step="1"
    value="10" id="tabuintensify" onchange=
    "updateSlider('tabuintensify')"> <span id=
    "tabuintensifyspan"></span><br>

    <span>Diversification Step</span><br>
    <input type="range" class="range" min="1" max="40" step="1"
    value="15" id="tabudiversify" onchange=
    "updateSlider('tabudiversify')"> <span id=
    "tabudiversifyspan"></span><br>

    <span>Step Size Reduction Step</span><br>
    <input type="range" class="range" min="1" max="40" step="1"
    value="25" id="tabustepreduce" onchange=
    "updateSlider('tabustepreduce')"> <span id=
    "tabustepreducespan"></span><br>

  </div><img id="birdFunctionContour" src="Bird.png" alt=
  "hidden image" style="display: none">
</body>
</html>
```

A.4 minmax.js

```
Array.prototype.min = function(comparer) {

    if (this.length === 0) return null;
    if (this.length === 1) return this[0];

    comparer = (comparer || Math.min);

    var v = this[0];
    for (var i = 1; i < this.length; i++) {
        v = comparer(this[i], v);
    }

    return v;
}

Array.prototype.max = function(comparer) {

    if (this.length === 0) return null;
    if (this.length === 1) return this[0];

    comparer = (comparer || Math.max);

    var v = this[0];
    for (var i = 1; i < this.length; i++) {
        v = comparer(this[i], v);
    }

    return v;
}
```

A.5 ex3.js

```
"use strict";

function draw_background( canvas, ctx ){
  var backgroundElem = document.getElementById( "birdFunctionContour" );
  ctx.clearRect ( 0 , 0 , canvas.width, canvas.height );
  ctx.drawImage( backgroundElem, 0, 0, canvas.width, canvas.height );
}

function clear_screen(){
  var canvas = getCanvasAndContext()[0];
  var context = getCanvasAndContext()[1];
  draw_background( canvas, context );
}

var evalCount = 0;
var resetEvalCount = function(){
  evalCount = 0;
};
var getEvalCount = function(){
  return evalCount;
}

function bird_function( x1, x2 ){
  evalCount = evalCount + 1;
  var y = Math.sin(x1) * Math.exp( Math.pow(1 - Math.cos(x2),2 ) ) +
      Math.cos(x2) * Math.exp ( Math.pow(1 - Math.sin(x1), 2) )+
```

```
      Math.pow(x1 - x2, 2);
  return y;
}

var getCanvasAndContext;

var drawPoint = function( x1, x2, colour ){

  var canvas = getCanvasAndContext()[0];
  var context = getCanvasAndContext()[1];

  var centerX = canvas.width * ( x1 + 6 )/12;
  var centerY = canvas.height * ( -x2 + 6 ) /12;
  var radius = 2;
  //window.console.log( "X1: " + x1 + " X2: " + x2 );
  context.beginPath();
  context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
  context.fillStyle =  colour || 'green';
  context.fill();
  context.lineWidth = 0.5;
  context.strokeStyle = 'black';
  //context.stroke();
}
var connectPoints = function( a1, a2, b1, b2 ){

  var canvas = getCanvasAndContext()[0];
  var context = getCanvasAndContext()[1];

  var aX = canvas.width * ( a1 + 6 )/12;
  var aY = canvas.height * ( -a2 + 6 ) /12;
  var bX = canvas.width * ( b1 + 6 )/12;
  var bY = canvas.height * ( -b2 + 6 ) /12;

  context.beginPath();
  context.moveTo(aX, aY);
  context.lineTo(bX, bY);
  context.stroke();
}

function clearChildren( node ){
  while (node.firstChild) {
    node.removeChild(node.firstChild);
  }
}

function displayCsvStringAsURL( string ){
  var d = document.getElementById("urlsection");
  var a = document.createElement("a");
  a.href = "data:text/csv," + encodeURIComponent( string );
  a.textContent = "data";
  clearChildren( d );
  d.appendChild(a);
}

function logMinimumHistory( minimumHistory ){
  var csvString = "Evaluations,␣x1,␣x2,␣y\n"
  minimumHistory.forEach( function( h ){
    csvString += h.evaluations + ",␣" + h.x1 + ",␣" + h.x2 + ",␣" + h.y + "\n";
  } );
  displayCsvStringAsURL( csvString );
}

```

```
function getItterationPause(){
  return document.getElementById( "pause" ).value;
}


function setRunning(){
  var d = document.getElementById("urlsection");
  clearChildren(d);
  d.textContent = "Running";
  document.getElementById("loadingAnim").style.display = "block";

  var buttons = document.getElementsByClassName( "algorithm" );
  [].forEach.call( buttons, function(b){
    b.disabled = true;
  } );
}

function finishRunning(){
  document.getElementById("loadingAnim").style.display = "";
  var buttons = document.getElementsByClassName( "algorithm" );
  [].forEach.call( buttons, function(b){
    b.disabled = false;
  } );
}

function updateSlider(id){
  var slider = document.getElementById( id );
  var label = document.getElementById( id + "span" );
  label.textContent = slider.value;
}

window.onload = function(){
  var canvas = document.getElementById( "c" );
  var ctx=canvas.getContext("2d");
  draw_background( canvas, ctx );

  var ranges = document.getElementsByClassName( "range" );
  [].forEach.call( ranges, function(r){
    updateSlider( r.id );
  } );

  getCanvasAndContext = function(){
    return [canvas, ctx];
  }
  finishRunning();
}
```

A.6 tabu.js

```
function TabuPoint( f, x1, x2 ){
  this.x1 = x1 || 0;
  this.x2 = x2 || 0;

  this.getValue = function(){
    return [this.x1, this.x2];
  }

  var lastCalled;
  this.getFValue = function(){
    if( lastCalled === undefined || !(this.isEqual( lastCalled )) ){
      this.fValue = f( this.x1, this.x2 );
```

```
      lastCalled = this.clone();
    }
    return( this.fValue );
  }
  this.isEqual = function( p ){
    if( p.x1 === this.x1 && p.x2 === this.x2 ){
      return true;
    }
    return false;
  }
  this.clone = function(){
    var o = new TabuPoint( f, this.x1, this.x2 );
    o.lastCalled = this.lastCalled;
    o.fValue = this.fValue;
    return o;
  }
  this.valid = function(){
    if( Math.abs( this.x1 ) <= 6.0 && Math.abs( this.x2 ) < 6.0 ){
      return true;
    }
    return false;
  }
}

function tabuMin( a, b ){
  return a.getFValue() < b.getFValue() ? a : b;
}

function tabuMax( a, b ){
  return a.getFValue() < b.getFValue() ? a : b;
}

function considerForMediumTermMemory( memory, point, size ){
  if( memory.length < size ){
    memory.push( point );
  } else {
    var max = memory.max( tabuMax )
    if( max.getFValue() > point.getFValue() ){
      // add the point
      var rIndex = memory.indexOf( max );
      memory.splice( rIndex, 1, point );
    }
  }
}


function addToMemory( memory, value, memSize ){
  memory.push( value );
  if( memory.length > memSize ){
    memory.splice( 0, memory.length - memSize );
  }
}

function getAveragePoint( memory, f ){
  var x1 = 0, x2 = 0;
  memory.forEach( function(m){
    x1 += m.x1/memory.length;
    x2 += m.x2/memory.length;
  } );
  return new TabuPoint( f, x1, x2 );
}

```

```
75  function setupLongTermMemory(size){
76    var m = Array(size);
77    for( var i = 0; i< size; i++ ){
78      m[i] = Array( size );
79      for( var j = 0; j< size; j++ ){
80        m[i][j] = 0;
81      }
82    }
83    return m;
84  }
85
86  function addToLongTermMemory(memory, point){
87    var i =  Math.floor( memory.length*( point.x1 + 6 )/12.01 );
88    var j =  Math.floor( memory.length*( point.x2 + 6 )/12.01 );
89    memory[i][j] +=1;
90  }
91
92  function getDiversePointFromLongTermMemory(memory, f){
93    for( var i = 0; i< memory.length; i++ ){
94      for( var j = 0; j< memory[i].length; j++ ){
95        if( memory[i][j] === 0 ){
96          memory[i][j] = 1;
97          window.console.log( "i:␣" + i + ",␣j:␣" + j + ",␣m:" + memory[i][j]);
98          return new TabuPoint( f,
99            12 * ((i+0.5)/memory.length) - 6,
100           12 * ((j+0.5)/memory.length) - 6
101         );
102       }
103     }
104   }
105   return new TabuPoint( f, 0, 0 );
106 }
107
108 function updateTabuMinimumHistory( minimumHist, minimum ){
109   var o = {
110     evaluations: getEvalCount(),
111     x1: minimum.x1,
112     x2: minimum.x2,
113     y: minimum.getFValue()
114   };
115   minimumHist.push( o );
116 }
117
118 function getShortSize(){
119   return document.getElementById( "tabushort" ).value;
120 }
121 function getMediumSize(){
122   return document.getElementById( "tabumedium" ).value;
123 }
124 function getLongSize(){
125   return document.getElementById( "tabulong" ).value;
126 }
127 function getIntensifyStep(){
128   return document.getElementById( "tabuintensify" ).value;
129 }
130 function getDiversifyStep(){
131   return document.getElementById( "tabudiversify" ).value;
132 }
133 function getReduceStep(){
134   return document.getElementById( "tabustepreduce" ).value;
135 }
136
```

```
137
138 function tabu_search( f ){
139
140   var shortTerm = [];
141   var mediumTerm = [];
142   var longTerm = setupLongTermMemory(getLongSize());
143
144   var minimumHist = [];
145
146   var point = new TabuPoint( f, 0, 0 );
147   var minimum = point;
148
149   var initialInterval = 1;
150   var interval = initialInterval;
151   minimum.interval = initialInterval;
152
153   resetEvalCount();
154   updateTabuMinimumHistory( minimumHist, minimum );
155
156   setRunning();
157
158   var isInShortTermMem = function( p ){
159     var found = false;
160     shortTerm.forEach( function( s ){
161       if( s.isEqual( p ) ){
162         found = true;
163       }
164     });
165     return found;
166   };
167
168   var improvementCounter = 0;
169
170   clear_screen();
171
172   var step = function(){
173     var nextSteps = [];
174     ["x1", "x2"].forEach( function( param ){
175       var inc = point.clone();
176       var dec = point.clone();
177       inc[param] += interval;
178       dec[param] -= interval;
179       if( !isInShortTermMem( inc ) && inc.valid() ){
180         nextSteps.push( inc );
181       }
182       if( !isInShortTermMem( dec ) && dec.valid() ){
183         nextSteps.push( dec );
184       }
185     } );
186     var best = nextSteps.min( tabuMin );
187     if( nextSteps.length === 0 ){
188       window.console.log( "All␣points␣are␣Tabu" );
189       best = point;
190     }
191
192     if( best.getFValue() < point.getFValue() ){
193       //pattern move
194       var change = { x1: best.x1 - point.x1, x2: best.x2 - point.x2 };
195       var pattern = point.clone();
196       pattern.x1 += 2.0 * change.x1;
197       pattern.x2 += 2.0 * change.x2;
198       if( pattern.getFValue() < best.getFValue() && pattern.valid() ){
```

```
199         best = pattern;
200       }
201     }
202
203     // store old point for Line Drawing purposes
204     var oldPoint = point;
205
206     point = best;
207
208     addToMemory( shortTerm, best, getShortSize() );
209     considerForMediumTermMemory( mediumTerm, best, getMediumSize() );
210     addToLongTermMemory( longTerm, best );
211
212     improvementCounter += 1;
213     if( point.getFValue() < minimum.getFValue() ){
214       improvementCounter = 0;
215       minimum = point;
216       minimum.interval = interval;
217       updateTabuMinimumHistory( minimumHist, minimum );
218     }
219
220     var colour = "green";
221     window.console.log
222     if( improvementCounter == getIntensifyStep() ){
223       //Intensify
224       window.console.log( "intensifying" );
225       point = getAveragePoint( mediumTerm, f );
226       colour = "red";
227     } else if( improvementCounter == getDiversifyStep() ) {
228       //Diversify
229       window.console.log( "diversifying" );
230       point = getDiversePointFromLongTermMemory(longTerm, f);
231       window.console.log( "␣point:␣" + point.x1 + ",␣" + point.x2 );
232       // clear the minimum term memory
233       minimumTerm = [];
234       //reset the step Size
235       interval = initialInterval;
236       colour = "cyan";
237     } else if( improvementCounter == getReduceStep() ){
238       //Step Size Reduction
239       window.console.log( "Step␣Size␣Reduce" );
240       point = minimum;
241       interval = 0.5 * minimum.interval;
242       colour = "yellow"
243       improvementCounter = 0;
244     } else {
245       connectPoints( oldPoint.x1, oldPoint.x2, point.x1, point.x2 );
246     }
247     drawPoint( point.x1, point.x2, colour );
248
249     if( point.getFValue() < minimum.getFValue() ){
250       improvementCounter = 0;
251       minimum = point;
252       minimum.interval = interval;
253       updateTabuMinimumHistory( minimumHist, minimum );
254     }
255
256     if( getEvalCount() < 1000 - 10 ){
257       window.setTimeout( step, getItterationPause() );
258     } else {
259       logMinimumHistory( minimumHist );
260       finishRunning();
```

```
    }
  }
  step();
}
```

A.7 genetic_algorithm.js

```
"use strict";

function GAPoint(){
  var precision = 16;
  this.x1 = Array(precision);
  this.x2 = Array(precision);

  var getSinglePoint = function( xval ){
    var val = -6;
    for( var i = 0; i< precision; i++ ){
      if( xval[i] ){
        val += 6.0 * Math.pow(0.5, i);
      }
    }
    return val;
  }

  this.getValue = function(){
    return [ getSinglePoint( this.x1 ), getSinglePoint( this.x2 ) ];
  }
  this.getFunctionValue = function( f ){
    if( this.fvalue === undefined ){
      var x = this.getValue();
      this.fvalue = f( x[0], x[1] );
    }
    return this.fvalue;

  }
}

function getRandomGAPoint(){
  var point = new GAPoint();
   var randomizeBool = function(){
    var a;
    if(Math.random()<.5){
      a = true;
    } else {
      a = false;
    }
    return a;
  };
  for( var i = 0; i< point.x1.length; i++ ){
    point.x1[i] = randomizeBool();
    point.x2[i] = randomizeBool();
  }
  return point;
}

function getSeveralRandomGAPoints( N ){
  var points = Array(N);
  for( var i = 0; i< N; i++ ){
    points[i] = getRandomGAPoint();
  }
  return points;
```

```
55  }
56
57  function drawGAPoints( points, colour ){
58
59    points.forEach( function(p) {
60      var x = p.getValue();
61      drawPoint( x[0], x[1], colour );
62    } );
63
64  }
65
66
67  function swapPoints( points, i, j ){
68    var tmp = points[i];
69    points[i] = points[j];
70    points[j] = tmp;
71  }
72
73  function getRandomInt(min, max) {
74      return Math.floor(Math.random() * (max - min + 1)) + min;
75  }
76
77  function fisherYatesShuffle( list ){
78    for( var i = 0; i< list.length; i++ ){
79      var j = getRandomInt( i, list.length -1 );
80      swapPoints( list, i,  j);
81    }
82  }
83
84
85  function split(a, n) {
86    var len = a.length;
87    var out = [];
88    var i = 0;
89    while (i < len) {
90      var size = Math.ceil((len - i) / n--);
91      out.push(a.slice(i, i += size));
92    }
93    return out;
94  }
95
96  function tournamentSelect( points, f, groups ){
97
98    var size = points.length / groups;
99    fisherYatesShuffle( points );
100   var splitGroups = split( points, groups );
101   var comparer = function( a, b ){
102     return a.getFunctionValue(f) < b.getFunctionValue(f) ? a : b;
103   }
104
105   var selected = [];
106
107   splitGroups.forEach( function(g){
108     selected.push( g.min(comparer) );
109   } );
110   return selected;
111 }
112
113 function fitnessProportionateSelect( points, f, N ){
114   var sum = 0;
115   //the function has range that's roughly -100 to 100
116   // it goes a little lower than this, but we can still use 100 - function as a
```

```
117   // fitness score
118   points.forEach( function(p){
119     sum += 100 - p.getFunctionValue(f);
120   } );
121   var selected = [];
122   for( var i = 0; i < N; i++ ){
123     var r = Math.random();
124     var j = 0;
125     while( r > 0 && j < points.length ){
126       r -= (100-points[j].getFunctionValue(f))/sum;
127       j++;
128     }
129     if( points[j-1] === undefined ) alert("scary");
130     selected.push( points[j-1] );
131   }
132   return selected;
133 }
134
135 function doSingleValCrossover( p1, p2, c1, c2, val ){
136   var crossoverPoint1 = getRandomInt( 1,
137     Math.floor( ( p1[val].length -1) * 0.75 ) );
138   var crossoverPoint2 = getRandomInt( crossoverPoint1,  p1[val].length );
139   for( var i = 0; i < p1.x1.length; i++ ){
140     if( i < crossoverPoint1 || i > crossoverPoint2 ){
141       c1[val][i] = p1[val][i];
142       c2[val][i] = p2[val][i];
143     } else {
144       c2[val][i] = p1[val][i];
145       c1[val][i] = p2[val][i];
146     }
147   }
148 }
149
150 function crossTwoPoints( p1, p2 ){
151   var c1 = new GAPoint();
152   var c2 = new GAPoint();
153   doSingleValCrossover( p1, p2, c1, c2, "x1" );
154   doSingleValCrossover( p1, p2, c1, c2, "x2" );
155   return [c1, c2];
156 }
157
158
159 function breedPoints( selectedPoints, nextGenSize ){
160   var newPoints = [];
161   for( var i = 0; i < nextGenSize/2; i += 1){
162     // pair each point in turn with a random point, and breed them
163     newPoints = newPoints.concat(
164       crossTwoPoints(
165         selectedPoints[i%selectedPoints.length],
166         selectedPoints[getRandomInt(0, selectedPoints.length-1)]
167       )
168     );
169   }
170   return newPoints;
171 }
172
173
174 function mutatePoint( point, value ){
175   var bit = getRandomInt( 0, point[value].length -1 );
176   if( point[value][bit]  ){
177     point[value][bit] = false;
178   } else {
```

```
179     point[value][bit] = true;
180   }
181 }
182
183 function getMutationRate(){
184   return document.getElementById( "gamutate" ).value;
185 }
186
187 function mutatePoints(points){
188
189   var mutationProbability = getMutationRate();
190
191   points.forEach( function(p){
192     if( Math.random() < mutationProbability ){
193       var value = "x1";
194       if(Math.random() < 0.5 ){
195         value = "x2";
196       }
197       mutatePoint( p, value );
198     }
199   } );
200 }
201
202 function updateGAMinimumHistory( points, f, minimumHistory ){
203   var comparer = function( a, b ){
204     return a.getFunctionValue(f) < b.getFunctionValue(f) ? a : b;
205   }
206
207   var min = points.min(comparer)
208   var x = min.getValue();
209   var y = min.getFunctionValue( f );
210
211   if( minimumHistory.length === 0 ||
212     minimumHistory[minimumHistory.length-1].y > y ){
213     var o = {};
214     o.x1 = x[0];
215     o.x2 = x[1];
216     o.y = y;
217     o.evaluations = getEvalCount();
218     minimumHistory.push( o );
219   }
220 }
221
222
223 function getGAPopulation(){
224   return document.getElementById( "gapopulation" ).value;
225 }
226
227 function getGAParentsCount(){
228   return document.getElementById( "gaparents" ).value;
229 }
230
231 function getGAParents(points, f){
232   var stratergy = document.getElementById( "gastratergy" ).value;
233   if( stratergy === "tournament" ){
234     return  tournamentSelect( points, f, getGAParentsCount() );
235   } else if( stratergy === "frequency" ){
236     return fitnessProportionateSelect( points, f, getGAParentsCount() );
237   }
238 }
239
240 function GAItteration( points, f, minimumHistory ){
```

```
241   clear_screen();
242
243   drawGAPoints( points, "green" );
244
245   var reproducing = getGAParents(points,  f );
246
247   drawGAPoints( reproducing, "red" );
248
249   var nextGen = breedPoints(reproducing, points.length);
250
251   updateGAMinimumHistory( points, f, minimumHistory);
252
253   var last = minimumHistory[minimumHistory.length-1];
254
255   drawPoint( last.x1, last.x2, "blue" );
256
257   mutatePoints( nextGen );
258
259   if( getEvalCount() < 1000 - nextGen.length){
260     window.setTimeout( function(){
261       GAItteration( nextGen, f, minimumHistory );
262     }, getItterationPause() );
263   } else {
264     logMinimumHistory( minimumHistory );
265     finishRunning();
266   }
267 }
268
269 function genetic_algorithm( f ){
270   resetEvalCount();
271   var nPoints = getGAPopulation();
272   var points = getSeveralRandomGAPoints(nPoints);
273   setRunning();
274   GAItteration(points, f, []);
275 }
```