Short Term Course on

"Advanced Optimization Techniques (AOT-15)"
May 18-22, 2015
Short Term Course

Advanced Optimization Techniques

Day 1 : (18-May-2015)
Short Term Course on Advanced Optimization Techniques
What optimization is all about

• **OPTIMIZATION** is the use of specific methods to determine the most cost-effective and efficient solution to a problem or design for a process.

• This technique is one of the major quantitative tools in industrial decision making.

• A wide variety of challenges in the design, construction, operation, and analysis of any industrial processes can be resolved by optimization.
Why optimize???

- To improve the initial design of equipment
- To enhance the operation of that equipment
- To realize the largest production
- The greatest profit
- The least energy usage and so on.
Types of optimization problems

**LINEAR PROBLEMS**
- Graphical method
- Simplex
- Dual simplex
- Two phase
- BIG-M etc.

**NON-LINEAR PROBLEMS**
- Newton’s method
- Steepest decent
- Conjugate gradient etc.

**PROBLEMS WITHOUT CONSTRAINTS**
- Langrangian method
- Quadratic Programming method
- Penalty and barrier method
- GRG method etc.

**PROBLEMS WITH CONSTRAINTS**
- Langrangian method
- Quadratic Programming method
- Penalty and barrier method
- GRG method etc.
LINEAR PROGRAMMING
Decision variables – Variables representing levels of activity of a firm.

Objective function - A linear mathematical relationship describing an objective of the firm, in terms of decision variables, that is to be maximized or minimized.

Constraints - Restrictions placed on the firm by the operating environment stated in linear relationships of the decision variables.

Parameters - Numerical coefficients and constants used in the objective function and constraint equations.
DEVELOPMENT OF A LP MODEL

- LP applied extensively to problems areas -
  - medical, transportation, operations,
  - financial, marketing, accounting,
  - human resources, and agriculture.

- Development and solution of all LP models can be examined in a four step process:
  1. identification of the problem as solvable by LP
  2. formulation of the mathematical model.
  3. solution.
  4. interpretation.
BASIC STEPS OF DEVELOPING A LP MODEL

Formulation

– Process of translating problem scenario into simple LP model framework with a set of mathematical relationships.

Solution

– Mathematical relationships resulting from formulation process are solved to identify optimal solution.
PROBLEM DEFINITION: FERTILIZER MIX EXAMPLE (1 of 7)

- Two brands of fertilizer available - Super-Gro, Crop-Quick.
- Field requires at least 16 pounds of nitrogen and 24 pounds of phosphate.
- Super-Gro costs $6 per bag, Crop-Quick $3 per bag.
- Problem: How much of each brand to purchase to minimize total cost of fertilizer given the following data?

<table>
<thead>
<tr>
<th>Brand</th>
<th>Nitrogen (lb/bag)</th>
<th>Phosphate (lb/bag)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super-gro</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Crop-quick</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
PROBLEM DEFINITION:
FERTILIZER MIX EXAMPLE (2 of 7)

Decision Variables:

\[ x_1 = \text{bags of Super-Gro} \]
\[ x_2 = \text{bags of Crop-Quick} \]

The Objective Function:

Minimize \[ Z = 6x_1 + 3x_2 \]

Model Constraints:

\[ 2x_1 + 4x_2 \geq 16 \text{ lb (nitrogen constraint)} \]
\[ 4x_1 + 3x_2 \geq 24 \text{ lb (phosphate constraint)} \]
\[ x_1, x_2 \geq 0 \text{ (non-negativity constraint)} \]
Minimize $Z = 6x_1 + 3x_2$
subject to: $2x_1 + 4x_2 \geq 16$
$4x_2 + 3x_2 \geq 24$
$x_1, x_2 \geq 0$

Graph of Both Model Constraints
Minimize $Z = 6x_1 + 3x_2$
subject to:  
\[2x_1 + 4x_2 \geq 16\]
\[4x_2 + 3x_2 \geq 24\]
\[x_1, x_2 \geq 0\]
Minimize $Z = 6x_1 + 3x_2$
subject to: $2x_1 + 4x_2 \geq 16$
$4x_2 + 3x_2 \geq 24$
$x_1, x_2 \geq 0$

$x_1 = 0$ bags of Super-gro
$x_2 = 8$ bags of Crop-quick
$Z = $24

Optimum Solution Point
IRREGULAR TYPES OF LINEAR PROGRAMMING PROBLEMS

• For some linear programming models, the general rules do not apply.

• Special types of problems include those with:
  - Redundancy
  - Infeasible solutions
  - Unbounded solutions
  - Multiple optimal solutions
**Redundancy**: A redundant constraint is a constraint that does not affect the feasible region in any way.

Maximize Profit

\[ \text{Maximize Profit} = 2X + 3Y \]

subject to:

\[ X + Y \leq 20 \]
\[ 2X + Y \leq 30 \]
\[ X \leq 25 \]
\[ X, Y \geq 0 \]
Infeasibility: A condition that arises when an LP problem has no solution that satisfies all of its constraints.

\[ X + 2Y \leq 6 \]
\[ 2X + Y \leq 8 \]
\[ X \geq 7 \]
**Unboundedness:** Sometimes an LP model will not have a finite solution

Maximize profit

\[ = $3X + $5Y \]

subject to:

\[ X \geq 5 \]

\[ Y \leq 10 \]

\[ X + 2Y \geq 10 \]

\[ X, Y \geq 0 \]
MULTIPLE OPTIMAL SOLUTIONS

• An LP problem may have more than one optimal solution.

  – Graphically, when the isoprofit (or isocost) line runs parallel to a constraint in the problem which lies in the direction in which isoprofit (or isocost) line is located.

  – In other words, when they have the same slope.
EXAMPLE: MULTIPLE OPTIMAL SOLUTIONS

Maximize profit = $3x + $2y

Subject to:

\[ 6X + 4Y \leq 24 \]
\[ X \leq 3 \]
\[ X, Y \geq 0 \]
EXAMPLE: MULTIPLE OPTIMAL SOLUTIONS

- At profit level of $12, isoprofit line will rest directly on top of first constraint line.
- This means that any point along the line between corner points 1 and 2 provides an optimal X and Y combination.
SIMPLEX METHOD

• Starting point: set of equations (objective function along with the equality constraints of the problem in canonical form)
• Convert to a minimization problem
• Convert sign of inequalities to get nonnegative values of bi
• Introduce slack variables
• Basic feasible solution from canonical equations
• Improve and optimize the solution
Example Maximize $F = x_1 + 2x_2 + x_3$
subject to

\[
\begin{align*}
2x_1 + x_2 - x_3 & \leq 2 \\
-2x_1 + x_2 - 5x_3 & \geq -6 \\
4x_1 + x_2 + x_3 & \leq 6 \\
x_i & \geq 0, \ i = 1, 2, 3
\end{align*}
\]

Solution
Change the sign of the objective function to convert it to a minimization problem and the signs of the inequalities (where necessary) so as to obtain nonnegative values of $b_i$.

Minimize $f = -x_1 - 2x_2 - x_3$

subject to

\[
\begin{align*}
2x_1 + x_2 - x_3 & \leq 2 \\
2x_1 - x_2 + 5x_3 & \leq 6 \\
4x_1 + x_2 + x_3 & \leq 6 \\
x_i & \geq 0, \ i = 1 \text{ to } 3
\end{align*}
\]

Introduce slack variables $x_4 \geq 0$, $x_5 \geq 0$, and $x_6 \geq 0$
Canonical form:

\[ 2x_1 + x_2 - x_3 + x_4 = 2 \]
\[ 2x_1 - x_2 + 5x_3 + x_5 = 6 \]
\[ 4x_1 + x_2 + x_3 + x_6 = 6 \]
\[ -x_1 - 2x_2 - x_3 - f = 0 \]

\( x_4, x_5, x_6, \) and \(-f\) treated as basic variables

Basic solution:

\( x_4 = 2, x_5 = 6, x_6 = 6 \) (basic variables)
\( x_1 = x_2 = x_3 = 0 \) (nonbasic variables)

\( f = 0 \)

\( \text{c''_1 = -1, c''_2 = -2, c''_3 = -1, not optimum} \)

To improve the present basic feasible solution, we first decide the variable \((x_s)\) to be brought into the basis as

\( c''_s = \min(c''_j < 0) = c''_2 = -2 \)

Thus \( x_2 \) enters the next basic set. To obtain the new canonical form, we select the pivot element \( a''_{rs} \) such that

\[ \frac{b''_r}{a''_{rs}} = \min \left( \frac{b''_i}{a''_{is}} \right) \quad a''_{is} > 0 \]
Result of pivoting:

<table>
<thead>
<tr>
<th>Basic Variables</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>-f</th>
<th>bi''</th>
<th>bi'' / ais'' for ais''&gt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>x2</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>-f</td>
<td>3</td>
<td>0</td>
<td>-3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

most negative c_i'' (x3 enters the next basis)

<table>
<thead>
<tr>
<th>Basic Variables</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>-f</th>
<th>bi''</th>
<th>bi'' / ais'' for ais''&gt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>x4</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>2</td>
<td>-1</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>-f</td>
<td>-1</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

most negative c_i'' (x2 enters the next basis)
Result of pivoting:

<table>
<thead>
<tr>
<th></th>
<th>x2</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>5/4</th>
<th>1/4</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1/4</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-3/2</td>
<td>-1/2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>11/4</td>
<td>3/4</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since all $c_i^*$ $\geq 0$, the present solution is optimum.
Unbounded Solution

**Example**

Minimize

\[ f = -3x_1 - 2x_2 \]

subject to

\[ x_1 - x_2 \leq 1 \]
\[ 3x_1 - 2x_2 \leq 6 \]
\[ x_1 \geq 0, \quad x_2 \geq 0 \]

**Solution**

Introducing slack variables \( x_3 \geq 0 \) and \( x_4 \geq 0 \),

\[ x_1 - x_2 + x_3 = 1 \]
\[ 3x_1 - 2x_2 + x_4 = 6 \]
\[ -3x_1 - 2x_2 - f = 0 \]

Basic feasible solution corresponding to this canonical form

\[ x_3 = 1, \quad x_4 = 6 \) (basic variables)\]
\[ x_1 = x_2 = 0 \) (non basic variables)\]
\[ f = 0 \]
### Basic Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$-f$</th>
<th>$b_i''$</th>
<th>$b_i''$ / $a_{i*}$ for $a_{i*} &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>3</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>$-f$</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

↑ most negative

Result of pivoting:

<table>
<thead>
<tr>
<th>Variables</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$-f$</th>
<th>$b_i''$</th>
<th>$b_i''$ / $a_{i*}$ for $a_{i*} &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>1</td>
<td>-3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$-f$</td>
<td>0</td>
<td>-5</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

↑ most negative

↑ smallest value
The value of $f$ can be decreased indefinitely without violating any of the constraints if we bring $x_3$ into the basis.

**NO BOUNDED SOLUTION**
Infinite Number of Solutions

Example
Minimize \[ f = -40x_1 - 100x_2 \]
subject to
\[
\begin{align*}
10x_1 + 5x_2 & \leq 2500 \\
4x_1 + 10x_2 & \leq 2000 \\
2x_1 + 3x_2 & \leq 900 \\
x_1 \geq 0, & \quad x_2 \geq 0
\end{align*}
\]

Solution
Introducing slack variables \( x_3 \geq 0, x_4 \geq 0 \) and \( x_5 \geq 0 \),
\[
\begin{align*}
10x_1 + 5x_2 + x_3 & = 2500 \\
4x_1 + 10x_2 + x_4 & = 2000 \\
2x_1 + 3x_2 + x_5 & = 900 \\
-40x_1 - 100x_2 - f & = 0
\end{align*}
\]
### Basic Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$-f$</th>
<th>$b_i''$ / $a_i''$ for $a_i'' &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_3$</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,500 500</td>
</tr>
<tr>
<td>$x_4$</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2,000 200</td>
</tr>
<tr>
<td>$x_5$</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>900 300</td>
</tr>
<tr>
<td>$-f$</td>
<td>-40</td>
<td>-100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Result of pivoting:

<table>
<thead>
<tr>
<th>Variables</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$-f$</th>
<th>$b_i''$ / $a_i''$ for $a_i'' &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_3$</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>-1/2</td>
<td>0</td>
<td>0</td>
<td>1,500</td>
</tr>
<tr>
<td>$x_2$</td>
<td>4/8</td>
<td>1</td>
<td>0</td>
<td>1/10</td>
<td>0</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>$x_5$</td>
<td>8/10</td>
<td>0</td>
<td>0</td>
<td>-3/10</td>
<td>1</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td>$-f$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>20,000</td>
</tr>
</tbody>
</table>

Since all $c_i'' \geq 0$, the present solution is optimum.
The optimum values are given by

- $x_2 = 200$, $x_3 = 1500$, $x_5 = 300$ (basic variables)
- $x_1 = x_4 = 0$ (non basic variables)
- $f_{\text{min}} = -20,000$
The cost coefficient corresponding to the non basic variable $x_1(c''_1)$ is zero. This is an indication that an alternative solution exists.

$x_1$ can be brought into the basis and the resulting new solution will also be an optimal basic feasible solution.

<table>
<thead>
<tr>
<th>Basic Variables</th>
<th>Variables</th>
<th>$X_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$-f$</th>
<th>$b_i'''$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1/8</td>
<td>-1/16</td>
<td>0</td>
<td>0</td>
<td>1500/8</td>
</tr>
<tr>
<td>$x_2$</td>
<td></td>
<td>0</td>
<td>1</td>
<td>-1/20</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>125</td>
</tr>
<tr>
<td>$x_5$</td>
<td></td>
<td>0</td>
<td>0</td>
<td>-1/10</td>
<td>-1/4</td>
<td>1</td>
<td>0</td>
<td>150</td>
</tr>
<tr>
<td>$-f$</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-10</td>
<td>0</td>
<td>1</td>
<td>20,000</td>
</tr>
</tbody>
</table>

The solution corresponding to this canonical form is given by

$x_1 = 1500/8, \quad x_2 = 125, \quad x_5 = 150$ (basic variables)

$x_3 = x_4 = 0$ (non basic variables)

$f_{\text{min}} = -20,000$

Thus the value of $f$ has not changed compared to the preceding value.
\[
X_1 = \begin{cases} 
0 \\
200 \\
1500 \\
0 \\
300 
\end{cases} \quad \text{and} \quad X_2 = \begin{cases} 
1500/8 \\
125 \\
0 \\
0 \\
150 
\end{cases}
\]

\[
X^* = \lambda X_1 + (1 - \lambda) X_2
\]

\[
X^* = \begin{bmatrix} 
x_1^* \\
x_2^* \\
x_3^* \\
x_4^* \\
x_5^*
\end{bmatrix} = \begin{bmatrix} 
(1 - \lambda) \frac{1500}{8} \\
200\lambda + (1 - \lambda)125 \\
1500\lambda \\
0 \\
300\lambda + (1 - \lambda)150
\end{bmatrix} = \begin{bmatrix} 
(1 - \lambda) \frac{1500}{8} \\
125 + 75\lambda \\
1500\lambda \\
0 \\
150 + 150\lambda
\end{bmatrix}
\]

\[0 \leq \lambda \leq 1\]
TWO PHASE OF SIMPLEX
TWO PHASES OF THE SIMPLEX METHOD

When initial feasible canonical form may not be readily available (LP problem does not have slack variables for some of the equations or when the slack variables have negative coefficients).

Phase I: simplex algorithm used to find if the feasible solution exists.

Phase II: simplex algorithm to find whether the problem has a bounded optimum. If a bounded optimum exists, it finds the basic feasible solution that is optimal.
Example  Minimize \( f = 2x_1 + 3x_2 + 2x_3 - x_4 + x_5 \)
subject to constrains
\[
3x_1 - 3x_2 + 4x_3 + 2x_4 - x_5 = 0
\]
\[
x_1 + x_2 + x_3 + 3x_4 + x_5 = 2
\]
\[
x_i \geq 0, \quad i = 1 \text{ to } 5
\]

Solution  Introducing the artificial variables \( y_1 \geq 0 \) and \( y_2 \geq 0 \), the equations can be written as follows:
\[
3x_1 - 3x_2 + 4x_3 + 2x_4 - x_5 + y_1 = 0
\]
\[
x_1 + x_2 + x_3 + 3x_4 + x_5 + y_2 = 2
\]
\[
2x_1 + 3x_2 + 2x_3 - x_4 + x_5 - f = 0
\]
By defining the infeasibility form \( w \) as
\[
w = y_1 + y_2
\]
the complete array of equations can be written as
\[
3x_1 - 3x_2 + 4x_3 + 2x_4 - x_5 + y_1 = 0
\]
\[
x_1 + x_2 + x_3 + 3x_4 + x_5 + y_2 = 2
\]
\[
2x_1 + 3x_2 + 2x_3 - x_4 + x_5 - f = 0
\]
\[
y_1 + y_2 - w = 0
\]
The last equation can be written as

\[-4x_1 + 2x_2 - 5x_3 - 5x_4 + 0x_5 - w = -2\]

**Phase I:**

<table>
<thead>
<tr>
<th>Basic Variables</th>
<th>Variables</th>
<th>Artificial Variables</th>
<th>(b_i'' / a_{is}'') for (a_{is}'' &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_1)</td>
<td>3</td>
<td>-3</td>
<td>4</td>
</tr>
<tr>
<td>(y_2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(-f)</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>(-w)</td>
<td>-4</td>
<td>4</td>
<td>-5</td>
</tr>
</tbody>
</table>

Since there is a tie between \(d''_3\) and \(d''_4\), \(d''_4\) is selected arbitrarily as the most negative \(d''_i\) for pivoting (\(x_4\) enters the next basis).
Result of pivoting:

<table>
<thead>
<tr>
<th></th>
<th>3/2</th>
<th>-3/2</th>
<th>2</th>
<th>1</th>
<th>-1/2</th>
<th>1/2</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>-7/2</td>
<td></td>
<td>-5</td>
<td>0</td>
<td>5/2</td>
<td>-3/2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/11</td>
<td>←</td>
</tr>
</tbody>
</table>

Result of pivoting (since $y_1$ and $y_2$ are dropped from basis, the columns corresponding to them need not be filled):

<table>
<thead>
<tr>
<th></th>
<th>6/11</th>
<th>0</th>
<th>7/11</th>
<th>2/11</th>
<th>Dropped</th>
<th>6/11</th>
</tr>
</thead>
<tbody>
<tr>
<td>x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>-7/11</td>
<td>1</td>
<td>-10/11</td>
<td>0</td>
<td>5/11</td>
<td>4/11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-f</td>
<td>98/22</td>
<td>0</td>
<td>118/22</td>
<td>0</td>
<td>-4/22</td>
<td>-6/11</td>
</tr>
<tr>
<td>-w</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- The present basic feasible solution does not contain any of the artificial variables $y_1$ and $y_2$
- The value of $w$ is reduced to 0

$\Rightarrow$ Phase I is completed
Phase II: Dropping the $w$ row

<table>
<thead>
<tr>
<th>Basic Variables</th>
<th>Variables</th>
<th>$b_i''$</th>
<th>$b''$ / $a_is''$ for $a_is'' &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_4$</td>
<td>$6/11$</td>
<td>$7/11$</td>
<td>$2/11$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$-7/11$</td>
<td>$1$</td>
<td>$-10/11$</td>
</tr>
<tr>
<td>$-f$</td>
<td>$98/22$</td>
<td>$0$</td>
<td>$118/22$</td>
</tr>
</tbody>
</table>

Result of pivoting:

<table>
<thead>
<tr>
<th>$x_4$</th>
<th>$4/5$</th>
<th>$-2/5$</th>
<th>$1$</th>
<th>$1$</th>
<th>$0$</th>
<th>$2/5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_5$</td>
<td>$-7/5$</td>
<td>$11/5$</td>
<td>$-2$</td>
<td>$0$</td>
<td>$1$</td>
<td>$4/5$</td>
</tr>
<tr>
<td>$-f$</td>
<td>$21/5$</td>
<td>$2/5$</td>
<td>$5$</td>
<td>$0$</td>
<td>$0$</td>
<td>$-2/5$</td>
</tr>
</tbody>
</table>

Now, since all $c_i''$ are nonnegative, phase II is completed. The (unique) optimal solution is given by

- $x_1 = x_2 = x_3 = 0$ (non basic variables)
- $x_4 = 2/5$, $x_5 = 4/5$ (basic variables)
- $f_{\text{min}} = 2/5$
BIG-M METHOD
Big-M Method of solving LPP

- In Big-M method, the notion is to make the artificial variables, through their coefficients in the objective function, so costly or unprofitable that any feasible solution to the real problem would be preferred....unless the original instance possessed no feasible solutions at all.

- We need to assign, in the objective function, coefficients to the artificial variables that are either very small (maximization problem) or very large (minimization problem); whatever this value, let us call it Big M.

- Indeed, the penalty is so costly that unless any of the respective variables' inclusion is warranted algorithmically, such variables will never be part of any feasible solution. This method removes artificial variables from the basis.

- The penalty will be designated by +M for minimization problem and by –M for a maximization problem and also M>0.
**Example:** Minimize \( Z = 600X_1 + 500X_2 \)
subject to constraints,

\[
\begin{align*}
2X_1 + X_2 &\geq 80 \\
X_1 + 2X_2 &\geq 60
\end{align*}
\]

**Step1:** Convert the LP problem into a system of linear equations.

We do this by rewriting the constraint inequalities as equations by subtracting new “surplus & artificial variables” and assigning them zero & +M coefficients respectively in the objective function as shown below

So the Objective Function would be:

\[
Z = 600X_1 + 500X_2 + 0.S_1 + 0.S_2 + M.A_1 + M.A_2
\]

subject to constraints,

\[
\begin{align*}
2X_1 + X_2 - S_1 + A_1 &= 80 \\
X_1 + 2X_2 - S_2 + A_2 &= 60 \\
X_1, X_2, S_1, S_2, A_1, A_2 &\geq 0
\end{align*}
\]
Step 2: Obtain a Basic Solution to the problem.
We do this by putting the decision variables \(X_1=X_2=S_1=S_2=0\), so that \(A_1=80\) and \(A_2=60\).
These are the initial values of *artificial variables*.

Step 3: Form the Initial Tableau as shown.

<table>
<thead>
<tr>
<th>(C_B)</th>
<th>Basic Variable (B)</th>
<th>Basic Sol’n (X_B)</th>
<th>(C_j)</th>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(A_1)</th>
<th>(A_2)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>(A_1)</td>
<td>80</td>
<td>600</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>M</td>
<td>(A_2)</td>
<td>60</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
<td>(3M)</td>
</tr>
<tr>
<td>(C_j-Z_j)</td>
<td>600-3M</td>
<td>500-3M</td>
<td>M</td>
<td>M</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
It is clear from the tableau that \( X_2 \) will enter and \( A_2 \) will leave the basis. Hence 2 is the key element in pivotal column. Now, the new row operations are as follows:

\[
\begin{align*}
R_2(\text{New}) &= R_2(\text{Old})/2 \\
R_1(\text{New}) &= R_1(\text{Old}) - 1 \cdot R_2(\text{New})
\end{align*}
\]

<table>
<thead>
<tr>
<th>( C_B )</th>
<th>Basic Variable (B)</th>
<th>Basic Sol’n (( X_B ))</th>
<th>( C_j )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( A_1 )</th>
<th>( \text{Ratio} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>( A_1 )</td>
<td>50</td>
<td>( 600 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( M )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>( X_2 )</td>
<td>30</td>
<td>( 500 )</td>
<td>( 1 )</td>
<td>( 0 )</td>
<td>( -1/2 )</td>
<td>( 0 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( Z )</td>
<td>( 3M/2 + 250 )</td>
<td>( 500 )</td>
<td>( -M )</td>
<td>( M/2 - 250 )</td>
<td>( M )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( C_j - Z_j )</td>
<td>( 350 - 3M/2 )</td>
<td>( 0 )</td>
<td>( M )</td>
<td>( 250 - M/2 )</td>
<td>( 0 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
It is clear from the tableau that $X_1$ will enter and $A_1$ will leave the basis. Hence 2 is the key element in pivotal column. Now, the new row operations are as follows:

\[ R_1(\text{New}) = R_1(\text{Old}) \times \frac{2}{3} \]
\[ R_2(\text{New}) = R_2(\text{Old}) - \left(\frac{1}{2}\right) R_1(\text{New}) \]

<table>
<thead>
<tr>
<th>$C_B$</th>
<th>Basic Variable (B)</th>
<th>Basic Sol’n ($X_B$)</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>$X_1$</td>
<td>100/3</td>
<td>1</td>
<td>0</td>
<td>-2/3</td>
<td>1/3</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>$X_2$</td>
<td>40/3</td>
<td>0</td>
<td>1</td>
<td>1/3</td>
<td>-2/3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$Z_j$</td>
<td>600 500</td>
<td>-700/3</td>
<td>-400/3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_j$</td>
<td>$Z_j$</td>
<td>0 0</td>
<td>700/3</td>
<td>400/3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Since all the values of \((C_j-Z_j)\) are either zero or positive and also both the artificial variables have been removed, an optimum solution has been arrived at with

\[ X_1 = \frac{100}{3}, \quad X_2 = \frac{40}{3} \quad \text{and} \quad Z = \frac{80,000}{3}. \]
DUAL, DUALITY & PRIMAL
Duality

**Primal**

Decision variables: $x$

\[ \text{min} \ Z = cx \]

s.t. \[ Ax \geq b \quad n \]

\[ x \geq 0 \quad m \]

**Dual**

Decision variables: $y$

\[ \text{max} \ W = yb \]

s.t. \[ yA \leq c \quad m \]

\[ y \geq 0 \quad n \]
Primal Problem

\[ a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2 \]
\[ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m \]
\[ c_1x_1 + c_2x_2 + \cdots + c_nx_n = f \]

\( (x_i \geq 0, i = 1 \text{ to } n, \text{ and } f \text{ is to be minimized}) \)
Dual Problem

\[ a_{11}y_1 + a_{12}y_2 + \cdots + a_{m1}y_m \leq c_1 \]
\[ a_{12}y_1 + a_{22}y_2 + \cdots + a_{m2}y_m \leq c_2 \]
\[ a_{1m}y_1 + a_{2m}y_2 + \cdots + a_{mn}y_m \leq c_n \]
\[ b_1x_1 + b_2x_2 + \cdots + b_mx_m = v \]

\((y_i \geq 0, i = 1 to m, \text{ and } v \text{ is to be minimized})\)
Primal

- Objective function: Minimize $\mathbf{x}^T \mathbf{c}$
- Variable $x_i \geq 0$
- Variable unrestricted in sign $x_i$
- $j$th constraint, $x_i = y_j$ (equality)
- $j$th constraint, $x_i \geq y_j$ (inequality)
- Coefficient matrix $\mathbf{A} \equiv [\mathbf{A}_1 \ldots \mathbf{A}_m]$
- Right-hand-side vector $\mathbf{b}$
- Cost coefficients $\mathbf{c}$

Dual

- Maximise $\mathbf{b}^T \mathbf{y}$
- $i$th constraint $\leq y_i$ (inequality)
- $i$th constraint $= y_i$ (equality)
- $j$th variable unrestricted in sign $y_j$
- $j$th variable $\geq 0$
- Coefficient matrix $\equiv \mathbf{A}^T [\mathbf{A}_1, \ldots, \mathbf{A}_m]^T$
- Right-hand-side vector $\mathbf{c}$
- Cost coefficients $\mathbf{b}$
Example

**Primal**

\[
\begin{align*}
\text{Min } Z &= 3x_1 + 5x_2 \\
\text{s. to } &\quad x_1 \geq 4 \\
&\quad 2x_2 \geq 12 \\
&\quad 3x_1 + 2x_2 = 18
\end{align*}
\]

\[x_1 \geq 0, x_2 \text{ is unrestricted in sign}\]

**Dual**

\[
\begin{align*}
\text{Max } W &= 4y_1 + 12y_2 + 18y_3 \\
\text{s. to } &\quad y_1 + 3y_3 \leq 3 \\
&\quad 2y_2 + 2y_3 = 5
\end{align*}
\]

\[y_1, y_2 \geq 0, y_3 \text{ is unrestricted in sign}\]
Dual Simplex Method

Steps:-
1. Select row r as the pivot row such that $b_r = \min b_i < 0$
2. Select column s as the pivot column such that

$$\begin{vmatrix} \frac{c_s}{-a_{rs}} \\ \frac{c_i}{-a_{ri}} \end{vmatrix} = \min$$

If all $a_{rij} \geq 0$, the primal will not have any feasible (optimal) solution.

3. Carry out a pivot operation on $a_{rs}$
4. Test for optimality: If all $b_i \geq 0$, the current solution is optimal and hence stop the iterative procedure. Otherwise, go to step 1.
Example

Min $f = x_1 + 2x_2$

s. to

$2x_1 + x_2 \geq 4$

$x_1 + 2x_2 \geq 7$

$x_1 \geq 0, x_2 \geq 0$

Solution-

Introducing surplus variables $x_3$ and $x_4$

Min $f = x_1 + 2x_2$

s. to

$-2x_1 - x_2 + x_3 = -4$

$-x_1 - 2x_2 + x_4 = -7$

$x_1, x_2, x_3, x_4 \geq 0$

Here, No. of variables = 4
No. of equations = 2

Hence, 2 non-basic variables i.e. $x_1$ and $x_2$
Example

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$-f$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_3$</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-4</td>
</tr>
<tr>
<td>$x_4$</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-7</td>
</tr>
<tr>
<td>$-f$</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
| $\left(\begin{array}{c}c \\
-a \end{array}\right)$ | 1     | 1     |       |       |      |     |

Min. pivot row

pivot column
### Example

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
<th>(-f)</th>
<th>(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_3)</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>(x_1)</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>(-f)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-7</td>
</tr>
</tbody>
</table>

All values of \(b_i \geq 0\).

Therefore, the optimal solution is obtained.

The solution is \(\min f = 7\)

\[ x_1 = 7, \ x_3 = 10, \ x_2 = 0, \ x_1 = 0 \]
GAMS : A Software Approach to Mathematical Programming

Dr. Parul Mathuria
Centre For Energy and Environment
MNIT, Jaipur
Introduction

• The General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical optimization.

• Originally developed by a group of economists from the World Bank in order to facilitate the resolution of large and complex non-linear models on personal computer.

• Used for modeling problems following types
  – Linear optimizations
  – Non-linear optimizations
  – Mixed integer optimizations.

• Transform mathematical representation to the representations required by the solvers.
Basic Features

• Advantages of GAMS
  (i) Simplicity of implementation,
  (ii) Portability and transferability between users and systems
  (iii) Easiness of technical update because of the constant inclusion of new algorithms

• The separation of logic and data has allowed a problem to be increased in size without causing an increase in the complexity of the representation.

• Explanatory text can be made part of the definition of all symbols and is reproduced whenever associated values are displayed.
Introduction to GAMS

In this introduction note
• The general structure of the GAMS program
• Detailed illustration
• Description of the output file.
Architecture of GAMS programming

• The structure of the input GAMS Code decomposed in three modules corresponding respectively to
  – data entry
  – model specification
  – solve procedure
• General rules
• it is necessary to proceed to the declaration of any element before using it. In particular, sets must be declared at the very beginning of the program.
• To end every statement by a semicolon in order to avoid unexpected compilation error.
• GAMS allows for statement on several lines or several statement on the same line. This property can help to reduce the length of the code or facilitate printing.
• Capital and small letters are not distinguished in GAMS.
GAMS Model Structure

Stage 1: Data/ Sets/ Variables/ Equations Entry

• Specification of indices and data (parameters, sets).
• Listing of names and types of variables and equations (constraints and objective function).

• Entering Data
  1. Sets
  2. Scalars
  3. Parameters
  4. Data entry by Tables
  5. Data entry by lists

• Entering Variables
  1. Variables Types
  2. Variable Attributes
GAMS Model Structure

• Entering equations
  1. Types of equations
  2. Equation values

Stage 2: Defining Model

• VARIABLES declaration
• Definition of the equations (constraints and objective function).
• Specification of bounds, initial values and special options.
  1. Types of problems
  2. Model definition
Stage 3: Resolution

- SOLVE statement
- Call to the optimization solver
- Results display

Code must be saved with the extension .gms

Conventions for naming the extensions of the files are as follows:
  - Input file: Filename.GMS
  - Output file: Filename.LST
Example 1

minimize \( Z = X_1^2 + X_2^2 + X_3^3 \)

s.t. \( 1 - \frac{X_3}{X_2} > 0 \)

\( X_1 - X_2 > 0 \)

\( X_1 \rightarrow X_2^2 + X_2 X_3 - 4 = 0 \)

\( 0 < X_1 < 5 \)

\( 0 < X_2 < 3 \)

\( 0 < X_3 < 3 \)
Example 1

Initial starting point:

\[ X_1 = 4 \]
\[ X_2 = 2 \]
\[ X_3 = 2 \]
Example 1

First we should note that the inequality

\[ 1 - \frac{X_3}{X_2} > 0 \]  

\( (2) \)

can actually be rearranged in linear form as:

\[ X_2 - X_3 > 0 \]  

\( (3) \)

Using (3) instead of (2) is a much better choice, not only because we avoid the potential for a division by zero, but also because we obtain a linear constraint which is easier to handle.
$TITLE Test Problem
$OFFSYMREF
$OFFSYMMLIST

* Example from Problem 8.26 in "Engineering Optimization"
* by Reklaitis, Ravindran and Ragsdell (1983)
*

VARIABLES X1, X2, X3, Z ;
POSITIVE VARIABLES X1, X2, X3 ;

EQUATIONS CON1, CON2, CON3, OBJ ;

CON1.. X2 - X3 =G= 0 ;
CON2.. X1 - X3 =G= 0 ;
CON3.. X1 - X2**2 + X1*X2 - 4 =E= 0 ;
OBJ.. Z =E= SQR(X1) + SQR(X2) + SQR(X3) ;

* Upper bounds
  X1.UP = 5 ;
  X2.UP = 3 ;
  X3.UP = 3 ;

* Initial point
  X1.L = 4 ;
  X2.L = 2 ;
  X3.L = 2 ;

MODEL TEST / ALL / ;

OPTION LIMROW = 0;
OPTION LIMCOL = 0;

SOLVE TEST USING NLP MINIMIZING Z;
Example 1  - Dollar Control Options

$TITLE Test Problem
$OFFSYMREF
$OFFSYMLIST

$title: This option sets the title in the page header of the listing file.

$onsymxref [On/Off Symbol Reference]:
This option controls the following,

  • Collection of cross references for identifiers like sets, parameters, and variables.
  • Cross-reference report of all collected symbols in listing file.
  • Listing of all referenced symbols and their explanatory text by symbol type in listing file.

$onsymlist [On/Off Symbol List]:
Controls the complete listing of all symbols that have been defined and their text.

Example 1 - Input file

VARIABLES x1, x2, x3, z ;
POSITIVE VARIABLES x1, x2, x3 ;

• The keyword VARIABLES is used to list our variables $x_1$, $x_2$ and $x_3$. Note that $Z$, the objective function value must also be included.

• The keyword POSITIVE VARIABLES is used to specify the non-negativity of $x_1$, $x_2$, $x_3$.

• The objective function values should not be included here as in general it might take positive or negative values.

• Finally, note that the semicolon ; must be included to specify the end of the lists.
Example 1 - Input file

The keyword EQUATIONS is for listing the names of the constraints and objective function.

- Constraints: CON1, CON2, CON3
- Objective function: OBJ

Note the semi-colon ; is needed at the end.
Example 1 - Input file

* Upper bounds
  X1.UP = 5 ;
  X2.UP = 3 ;
  X3.UP = 3 ;

* Initial point
  X1.L = 4 ;
  X2.L = 2 ;
  X3.L = ;

Next in the input file we specify the upper bounds and initial values. This is done by adding a subfield to the variables.

.LO lower bound
.UP upper bound
.L level value, meaning actual value (initial or final)
Example 1 - Input file

```
MODEL TEST / ALL /

OPTION LIMROW = 0;
OPTION LIMCOL = 0;
```

The keyword **MODEL** is used to name our model and to specify which equations should be used. In this case, we name our model as TEST and specify that all equations be used.

The **OPTION** statements are used to suppress output for debugging the compilation of the equations.

**OPTION LIMROW = 0** and **OPTION LIMCOL = 0** to avoid long output files.
Example 1 - Input file

Finally, we invoke the optimization algorithm with the SOLVE statement. Here the format is as follows:

SOLVE (model name) USING (solver type) MINIMIZING (objective variable)
or
SOLVE (model name) USING (solver type) MAXIMIZING (objective variable)
Test Problem

* Example from Problem 8.26 in "Engineering Optimization" by Reklaitis, Ravindran and Ragsdell (1983)

VARIABLES X1, X2, X3, Z;
POSITIVE VARIABLES X1, X2, X3;

EQUATIONS CON1, CON2, CON3, OBJ;

CON1.. X2 - X3 =G= 0;
CON2.. X1 - X3 =G= 0;
CON3.. X1 - X2**2 + X1*X2 - 4 =E= 0;
OBJ.. Z =E= SQR(X1) + SQR(X2) + SQR(X3);

* Upper bounds
X1.UP = 5;
X2.UP = 3;
X3.UP = 3;

* Initial point
X1.L = 4;
X2.L = 2;
X3.L = 2;

MODEL TEST / ALL /
OPTION LIMROW = 0;
OPTION LIMCOL = 0;
SOLVE TEST USING NLP MINIMIZING Z;

MODEL STATISTICS
BLOCKS OF EQUATIONS 4 SINGLE EQUATIONS 4
BLOCKS OF VARIABLES 4 SINGLE VARIABLES 4
NON ZERO ELEMENTS 10 NON LINEAR N-Z 5
DERIVATIVE POOL 6 CONSTANT POOL 2
CODE LENGTH 57
GENERATION TIME 0.000 SECONDS

EXECUTION TIME = 0.110 SECONDS VERID TP5-00-038
GAMS 2.25 PC AT/XT 09/16/97
23:48:05 PAGE 3
Test Problem
Solution Report
SOLVE TEST USING NLP FROM LINE 34

SOLVE SUMMARY
MODEL TEST
OBJECTIVE Z
TYPE NLP DIRECTION MINIMIZE
SOLVER MINOS5 FROM LINE 34

**** SOLVER STATUS 1 NORMAL COMPLETION
**** MODEL STATUS 2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE 7.2177

RESOURCE USAGE, LIMIT
0.336 1000.000
ITERATION COUNT, LIMIT
15 1000
EVALUATION ERRORS
0 0

M I N O S 5.3 (Nov 1990) Ver: 225-DOS-02
B. A. Murtagh, University of New South Wales
P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright
Systems Optimization Laboratory, Stanford University.

DEMONSTRATION MODE
You do not have a full license for this program.
The following restrictions apply:
Total nonzero elements: 1000
Nonlinear nonzero elements: 300

Estimate work space needed -- 39 Kb
Work space allocated -- 160 Kb

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED
0 ERRORS

EXECUTION TIME = 0.050 SECONDS VERID TP5-00-038
GAMS 2.25 PC AT/XT 09/16/97
23:48:05 PAGE 4
Test Problem
Solution Report
SOLVE TEST USING NLP FROM LINE 34

EXECUTION TIME = 0.110 SECONDS VERID TP5-00-038
GAMS 2.25 PC AT/XT 09/16/97
23:48:05 PAGE 3
Test Problem
Solution Report
SOLVE TEST USING NLP FROM LINE 34

GAMS DEMONSTRATION VERSION
**** FILE SUMMARY
INPUT C:\GAMS\TEST.GMS
OUTPUT C:\GAMS\TEST.LST
Test Problem

* Example from Problem 8.26 in "Engineering Optimization" by Reklaitis, Ravindran and Ragsdell (1983)

VARIABLES X1, X2, X3, Z;
POSITIVE VARIABLES X1, X2, X3;

EQUATIONS CON1, CON2, CON3, OBJ;

CON1.. X2 - X3 =G= 0;
CON2.. X1 - X3 =G= 0;
CON3.. X1 - X2**2 + X1*X2 - 4 =E= 0;
OBJ.. Z =E= SQR(X1) + SQR(X2) + SQR(X3);

* Upper bounds
X1.UP = 5;
X2.UP = 3;
X3.UP = 3;

* Initial point
X1.L = 4;
X2.L = 2;
X3.L = 2;

MODEL TEST / ALL /

OPTION LIMROW = 0;
OPTION LIMCOL = 0;

SOLVE TEST USING NLP MINIMIZING Z;

The derivative pool refers to the fact that analytical gradients for the nonlinear model have been generated by GAMS.
Test Problem
Solution Report  SOLVE TEST USING NLP FROM LINE 34

SOLVE SUMMARY

MODEL   TEST
TYPE    NLP
SOLVER  MINOS5

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE                7.2177

RESOURCES USAGE, LIMIT          0.336     1000.000
ITERATION COUNT, LIMIT        15         1000
EVALUATION ERRORS              0            0

MINOS 5.3 (Nov 1990)         Ver: 225-DOS-02
B. A. Murtagh, University of New South Wales
and
P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright
Systems Optimization Laboratory, Stanford University.

DEMONSTRATION MODE
You do not have a full license for this program.
The following size restrictions apply:
Total nonzero elements:  1000
Nonlinear nonzero elements:  300

Estimate work space needed --  39 Kb
Work space allocated --  160 Kb

EXIT -- OPTIMAL SOLUTION FOUND
MAJOR ITNS, LIMIT             7     200
FUNOBJ, FUNCON CALLS         34      34
SUPERBASICS  1
INTERPRETER USAGE .00
NORM RG / NORM PI  9.678E-10

LOWER     LEVEL     UPPER    MARGINAL
---- EQU CON1            .        0.916     +INF       .
---- EQU CON2            .        2.626     +INF       .
---- EQU CON3           4.000     4.000     4.000     2.637
---- EQU OBJ             .         .         .        1.000

LOWER     LEVEL     UPPER    MARGINAL
---- VAR X1              .        2.526     5.000      .
---- VAR X2              .        0.916     3.000      EPS
---- VAR X3              .         .        3.000      EPS
---- VAR Z               -INF      7.218     +INF       .

**** REPORT SUMMARY :        0     NONOPT
0 INFEASIBLE
0 UNBOUNDED
0 ERRORS

The column MARGINAL gives the dual variables or multipliers.
Working with GAMS - Basics

- Running Model
- Finding Compilation error
- Finding Feasibility of Model
- Result interpretation
Working with GAMS – Flow control

- Loop statement
- If else statement
- While statement
- For statement
Working with GAMS - Interfacing

• Excel interfacing
  1. Syntax
  2. Reading excel files
  3. Unloading results into excel files

• MATLAB interfacing
  1. Syntax
  2. Reading MATLAB codes
Example 2

• Transportation Problem

1. Problem definition
To allocate ships to transport personnel from different port to a training center.

2. Objective
Ships are allocated such that minimum voyages are made to transport all men.
Sets

- Fundamental building block.
- A set contains elements which may be character or numeric.

1. Sets set_name “Label” /a, b, c/ ; “declaration”
   set_name is alphanumeric starting with a letter. All identifiers in the model follow the same.

2. Set t "time" /1991 * 2000/ ; “sequence”

3. Sets
   s "Sector" / sector1
      sector 2 /
   r "regions" / northern
      eastern/ ;

4. Set sector_region (s,r) “one to one mapping”
Sets

• In the given example

  Sets i ports / Chennai, Surat, Kandla, Mumbai /
  j voyages / v-01* v-15 /
  k ship classes / small, medium, large /
  sc (i, k) ship capability /
    Chennai.(small, medium, large)
    (Surat, Kandla).(medium, large)
    Mumbai.large /
  vc (j, k) voyage capability ;
Scalars and parameters

- Scalar is data type for single data entry.
  1. Scalar scalar_name “label” /20/;
  2. Scalar rate “rate of interest”;
     rate = 0.05;
  In the given example
   Scalar w1 ship assignment weight / 1.00 /
       w2 ship distance traveled weight / .01 /
       w3 personnel distance travel weight / .0001 /;
- Parameter is data type for list oriented data
  1. Set j states /Rajasthan, M.P./;
     Parameters district (j) district in states
       /Rajasthan 32, M.P. 51/;
  Default value of a parameter is zero.
Parameters

2. Parameter
   salaries (employee, manager, department)
     /Kamal .Sundar .toy = 6000
     Kunal .Smita .Furniture = 9000
     Komal .Gaurav .cosmetics = 8000 /

In the given example
Parameter  \( d(i) \) number of men at port \( i \) needing transport
  / Chennai = 475, Surat = 659
  Kandla = 672, Mumbai = 1123 /
shipcap(k) ship capacity in men
  / small = 100
  medium = 200
  large = 600 /
n(k) number of ships of class \( k \) available
  / small 2, medium 3, large 4 /
Table

- Tabular data can be declared and initialized using “Table” statement.
- The table statement is the only statement in the GAMS language that is not free format.
- The character positions of the numeric table entries must overlap the character positions of the column headings.
- The sequence of signed numbers forming a row must be on the same line.
- The column section has to fit on one line.
- A specific column can appear only once in the entire table.
- Only one identifier can be declared and initialized in a table statement.
Table

1. Sets m subjects
   /Maths, Physics, Chemistry, Biology/
   i students
   /Karan, Radha, Milan/;

Table marks (m,i) marks obtained out of 100

<table>
<thead>
<tr>
<th></th>
<th>Karan</th>
<th>Radha</th>
<th>Milan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maths</td>
<td>75</td>
<td>89</td>
<td>91</td>
</tr>
<tr>
<td>Physics</td>
<td>67</td>
<td>72</td>
<td>82</td>
</tr>
<tr>
<td>Chemistry</td>
<td>77</td>
<td>89</td>
<td>90</td>
</tr>
<tr>
<td>Biology</td>
<td>91</td>
<td>90</td>
<td>90</td>
</tr>
</tbody>
</table>
In the given example

<table>
<thead>
<tr>
<th></th>
<th>Chennai</th>
<th>Surat</th>
<th>Kandla</th>
<th>Mumbai</th>
</tr>
</thead>
<tbody>
<tr>
<td>v-01</td>
<td>370</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-02</td>
<td>460</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-03</td>
<td>600</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-04</td>
<td>750</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-05</td>
<td>515</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-06</td>
<td>640</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-07</td>
<td>810</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-08</td>
<td>665</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v-09</td>
<td>665</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-10</td>
<td>800</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v-11</td>
<td>720</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v-12</td>
<td>860</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-13</td>
<td>840</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v-14</td>
<td>865</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v-15</td>
<td>920</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Variables

- Entities whose values become known after model is solved.
- There are five types of variables.
  1. Free (Default)
     No bounds on variable.
  2. Positive Variable
     Only positive values are allowed.
  3. Negative Variable
     Only negative values are allowed.
  4. Binary Variable
     Discrete variables that can take only 0 or 1 value.
  5. Integer Variable
     Discrete variables that can take integer values.
Variables

- **Variable** $a$ is unknown;
- **Positive variable** $u(i,j)$, $i,j$ are elements of sets;
- All default bounds set at declaration time can be changed using assignment statements.
  - $u.\text{up}(i,j) = 1000$;
  - $a.\text{lo} = \text{inf}$;
  - .fx variable suffix is used if lower and upper bounds are equal.
  - $p.\text{fx}('a', 'b', 'c') = 200$;
Variable Attributes

- **.lo** - The lower bound for the variable.
- **.up** - The upper bound for the variable.
- **.fx** - The fixed value for the variable.
- **.l** - The activity level for the variable. This is also equivalent to the current value of the variable.
- **.m** - The marginal value (also called dual value) for the variable.
- **.scale** - This is the scaling factor on the variable. This is normally an issue with nonlinear programming.
- **.prior** - This is the branching priority value of a variable. This parameter is used in mixed integer programming models only.
Variables

- In the given problem

Variables: $z(j,k)$ number of times voyage $jk$ is used

$y(j,k,i)$ number of men transported from port $i$ via voyage $jk$

Objective: $\text{obj}$;

Integer Variables: $z$;

Positive variables: $y$
Equations

• Equations are the GAMS names for the symbolic algebraic relationships that is used to generate the constraints in the model.
• Declaration of equation equations
cost 'total cost definition'
invb(q) 'inventory balance'
sbal(q,s) 'shift employment balance' ;
• Equation Definition
cost ... a =e= b + c ;
Equations

- **=e=** **Equality** - RHS must equal LHS
- **=g=** **Greater than** - LHS must be greater than or equal to RHS.
- **=l=** **Less than** – LHS must be less than or equal to RHS.
- **=n=** - No relationships enforced between LHS and RHS.
- **=x=** **External equation** - Only supported by selected solvers.
- **=c=** **Conic constraint** - Only supported by selected solvers.
Equations

• Scalar equations
  total.. \( c = \sum(i, y(i)) \);

• Indexed equations
  The index sets to the left of the '..' are called the
domain of definition of the equation.
Domain checking ensures that the domain over which
an equation is defined must be the set or a subset of
the set over which the equation is declared.
  cost(t) ... all (t) = g= mango(t) + papaya(t) ;
Dollar Condition

- The dollar operator operates with a logical condition. The term $(condition)$ can be read as 'such that condition is valid' where condition is a logical condition.
- The dollar logical conditions cannot contain variables. Variable attributes (like .l and .m) are permitted however.
- if \((b > 1.5)\), then \(a = 2\) can be written as \(a$(b > 1.5)$ = 2 ;
- \(d(i)$( w(i)$q(i) ) = v(i)\); is a nested dollar condition which means that The assignation will be executed for element \(d(i)\) which simultaneously are elements of \(w\) and \(q\).
- The effect of the dollar condition is significantly different depending on which side of the assignment it is in.
Dollor Condition

• Dollor on right
  1. For an assignment statement with a dollar condition on the RHS, an
     assignment is always made.
  2. For example \( x = 2 \$(y > 1.5) \);
     in words means
     if \((y > 1.5)\) then \((x = 2)\), else \((x = 0)\)
• Dollor on left
  1. For an assignment statement with a dollar condition on the LHS, no
     assignment is made unless the logical condition is satisfied.
  2. For example \( x$( y \neq 0) = (1/y)-2 \);
     in words means
     if \(y = 0\) then no assignment is made.
Conditional Equations

- A dollar operator within an equation is analogous to the dollar control on the right of assignments.
  
- \( mb(i) .. x(i) = g = y(i) + (e(i) - m(i)) \$t(i) \)
  
  means that the term is added to the RHS of the equation only for those elements of \( i \) that belong to \( t(i) \).

- Dollar control over domain definition is analogous to the dollar control on the left of the assignments.

- \( p(j,i,t) \$(\text{ord}(t) \leq 12) .. Ts(j,i,t) = e = Te(j,i,t) \)
  
  \( t \) is set of hours in a day. If order of \( t \) is less than 12 then the assignment is made.
Conditional Equation

• In the given example, equations are declared and defined as
  Equations
  objdef
  demand (i)     pick up all the men at port i
  voycap(j,k)    observe variable capacity of voyage jk
  shiplim(k)     observe limit of class k

  demand(i)..<  Sum ((j,k)$ (a(j,i)$v c(j,k), y(j,k,i)) =g=  d(i) ;
  voycap(j,k)$vc(j,k).  Sum (i$a(j,i), y(j,k,i)) =l=  shipcap(k)*z(j,k) ;
  shiplim(k)..<  Sum (j$vc(j,k), z(j,k)) =l=  n(k) ;
  objdef.<  obj =e=  w1* sum((j,k)$vc(j,k), z(j,k)) +
          w2*sum((j,k)$vc(j,k), a(j,"dist")*z(j,k)) +
          w3*sum((j,k,i)$ (a(j,i)$vc(j,k)), a(j,"dist")*y(j,k,i)) ;
Model Statement

• The model statement is used to collect equations into groups and to label them so that they can be solved.

• Model transport "a transportation model" / all / ;

• The model is called transport and the keyword all is a shorthand for all known (declared) equations.

• In the given example, if model name is navy then

  Model navy /all/ ;
Solve Statement

- There are different types of problems and their identifiers.
- LP Linear programming.
- QCP Quadratic constraint programming.
- NLP Nonlinear programming.
- DNLP Nonlinear programming with discontinuous derivatives.
- RMIP Relaxed mixed integer programming. The integer and binary variables can assume any values between their bounds.
Solve Statement

• MIP Mixed integer programming.
• RMIQCP Relaxed mixed integer quadratic constraint programming.
• RMINLP Relaxed mixed integer nonlinear programming.
• MIQCP Mixed integer quadratic constraint programming.
• MINLP Mixed integer nonlinear programming.
• RMPEC Relaxed Mathematical Programs with Equilibrium Constraints.
• MPEC Mathematical Programs with Equilibrium Constraints.
• MCP Mixed Complementarity Problem.
• CNS Constrained Nonlinear System.
• EMP Extended Mathematical Program.
Solve Statement

• GAMS itself does not solve the problem, but passes the problem definition to one of a number of separate solver programs.
• The objective variable must be scalar and type free, and must appear in the least one of the equations in the model.
• For example
  Solve transport using lp minimizing cost ;
• In the given example
  Solve navy minimizing obj using mip ;
• Display statement is used for displaying the current value of the variables.
• Display y.l, z.l ;
Example in GAMS

Sets
  i  ports / chennai, Surat, kandla, Mumbai /
  j  voyages / v-01* v-15 /
  k  ship classes / small, medium, large /
sc(i,k) ship capability / Chennai. (small, medium, large)
                          (surat, Kandla). (medium, large)
                          Mumbai. large /
vc(j,k) voyage capability :

Parameter
d(i) number of men at port i needing transport /
    chennai = 475, surat = 659
    Kandla = 672, Mumbai = 1123 /
shipcap(k) ship capacity in men / small 100
                          medium 200
                          large 600 /
n(k) number of ships of class k available
    / small 1, medium 3, large 4 /
Table a(j,*) assignments of ports to voyages

<table>
<thead>
<tr>
<th></th>
<th>chennai</th>
<th>Surat</th>
<th>Kandla</th>
<th>Mumbai</th>
</tr>
</thead>
<tbody>
<tr>
<td>v-01</td>
<td>370</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>v-02</td>
<td>460</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-03</td>
<td>600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-04</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-05</td>
<td>515</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>v-06</td>
<td>640</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-07</td>
<td>810</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-08</td>
<td>665</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v-09</td>
<td>665</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-10</td>
<td>800</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-11</td>
<td>720</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>v-12</td>
<td>860</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>v-13</td>
<td>840</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>v-14</td>
<td>865</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-15</td>
<td>920</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Example in GAMS

```plaintext
vc(j,k) = prod(i$a(j,i), sc(i,k)); Display vc;

Variables
  z(j,k) number of times voyage jk is used
  y(j,k,i) number of men transported from port i via voyage jk
  obj;

Integer Variables z; positive variables y;

Equations
  objdef
    demand(i) pick up all the men at port i
    voycap(j,k) observe variable capacity of voyage jk
    shiplim(k) observe limit on class k;

  demand(i).. =g= sum((j,k)$(a(j,i)$vc(j,k)), y(j,k,i));
  voycap(j,k)$vc(j,k).. =l= shipcap(k)*z(j,k);
  shiplim(k).. =l= n(k);

  objdef.. =e= sum((j,k)$(a(j,i)$vc(j,k), z(j,k))
                  + 2*sum((j,k)$vc(j,k), a(j,"dist")*z(j,k))
                  + 3*sum((j,k,i)$(a(j,i)$vc(j,k)), a(j,"dist")*y(j,k,i));

Model navy /all/;

z.up(j,k)$vc(j,k) = n(k);

Solve navy minimizing obj using mip;

Display y.l, z.l;
```
GAMS Outputs

• Echo print of input file
  It is always the first part of the output file. It is just a listing of the input with the lines numbers added.

```
GAMS Rev 238  WEX-WEI 23.8.2 x86_64/MS Windows     05/15/15 11:07:38 Page 1
General Algebraic Modeling System
Compilation

1    Sets i ports / chennai, surat, kandla, mumbai /
2      j voyages / 01* v-15 /
3      k ship classes / small, medium, large /
4      sc(i,k) ship capability / chennai.(small,medium,large)
5           (surat,kandla).(medium,large)
6           (mumbai.large /
7      vc(j,k) voyage capability ;
8
9    Parameter d(i) number of men at port i needing transport
10       / chennai = 475, surat = 659
11         kandla = 672, mumbai = 1123 /
12      shipcap(k) ship capacity in men / small 100
13         medium 200
14         large 600 /
15
16      n(k) number of ships of class k available
17          / small 2, medium 3, large 4 / ;
18  Table a(j,*) assignment of ports to voyages
```
• Equation listing

The equation listing is an extremely useful debugging aid. It shows the variables that appear in each constraint, and what the individual coefficients and RHS value evaluate to after the data manipulations have been done.

```plaintext
---- objdef -E-

objdef..  -  4.7*z(v-01,small)  -  4.7*z(v-01,medium)  -  4.7*z(v-01,large)
   -  5.6*z(v-02,medium)  -  5.6*z(v-02,large)  -  7*z(v-03,medium)
   -  7*z(v-03,large)  -  8.5*z(v-04,large)  -  6.15*z(v-05,medium)
   -  6.15*z(v-05,large)  -  7.4*z(v-06,medium)  -  7.4*z(v-06,large)
   -  9.65*z(v-07,large)  -  7.65*z(v-08,medium)  -  7.65*z(v-08,large)
   -  7.65*z(v-09,large)  -  9*z(v-10,large)  -  8.2*z(v-11,medium)
   -  8.2*z(v-11,large)  -  9.6*z(v-12,large)  -  9.4*z(v-13,large)
   -  9.6*z(v-14,large)  -  10.2*z(v-15,large)  -  0.037*y(v-01,small,chennai)
   -  0.037*y(v-01,medium,chennai)  -  0.037*y(v-01,large,chennai)
```
GAMS Output

- Column Listing
  This is a list of the individual coefficients sorted by column rather than by row.
GAMS Output

- Model Statistics

Its use is to provide details on the size and nonlinearity of the model.

The BLOCK counts refer to GAMS equations and variables, the SINGLE counts to individual rows and columns in the problem generated.

<table>
<thead>
<tr>
<th>MODEL STATISTICS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCKS OF EQUATIONS</td>
<td>4</td>
</tr>
<tr>
<td>BLOCKS OF VARIABLES</td>
<td>3</td>
</tr>
<tr>
<td>NON ZERO ELEMENTS</td>
<td>205</td>
</tr>
<tr>
<td>SINGLE EQUATIONS</td>
<td>31</td>
</tr>
<tr>
<td>SINGLE VARIABLES</td>
<td>69</td>
</tr>
<tr>
<td>DISCRETE VARIABLES</td>
<td>23</td>
</tr>
</tbody>
</table>

GENERATION TIME = 0.046 SECONDS 4 Mb WEX238-238 Apr 3, 2012
GAMS Outputs

• Solve Summary

It gives the summary of model solution such that model type of the model being solved, name of the solver used to solve the model, name of the objective variable, direction of optimization being performed etc.

```
   SOLVE SUMMARY

   MODEL      TYPE       HIP
   SOLVER      CPLEX

   *** SOLVER STATUS    1 Normal Completion
   *** MODEL STATUS     8 Integer Solution
   *** OBJECTIVE VALUE  221.5785

   RESOURCE USAGE, LIMIT       0.187       1000.000
   ITERATION COUNT, LIMIT     21         2000000000
```

GAMS Output

- Solution Listing

It is row-by-row then column-by-column listing of the solutions returned to GAMS by the solver program.

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Chennai</td>
<td>475.000</td>
<td>475.000</td>
<td>+INF</td>
</tr>
<tr>
<td>Surat</td>
<td>659.000</td>
<td>659.000</td>
<td>+INF</td>
</tr>
<tr>
<td>Kandla</td>
<td>672.000</td>
<td>672.000</td>
<td>+INF</td>
</tr>
<tr>
<td>Mumbai</td>
<td>1123.000</td>
<td>1123.000</td>
<td>+INF</td>
</tr>
</tbody>
</table>
• Display

---- 72 VARIABLE y.L number of men transported from port i via voyage jk

<table>
<thead>
<tr>
<th></th>
<th>chennai</th>
<th>Surat</th>
<th>kandla</th>
<th>Mumbai</th>
</tr>
</thead>
<tbody>
<tr>
<td>v-01.medium</td>
<td>400.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-03.large</td>
<td></td>
<td></td>
<td></td>
<td>600.000</td>
</tr>
<tr>
<td>v-05.medium</td>
<td>75.000</td>
<td>125.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-08.large</td>
<td>457.000</td>
<td>72.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-09.large</td>
<td>72.000</td>
<td></td>
<td></td>
<td>1123.000</td>
</tr>
</tbody>
</table>

---- 72 VARIABLE number of times voyage jk is used

<table>
<thead>
<tr>
<th></th>
<th>medium</th>
<th>large</th>
</tr>
</thead>
<tbody>
<tr>
<td>v-01</td>
<td>2.000</td>
<td></td>
</tr>
<tr>
<td>v-03</td>
<td></td>
<td>1.000</td>
</tr>
<tr>
<td>v-05</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>v-08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-09</td>
<td></td>
<td>2.000</td>
</tr>
</tbody>
</table>
Dollar Control Option

- The Dollar Control Options are used to indicated compiler directives and options.
- Dollar control options are not part of the GAMS language so they do not appear on the compiler listing unless an error had been detected.
- For example $call 'dir'
  This command makes a directory listing on a PC.
Installation of GAMS

- Installation notes available for Windows, Mac, and UNIX
- Manuals are also available online
- Without a valid GAMS license the system will operate as a free demo system:
  - Model limits:
    - Number of constraints and variables: 300
    - Number of nonzero elements: 2000 (of which 1000 nonlinear)
    - Number of discrete variables: 50 (including semi continuous, semi integer and member of SOS-Sets)
  - Additional Global solver limits: Number of constraints and variables: 10
- The GAMS log will indicate that your system runs in demo mode:
  GAMS 24.3.3 Copyright (C) 1987-2014 GAMS Development. All rights reserved Licensee:
  GAMS Development Corporation, Washington, DC G871201/0000CA-ANY Free Demo, 202-
  342-0180, sales@gams.com, www.gams.com DC0000
- GAMS will terminate with a licensing error if you hit one of the limits above:
  *** Status: Terminated due to a licensing error *** Inspect listing file for more information
- http://www.gams.com/download/
References

1. A GAMS tutorial
   www.gams.com/dd/docs/gams/Tutorial.pdf

2. GAMS – A USER’s GUIDE
   pages.cs.wisc.edu/~swright/635/docs/GAMSUsersGuide.pdf
OPTIMIZATION OF CHEMICAL PROCESSES
WHAT OPTIMIZATION IS ALL ABOUT

• OPTIMIZATION is the use of specific methods to determine the most cost-effective and efficient solution to a problem or design for a process.

• This technique is one of the major quantitative tools in industrial decision making.

• A wide variety of problems in the design, construction, operation, and analysis of chemical plants (as well as many other industrial processes) can be resolved by optimization.
WHY OPTIMIZE?

- To improve the initial design of equipment
- To enhance the operation of that equipment
- To realize the largest production
- The greatest profit
- The least energy usage and so on.
TYPES OF OPTIMIZATION PROBLEMS

LINEAR PROBLEMS

Can be solved by:
- Graphical method
- Simplex
- Dual simplex
- Two phase
- BIG-M etc.

NON-LINEAR PROBLEMS

PROBLEMS WITHOUT CONSTRAINTS

Can be solved by:
- Newton’s method
- Steepest decent
- Conjugate gradient etc.

PROBLEMS WITH CONSTRAINTS

Can be solved by:
- Langrangian method
- Penalty and barrier method
- GRG method etc.
NON LINEAR PROGRAMMING
Conjugate Gradient (Fletcher-Reeves) Method
Fletcher-Reeves Method

The iterative procedure of Fletcher-Reeves method can be stated as follows:

1. Start with an arbitrary initial point $X_1$.
2. Set the first search direction $S_1 = -\nabla f(X_1) = -\nabla f_i$
3. Find the point $X_2$ according to the relation

$$X_2 = X_1 + \lambda_i^* S_i$$

where $\lambda_i^*$ is the optimal step length in the direction $S_i$. Set $i=2$ and go to the next step.

4. Find $\nabla f_i = \nabla f(X_i)$, and set

$$S_i = -\nabla f_i + \frac{|\nabla f_i|}{|\nabla f_{i-1}|^2} S_{i-1}$$

5. Compute the optimum step length $\lambda_i^*$ in the direction $S_i$, and find the new point

$$X_i = X_{i-1} + \lambda_i^* S_i$$

6. Test for the optimality of the point $X_{i+1}$. If $X_{i+1}$ is optimum, stop the process. Otherwise set the value of $i = i + 1$ and go to step 4.
Example

Minimize

$$f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

starting from the point

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Solution:

Iteration 1

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 + 4x_1 + 2x_2 \\ 1 + 2x_1 + 2x_2 \end{bmatrix}$$

$$\nabla f_1 = \nabla f(x_1) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$s_1 = -\nabla f_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The search direction is taken as:
Example

- To find the optimal step length $\lambda_1^*$ along $S_1$, we minimize $f(X_1 + \lambda_1 S_1)$ with respect to $\lambda_1$. Here

$$f(X_1 + \lambda_1 S_1) = f(-\lambda_1, -\lambda_1) = \lambda_1^2 - 2\lambda_1$$

$$\frac{df}{d\lambda_1} = 0 \Rightarrow \lambda_1^* = 1$$

- Therefore

$$X_2 = X_1 + \lambda_1 S_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 1 \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$
Example

Iteration 2: Since

\[ \nabla f_2 = \nabla f(x_2) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \]

the equation

\[ S_i = -\nabla f_i + \frac{|\nabla f_i|^2}{|\nabla f_{i-1}|^2} S_{i-1} \]

gives the next search direction as

\[ S_1 = -\nabla f_2 + \frac{|\nabla f_2|^2}{|\nabla f_1|^2} S_1 \]

where

\[ |\nabla f_1|^2 = 2 \quad \text{and} \quad |\nabla f_2|^2 = 2 \]

Therefore

\[ S_2 = -\begin{bmatrix} -1 \\ -1 \end{bmatrix} + \frac{2}{2} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ +2 \end{bmatrix} \]
Example

- To find $\lambda_2^*$, we minimize

$$f(X_2 + \lambda_2 S_2) = f(-1, 1 + 2\lambda_2)$$

$$= -1 - (1 + 2\lambda_2) + \frac{2(1 + 2\lambda_2) + (1 + 2\lambda_2)^2}{4\lambda_2^2 - 2\lambda_2 - 1}$$

with respect to $\lambda_2$.

Thus the optimum point is reached in two iterations. Even if we do not know this point to be optimum, we will not be able to move from this point in the next iteration. This can be verified as follows:

$$X_3 = X_2 + \lambda_2^* S_2 = \left\{\begin{array}{c}
-1 \\
1 \\
0 \\
4 \\
2 \\
-1 \\
1.5
\end{array}\right\}$$
Example

Iteration 3:

Now

\[ \nabla f_3 = \nabla f(X_3) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad |\nabla f_2|^2 = 2, \quad \text{and} \quad |\nabla f_3|^2 = 0. \]

Thus,

\[ s_3 = -\nabla f_3 + (|\nabla f_3|^2/|\nabla f_2|^2)s_2 = -\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \left( \frac{0}{2} \right) \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \]

This shows that there is no search direction to reduce \( f \) further, and hence \( X_3 \) is optimum.
Hooke and Jeeves’ Method
Hooke and Jeeves’ Method

- The pattern search method of Hooke and Jeeves is a sequential technique each step of which consists of two kinds of moves, the exploratory move and the pattern move.

- The first kind of move is included to explore the local behaviour of the objective function and the second kind of move is included to take advantage of the pattern direction.

- The general procedure can be described by the following steps:

1. Start with an arbitrarily chosen point

   \[ X_1 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \]

   called the starting base point, and prescribed step lengths \( \Delta x_i \) in each of the coordinate directions \( u_i, i=1,2,\ldots,n \). Set \( k=1 \).
Hooke and Jeeve’s method

2. Compute \( f_k' = f(X_k) \). Set \( i = 1, \ Y_{k_0} = X_k \), where the point \( Y_{k_0} \) indicates the temporary base point \( X_k \) by perturbing the \( j \)th component of \( X_k \). Then start the exploratory move as stated in Step 3.

3. The variable \( x_i \) is perturbed about the current temporary base point \( Y_{k,i-1} \) to obtain the new temporary base point as

\[
Y_{k,i} = \begin{cases} 
Y_{k,i-1} + \Delta x_i u_i & \text{if } f^+ = f(Y_{k,i-1} + \Delta x_i u_i) < f = f(Y_{k,i-1}) \\
Y_{k,i-1} - \Delta x_i u_i & \text{if } f^- = f(Y_{k,i-1} - \Delta x_i u_i) < f = f(Y_{k,i-1}) < f^+ = f(Y_{k,i-1} + \Delta x_i u_i) \\
Y_{k,i-1} & \text{if } f = f(Y_{k,i-1}) < \min(f^+, f^-)
\end{cases}
\]

This process of finding the new temporary base point is continued for \( i = 1, 2, \ldots \) until \( x_n \) is perturbed to find \( Y_{k,n} \).
Hooke and Jeeves’ Method

4. If the point \( Y_{k,n} \) remains the same as \( X_k \), reduce the step lengths \( \Delta x_i \) (say, by a factor of 2), set \( i = 1 \) and go to step 3. If \( Y_{k,n} \) is different from \( X_k \), obtain the new base point as

\[
X_{k+1} = Y_{k,n}
\]

and go to step 5.

5. With the help of the base points \( X_k \) and \( X_{k+I} \), establish a pattern direction \( S \) as

\[
S = X_{k+I} - X_k
\]

and find a point \( Y_{k+1,0} \) as

\[
Y_{k+1,0} = X_{k+1} + \lambda S
\]

where \( \lambda \) is the step length, which can be taken as 1 for simplicity. Alternatively, we can solve a one-dimensional minimization problem in the direction \( S \) and use the optimum step length \( \lambda^* \) in place of \( \lambda \) in the equation

\[
Y_{k+1,0} = X_{k+1} + \lambda S
\]
Hooke and Jeeves’ Method

6. Set $k = k + 1$, $f_k = f(Y_{k0})$, $i = 1$, and repeat step 3. If at the end of step 3, $f(Y_{k,n}) < f(X_k)$, we take the new base point $X_{k+1} = Y_{k,n}$ and go to step 5. On the other hand, if $f(Y_{k,n}) \geq f(X_k)$, set $X_{k+1} = X_k$, reduce the step lengths $\Delta x_i$, set $k = k + 1$, and go to step 2.

7. The process is assumed to have converged whenever the step lengths fall below a small quantity $\varepsilon$. Thus the process is terminated if

$$\max_i(\Delta x_i) < \varepsilon$$
Example

Minimize

\[ f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \]

starting from the point

\[ x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

Take \( \Delta x_1 = \Delta x_2 = 0.8 \) and \( \varepsilon = 0.1 \).

Solution:

Step 1: We take the starting base point as

\[ x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

and step lengths as \( \Delta x_1 = \Delta x_2 = 0.8 \) along the coordinate directions \( u_1 \) and \( u_2 \), respectively. Set \( k=1 \).
Example

Step 2: \( f_1 = f(X_1) = 0, i=1, \) and

\[
Y_{10} = X_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

Step 3: To find the new temporary base point, we set \( i = 1 \) and evaluate \( f = f(Y_{10}) = 0.0 \)

\[
f^+ = f(Y_{10} + \Delta x_1 u_1) = f(0.8, 0.0) = 2.08
\]

\[
f^- = f(Y_{10} - \Delta x_1 u_1) = f(-0.8, 0.0) = 0.48
\]

Since \( f < \min(f^+, f^-) \), we take \( Y_{11} = X_1 \). Next we set \( i = 2, \) and evaluate

\[
f = f(Y_{11}) = 0.0 \text{ and }
\]

\[
f^+ = f(Y_{11} + \Delta x_2 u_2) = f(0.0, 0.8) = -0.16
\]

Since \( f^+ < f \), we set

\[
Y_{12} = \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix}
\]

*** \( Y_{kj} \) indicates the temporary base point \( X_k \) by perturbing the \( j \)th component of \( X_k \)
**Example**

**Step 4:** As $Y_{12}$ is different from $X_1$, the new base point is taken as:

$$X_2 = \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix}$$

**Step 5:** A pattern direction is established as:

$$S = X_2 - X_1 = \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix} - \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix}$$

The optimal step length $\lambda^*$ is found by minimizing

$$f(X_2 + \lambda S) = f(0.0, 0.8 + 0.8\lambda) = 0.64\lambda^2 + 0.48\lambda - 0.16$$

As $df/d\lambda = 1.28 \lambda + 0.48 = 0$ at $\lambda^* = -0.375$, we obtain the point $Y_{20}$ as

$$Y_{20} = X_2 + \lambda^* S = \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix} - 0.375 \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.8 \end{bmatrix} - \begin{bmatrix} 0.0 \\ 0.5 \end{bmatrix}$$
Example

Step 6: Set $k = 2$, $f = f_2 = f(Y_{20}) = -0.25$, and repeat step 3. Thus, with $i = 1$, we evaluate

$$f^+ = f(Y_{20} + \Delta x_1 u_1) = f(0.8,0.5) = 2.63$$

$$f^- = f(Y_{20} - \Delta x_1 u_1) = f(-0.8,0.5) = -0.57$$

Since $f^- < f < f^+$, we take

$$Y_{21} = \begin{bmatrix} -0.8 \\ 0.5 \end{bmatrix}$$

Next, we set $i = 2$ and evaluate

$$f^+ = f(Y_{21} + \Delta x_2 u_2) = f(-0.8,1.3) = -1.21$$

$$Y_{22} = \begin{bmatrix} -0.8 \\ 1.3 \end{bmatrix}$$

As $f^+ < f$, we take

Since $f(Y_{22}) = -1.21 < f(X_2) = -0.25$, we take the new base point as:

$$X_{23} = \begin{bmatrix} 0.8 \\ 1.3 \end{bmatrix}$$
Step 6 continued: After selection of the new base point, we go to step 5.

This procedure has to be continued until the optimum point is found.

\[
X_{\text{opt}} = \begin{bmatrix} -1.0 \\ 4.5 \end{bmatrix}
\]
Newton’s method for multivariable
Newton’s method for multivariable

Let the function in \( f(x) = f(x_1,x_2,...............x_n) \)

Local approximation: 2nd order Taylor series:

\[
 f(x + \Delta x) \approx f(x) + (\nabla f)^T \Delta x + \frac{1}{2} \Delta x^T \mathbf{H} \Delta x = \tilde{f}(\Delta x)
\]

First order sufficiency condition for the function is the derivative of function with respect to \( \Delta X \) should be zero applied to approximation:

\[
 \nabla \tilde{f}(\Delta x) = 0 \quad \Rightarrow \quad \nabla f + \mathbf{H} \Delta x = \mathbf{0}, \quad \Rightarrow \quad \Delta x = -\mathbf{H}^{-1} \nabla f
\]

\( \mathbf{H} \) is Hessain matrix.
• Step: \[ s = -H^{-1}\nabla f \]

\[ \text{Evaluated at } x \]

\[ S \text{ is search direction.} \]

• Update: \[ x_{k+1} = x_k + s_k \]

\[ X_k \text{ is } k\text{th iteration} \]

Initially \( k \) will be 0.

• Newton’s method has quadratic convergence (best!) close to optimum:

\[ \frac{||x_{k+1} - x^*||^2}{||x_k - x^*||^2} \leq C \]

\[ f(x, y, z) \]

\[ \nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \]

\[ H = \begin{pmatrix}
\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\
\frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\
\frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2}
\end{pmatrix} \]
Newton’s Method Example

For an example, we will use the same problem as before:

Minimize $f(x_1, x_2, x_3) = (x_1)^2 + x_1(1 - x_2) + (x_2)^2$

$- x_2x_3 + (x_3)^2 + x_3$

$\nabla f(x) = \begin{bmatrix}
-2x_1 - x_2 + 1 \\
-x_1 + 2x_2 - x_3 \\
-x_2 + 2x_3 + 1
\end{bmatrix}$
Newton’s Method Example

The Hessian is:

\[
H = \begin{pmatrix}
\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\
\frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\
\frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2}
\end{pmatrix}
\]

\[
H(x) = \nabla^2 f(x) = \begin{bmatrix}
2 & -1 & 0 \\
-1 & 2 & -1 \\
0 & -1 & 2
\end{bmatrix}
\]

And we will need the inverse of the Hessian:

\[
[H(x)]^{-1} = \begin{bmatrix}
2 & -1 & 0 \\
-1 & 2 & -1 \\
0 & -1 & 2
\end{bmatrix}^{-1} = \begin{bmatrix}
\frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\
\frac{1}{2} & 1 & \frac{1}{2} \\
\frac{1}{4} & \frac{1}{2} & \frac{3}{4}
\end{bmatrix}
\]
• Using the formula,

\[ H^{-1} = \frac{1}{|H|} \quad \text{(matrix of co-factors)}^T \]

\[ = \frac{1}{4} \quad \text{(matrix of co-factors)}^T \]

\[
\begin{pmatrix}
\frac{1}{4} & 3 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1
\end{pmatrix}^T
\]
Newton’s Method Example

So, pick

\[ \mathbf{x}^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \]

Calculate the gradient for the 1\(^{st}\) iteration:

\[ \nabla f(\mathbf{x}^0) = \begin{bmatrix} 0 - 0 - 1 \\ -0 + 0 - 0 \\ 0 + 0 + 1 \end{bmatrix} \]

\[ \Rightarrow \nabla f(\mathbf{x}^0) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \]
Newton’s Method Example

So, the new \( x \) is:

\[
x^1 = x^0 - \left[ \nabla^2 f(x^0) \right]^{-1} \cdot \nabla f(x^0)
\]

\[
= \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix} - \begin{bmatrix}
\frac{3}{4} & 1 & 1 \\
1 & 2 & 1 \\
\frac{1}{4} & \frac{1}{2} & 3
\end{bmatrix} \cdot \begin{bmatrix}
1 \\
0 \\
1
\end{bmatrix}
\]

\[
\therefore x^1 = \begin{bmatrix}
-1 \\
-1 \\
-1
\end{bmatrix}
\]
Newton’s Method Example

Now calculate the new gradient:

\[
\nabla f(x^1) = \begin{bmatrix}
-2 + 1 + 1 \\
1 - 2 + 1 \\
1 - 2 + 1
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\]

Since the gradient is zero, the method has converged.
• It uses the 2\textsuperscript{nd} derivative, Newton’s Method models quadratic functions exactly and can find the optimum point in one iteration.

• If the function had been a higher order, the Hessian would not have been constant and it would have been much more work to calculate the Hessian and take the inverse for each iteration.
PENALTY AND BARRIER METHOD
Two Classes of Sequential Methods

1) First class uses a sequence of infeasible points and feasibility is obtained only at the optimum. These are referred to as penalty function or exterior-point penalty function methods.

2) Second class is characterized by the property of preserving feasibility at all times. These are referred to as barrier function methods or interior-point penalty function methods.
General classical constrained minimization problem

\[ \min \ f(x) \]

subject to

\[ g(x) < 0 \quad j = 1,2,\ldots,n \]
\[ h(x) = 0 \quad I = 1,2,\ldots,m \]

This is achieved by either

- adding a **penalty** for infeasibility and forcing the solution to feasibility
  and subsequent optimum, or

- adding a **barrier** to ensure that a feasible solution never becomes
  infeasible
Penalty method

- Consider the following frequently used general purpose penalty function:
  \[ P(x,r) = f(x) + r \left[ \sum_{i=1}^{m} h_i^2(x) + \sum_{j=1}^{n} \max\{0, g_j(x)\}^2 \right] \]
- Choose \( P(x) \) and a sequence of \( r \) such that when \( r \) goes to infinity, the solution \( x^* \) is found.
- For example, for \( r = 1 \) and we solve the problem. Then, for the second iteration \( r \) is increased by a factor 10 and the problem is resolved starting from the previous solution.
- Note that an increasing value of \( r \) will increase the effect of the penalties on \( P(x,r) \).
- The process is terminated when no improvement in \( f(x) \) is found and all constraints are satisfied.
Example:

\[
\begin{align*}
\min & \quad \frac{1}{2}[(x_1 - 3)^2 + (x_2 - 2)^2] \\
\text{s.t.} & \quad -x_1 + x_2 \leq 0 \\
& \quad x_1 + x_2 \leq 1 \\
& \quad -x_2 \leq 0
\end{align*}
\]

\[x^* = (1, 0)^T\]
$X^0 = (3,2)$ as initial guess, so the second constraint does not satisfy the inequality, only that constraint is the active one only.

From the general penalty function, we get

\[
P(x,r) = \frac{1}{2}[(x_1 - 3)^2 + (x_2 - 2)^2] + r[(x_1 + x_2 - 1)^2]
\]

Differentiate $P(x,r)$ w.r.t the $x_1$ and $x_2$ & solve them for calculating relation between $x_1$, $x_2$ and $r$.

We get, $x_1 - 1 = x_2$

\[
x_1 = \frac{3 - 2r}{1 + 4r}
\]
<table>
<thead>
<tr>
<th>r</th>
<th>x1</th>
<th>x2</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>-0.415</td>
<td>-1.415</td>
<td>6.832</td>
</tr>
<tr>
<td>100</td>
<td>-0.491</td>
<td>-1.491</td>
<td>9.241</td>
</tr>
<tr>
<td>1000</td>
<td>-0.499</td>
<td>-1.499</td>
<td>12.243</td>
</tr>
<tr>
<td>10000</td>
<td>-0.5</td>
<td>-1.5</td>
<td>12.250</td>
</tr>
<tr>
<td>100000</td>
<td>-0.5</td>
<td>-1.5</td>
<td>12.250</td>
</tr>
</tbody>
</table>

- At the value of \( r = 100000 \), we can stop our calculation and the optimal solution is \( x^* = (-0.5, -1.5)^T \) with objective function value \( f(x^*) = 12.250 \)
Barrier method

• Barrier functions are similar to penalty functions and the algorithm for barrier functions is essentially the same as that for penalty functions.

    Introduce the barrier function

• \( B(x,r) = f(x) - r \{ \ln g_i(x) \} \)

where \( B(x,r) \) is the barrier function and \( r \) is the penalty factor which is supposed to go to zero

• The process is terminated when no improvement in \( f(x) \) is found and all constraints are satisfied.
Example \[ f(x) = x_1^2 + x_2^2 \]

\[ s.t \quad -x_1 - x_2^2 + 2 \leq 0 \]

\[ B(x,r) = x_1^2 + x_2^2 + r[\ln (-x_1 - x_2^2 + 2)] \]

Differentiate B(x,r) w.r.t the $x_1$ and $x_2$ & solve them for calculating relation between $x_1$, $x_2$ and $r$.

We get, \[ x_1 = .5 \]

\[ x_2 = \pm \sqrt{\frac{3}{r} + r} \]

Here we neglecting the negative values of $x_2$. 
<table>
<thead>
<tr>
<th>r</th>
<th>x1</th>
<th>x2</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>.5</td>
<td>31.646</td>
<td>1001.719</td>
</tr>
<tr>
<td>100</td>
<td>.5</td>
<td>10.075</td>
<td>101.756</td>
</tr>
<tr>
<td>10</td>
<td>.5</td>
<td>3.391</td>
<td>11.749</td>
</tr>
<tr>
<td>0</td>
<td>.5</td>
<td>1.225</td>
<td>1.752</td>
</tr>
<tr>
<td>.001</td>
<td>.5</td>
<td>1.225</td>
<td>1.752</td>
</tr>
</tbody>
</table>

The optimal solution is $x^* = (0.5, 1.225)^T$ with objective function value $f(x^*) = 1.75$. 
• Some prefer barrier methods because even if they do not converge, you will still have a feasible solution but barrier functions can only be used when all constraints are inequality constraints. This is because we need the feasible solutions to be interior points.

• Others prefer penalty function methods because you are less likely to be stuck in a feasible pocket with a local minimum. Penalty methods are more robust because in practice you may often have an infeasible starting point. However, penalty functions typically require more function evaluations.
Generalized Reduced Gradient Method

One of popular search methods for optimizing constrained functions is the Generalized Reduced Gradient method (GRG).

Cost Function: \( y = y(x_1, x_2, \ldots, x_n) \)

Constraints: \( \varphi_1(x_1, x_2, \ldots, x_n) = 0 \)
\( \varphi_2(x_1, x_2, \ldots, x_n) = 0 \)
\( \ldots \)
\( \varphi_m(x_1, x_2, \ldots, x_n) = 0 \)

\( n \): Number of variables
\( n-m \): Number of decision Variables
\( m \): Number of constraints
Assumed: \( n=4 \), \( m=2 \)

We want to optimize cost function

\[
y = y(x_1, x_2, x_3, x_4)
\]

Subject to

\[
\varphi_1(x_1, x_2, x_3, x_4) = 0
\]
\[
\varphi_2(x_1, x_2, x_3, x_4) = 0
\]

Choose \( x_3 \) and \( x_4 \) two decision variables.
\[
\Delta \varphi_1 = \frac{\partial \varphi_1}{\partial x_1} \Delta x_1 + \frac{\partial \varphi_1}{\partial x_2} \Delta x_2 + \frac{\partial \varphi_1}{\partial x_3} \Delta x_3 + \frac{\partial \varphi_1}{\partial x_4} \Delta x_4 = 0
\]

\[
\Delta \varphi_2 = \frac{\partial \varphi_2}{\partial x_1} \Delta x_1 + \frac{\partial \varphi_2}{\partial x_2} \Delta x_2 + \frac{\partial \varphi_2}{\partial x_3} \Delta x_3 + \frac{\partial \varphi_2}{\partial x_4} \Delta x_4 = 0
\]

Where \( \varphi'_{ij} = \frac{\partial \varphi_i}{\partial x_j} \), Therefore

\[
\begin{bmatrix}
\varphi'_{11} & \varphi'_{12} \\
\varphi'_{21} & \varphi'_{22}
\end{bmatrix}_{m \times m}
\begin{bmatrix}
\Delta x_1 \\
\Delta x_2
\end{bmatrix}_{m \times 1} = -\begin{bmatrix}
\varphi'_{13} & \varphi'_{14} \\
\varphi'_{23} & \varphi'_{24}
\end{bmatrix}_{m \times (n-m)}
\begin{bmatrix}
\Delta x_3 \\
\Delta x_4
\end{bmatrix}_{(n-m) \times 1}
\]

Thus

\[
\begin{bmatrix}
\Delta x_1 \\
\Delta x_2
\end{bmatrix} = -J^{-1}_{m \times m}
\begin{bmatrix}
\varphi'_{13} & \varphi'_{14} \\
\varphi'_{23} & \varphi'_{24}
\end{bmatrix}
\begin{bmatrix}
\Delta x_3 \\
\Delta x_4
\end{bmatrix} = -J^{-1}_{m \times m}(k_3 \Delta x_3 + k_4 \Delta x_4)
\]
\[ y_{new} = y_{old} + \sum_{i=1}^{n} \frac{\partial y}{\partial x_i} \bigg|_{x_{old}} \Delta x_i \]

\[ y_{new} = y_{old} + \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} y'_3 \\ y'_4 \end{bmatrix} \begin{bmatrix} \Delta x_3 \\ \Delta x_4 \end{bmatrix} \]

With substituting for \( x \) and \( y \) from previous part, we have

\[ y_{new} = y_{old} + \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} \begin{bmatrix} - J^{-1} \end{bmatrix} k_3 + y'_3 \} \Delta x_3 \\
+ \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} \begin{bmatrix} - J^{-1} \end{bmatrix} k_4 + y'_4 \} \Delta x_4 \]
Therefore we determine **Generalized Reduced Gradient**.

\[
GRG_i = \begin{bmatrix} y'_1 & \ldots & y'_m \end{bmatrix}_{1 \times m} \left( - J^{-1}_{m \times m} \right) k_{i \times 1} + y'_i
\]

for

\[
i = n - m, \ldots, n
\]

\[
y_{new} = y_{old} + \sum_{i=n}^{n} GRG_i \Delta x_i
\]

If \( GRG_i > 0 \) for minimization, choose \( \Delta x_i < 0 \)

If \( GRG_i < 0 \) for minimization, choose \( \Delta x_i > 0 \)
Generalized Reduced Gradient Algorithm

Choose decision variables \((n-m)\) and their step sizes \(\Delta x_i\)

Initialize decision variables

Move towards constraints

Calculate \(GRG_i\)

\(i = i + 1\)

\(GRG_i < 0\)

\(\Delta x_i > 0\)

\(\sum_{i=n-m}^{y_{i+1}} = x_i + \Delta x_i\)

Move towards constraints

\(GRG_i > 0\)

\(\Delta x_i < 0\)

\(\sum_{i=n-m}^{y_{i+1}} = y_i + \sum_{i=n-m}^{GRG_i,\Delta x_i}\)

Decreased \(\Delta x_i\)

\(GRG_i > \xi\)

\(y_{i+1} > y_i\)

\(\sum_{i=n-m}^{y_{i+1}} = y_i + \sum_{i=n-m}^{GRG_i,\Delta x_i}\)

\(GRG_i < \xi\)

\(x_{Optimum} = x_{i+1}\)
Example

Initial step size $\Delta x_1 = 0.1$ 

Last step size $\Delta x_{55} = 9.5367 \times 10^{-8}$
Consider the following problem:

Minimize: \(2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2\)

Subject to: 
- \(x_1 + x_2 + x_3 = 2\)
- \(x_1 + 5x_2 + x_4 = 5\)
- \(x_1, x_2, x_3, x_4 \geq 0\)

We set our starting point as: 
- \(x_1 = (0, 0, 2, 5)^T\)

\(\nabla f(x) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6, 0, 0)^T\)
Step 1:
At $x_1 = (0,0,2,5)^T$, 

$$\nabla f(x_1) = (-4, -6, 0, 0)^T$$

$I_1 = \{3,4\}$ so that $B = [a_3,a_4]$ and $N = [a_1,a_2]$

$I_\kappa = \text{Index set of the } m \text{ largest component of } x_\kappa \hspace{1cm} \text{(1)}$

$$B = \{a_j: j \in I_\kappa\}, \hspace{0.5cm} N = \{a_i: j \notin I_\kappa\}, \hspace{1cm} \text{ (2)}$$

Then we know that: $\nabla f_B(x_1) = (0,0)^T$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix}$$

Again, 

$$r_1 = \nabla f(x)^T - \nabla f_B(x)^TB^TA \hspace{1cm} \text{(3)}$$

$$= (-4,-6,0,0) - (0,0) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix}$$
Again, \( d_j = -r_j \) if \( j \notin \mathcal{K} \) and \( r_j \leq 0 \)

\[
= -x_j r_j, \text{if } j \notin \mathcal{K} \text{ and } r_j > 0 \ ........(4)
\]

\[d_N = (d_1, d_2)^T = (4, 6)^T\]

Also,

\[d_B = -B^1 N d_N ........(5)\]

\[
= -\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix} [4 \ 6]^T
\]

\[= (-10, -34)^T\]

Thus, \( d_I = (d_N, d_B) = (4, 6, -10, -34)^T \)
Step 2:
\[ x_j = (0,0,2,5)^T \text{ and } \]
\[ d_j = (4,6,-10,-34)^T \]

\[ \lambda_{\text{max}} = \min -\frac{x_{jk}}{d_{jk}} \text{ if } d_k < 0 \]
\[ = \infty \quad \text{ if } d_k \geq 0 \quad \ldots \ldots \quad (6) \]

\[ \lambda_{\text{max}} = \{-\frac{x_{31}}{d_{31}}, -\frac{x_{41}}{d_{41}}\} \]
\[ = \min \{2/10, 5/34\} \]
\[ = 5/34 \]

Then we solve the following problem:

Minimize: \[ f(x_1 + \lambda_1 d_1) = 56 \lambda_1^2 - 52 \lambda_1 \]
Such that: \[ 0 \leq \lambda_1 \leq 5/34 \]

Clearly, \[ \lambda_1 = 5/34 \]
\[ x_2 = x_1 + \lambda_1 d_1 = (10/17, 15/17, 9/17, 0)^T \]
Iteration 2

Step 1:

At the new starting point $x_2 = (10/17, 15/17, 9/17, 0)^T$,

we have $\nabla f(x_2) = (-58/17, -62/17, 0, 0)^T$

$l_1 = \{1, 2\}$ so that $B = [a_1, a_2]$ and $N = [a_3, a_4]$

Then we know that $\nabla f_5(x_2) = (-58/17, -62/17)^T$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix}$$

Again,

$$f_t = \nabla f(x)^T - \nabla_B f(x)^T B^T A$$

$$= (-58/17, -62/17, 0, 0) - (-58/17, -62/17) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix}$$

$$= (0, 0, 57/17, 1/17)$$
\[ d_N = (d_3, d_4)^T \]
\[ = (-(x_{32})(r_{32}), -(x_{42})(r_{42}))^T \]
\[ = (-(9/17)(57/17), -(0)(1/17))^T \]
\[ = (-513/289, 0)^T \]

Also,

\[ d_B = -B^T N d_N \]
\[ = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -513/289 \\ 0 \end{bmatrix}^T \]
\[ = \begin{bmatrix} 2565/1156 \\ -513/1156 \end{bmatrix}^T \]

Thus, \[ d_2 = (d_B, d_N) \]
\[ = (2565/1156, -513/1156, -513/289, 0)^T \]
Step 2:

\[ x_2 = \left( \frac{10}{17}, 15/17, 9/17, 0 \right) \text{T} \quad \text{and} \]

\[ d_2 = \left( \frac{2565}{1156}, -\frac{513}{1156}, -\frac{513}{289}, 0 \right) \text{T} \]

\[ \lambda_{\text{max}} = \left\{ -\frac{x_{22}}{d_{22}}, -\frac{x_{32}}{d_{32}} \right\} \]

\[ = \min \left\{ -(15/17)/-(513/1156), -(9/17)/-(513/289) \right\} \]

\[ = 17/57 \]

*Then we solve the following problem:*

Minimize: \( f(x_2 + \lambda d_2) = 12.21 \lambda_2 - 5.95 \lambda_2 - 6.436 \)

Such that: \( 0 \leq \lambda_2 \leq 17/57 \)
Clearly,\n\[ \lambda_2 = \frac{68}{279} \] that minimizes the function\n\[ f(x_2 + \lambda_2 d_2) = 12.21 \lambda_2^2 - 5.95 \lambda_2 - 6.436 \]

Thus,\n\[ x_3 = x_2 + \lambda_2 d_2 = (\frac{35}{31}, \frac{24}{31}, \frac{3}{31}, 0)^T \]
Iteration 3

Step 1:

\[ x_3 = (35/31, 24/31, 3/31, 0)^T, \]

we have, \( \nabla f(x_3) = (-32/31, -160/31, 0, 0)^T \)

\( I_3 = \{1, 2\} \) so that \( B = [a_1, a_2] \) and \( N = [a_3, a_4] \)

Again,

\[ r_t = \nabla f(x)^T - \nabla_B f(x)^T B^{-1} A \]

\[ = (-32/31, -160/31, 0, 0) - (-32/31, -160/31) \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix} \]

\[ = (0, 0, 0, 32/31) \]
\[ d_N = (d_3, d_4)^T \\
= (-x_{33})(r_{33}), -(x_{43})(r_{43}))^T \\
= - (3/31)(0), -(0)(32/31))^T \\
= (0, 0)^T \\
\]

Also,

\[ d_B = -B^{-1}Nd_N = - \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}^T \\
= [0 \\ 0]^T \\
\]

\[ d = (d_B, d_N) = (0, 0, 0, 0)^T \\
\]

Thus,

\[ x_3 = (35/31, 24/31, 3/31, 0)^T \text{ is a KKT (Optimal solution) for this problem.} \]

Where, KKT means Karush-Kuhn-Tucker Point
A linearly constrained optimization problem with a quadratic objective function is called “Quadratic programming”

It forms the Basis for the several general Non linear Programming Algorithms

It is procedure that minimizes a quadratic function of n variables subjected to m linear equality or inequality or both types of constraints.
The general quadratic program can be written as
Minimize $f(x) = c^T x + \frac{1}{2} x^T Q x$
subject to $A x \leq b$ and $x \geq 0$

where $c$ is an $n$-dimensional row vector
describing the coefficients of the linear terms in
the objective function, and $Q$ is an $(n \times n)$
symmetric matrix describing the coefficients of
the quadratic terms.
The Lagrangian function for the quadratic program is

\[ L(x, \mu) = cx + \frac{1}{2} x^T Q x + \mu(Ax - b), \]

where \( \mu \) is an \( m \)-dimensional row vector.

And KKT conditions are given by

\[ \frac{\partial L}{\partial x_j} \geq 0, j = 1, 2, \ldots, n \]

\[ c + x^T Q + \mu A \geq 0 \rightarrow 1 \]

\[ \frac{\partial L}{\partial \mu_i} \leq 0, i = 1, 2, \ldots, m \]

\[ Ax - b \leq 0 \rightarrow 2 \]
\[
x_j \frac{\partial L}{\partial x_j} = 0, j = 1, 2, \ldots, n \quad \quad \quad x^T(c^T + Qx + A^T \mu) = 0 \quad \rightarrow 3
\]

\[
\mu_i g_i(x) = 0, \quad i = 1, \ldots, m \quad \mu(Ax - b) = 0 \quad \rightarrow 4
\]

\[
x_j \geq 0, \quad j = 1, 2, \ldots, n \quad \quad \quad x \geq 0 \quad \rightarrow 5
\]

\[
\mu_i \geq 0, \quad i = 1, 2, \ldots, m \quad \mu \geq 0 \quad \rightarrow 6
\]
To put (1) – (6) into a more manageable form we introduce nonnegative surplus variables $y$ to the inequalities in (1) and nonnegative slack variables $v$ to the inequalities in (2) to obtain the equations $c^T + Qx + A^T \mu^T - y = 0$ and $A x - b + v = 0$. 
The KKT conditions can now be written with the constants moved to the right-hand side.

\[ Qx + A^T \mu^T - y = -c^T \quad \rightarrow 7 \]
\[ Ax + v = b \quad \rightarrow 8 \]
\[ x \geq 0, \mu \geq 0, y \geq 0, v \geq 0 \quad \rightarrow 9 \]
\[ y^T x = 0, \mu v = 0 \quad \rightarrow 10 \]

The first two expressions are linear equalities (7 and 8), the third restricts all the variables to be nonnegative (9), and the fourth prescribes complementary slackness (10).
EXAMPLE:
Minimize \( f(x) = -8x_1 - 16x_2 + x_1^2 + 4x_2^2 \)
Subjected to
\( x_1 + x_2 \leq 5 \)
\( x_1 \leq 3 \)
\( x_1 \geq 0, \ x_2 \geq 0 \)

Solution:
\[
\mathbf{c}^T = \begin{bmatrix} -8 \\ -16 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}
\]
\[
\mathbf{b} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}
\]
\( x = [x_1, x_2] \quad y = [y_1, y_2] \quad \mu = [\mu_1, \mu_2] \quad \text{and} \quad v = [v_1, v_2] \)

The linear constraints (7) and (8) take the following form

\[
\begin{align*}
2x_1 + \mu_1 + \mu_2 - x_1 &= 8 \\
8x_2 + \mu_1 - y_2 &= 16 \\
x_1 + x_2 + v_1 &= 5 \\
x_1 + v_2 &= 3
\end{align*}
\]
we add artificial variables to each constraint and minimize their sum.

\[
\begin{align*}
2x_1 + \mu_1 + \mu_2 - y_1 + a_1 &= 8 \\
8x_2 + \mu_1 - y_2 + a_2 &= 16 \\
x_1 + x_2 + v_1 + a_3 &= 5 \\
x_1 + v_2 + a_4 &= 3
\end{align*}
\]

Now by Simplex method we minimize

\[a_1 + a_2 + a_3 + a_4\]

all variables \(\geq 0\) and complementarity conditions
<table>
<thead>
<tr>
<th>BV</th>
<th>x1</th>
<th>x2</th>
<th>μ1</th>
<th>μ2</th>
<th>y1</th>
<th>y2</th>
<th>v1</th>
<th>v2</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>b</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>a2</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>a3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>a4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>-W</td>
<td>-4</td>
<td>-9</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

MOST NEGATIVE VALUE
<table>
<thead>
<tr>
<th>BV</th>
<th>x₁</th>
<th>x₂</th>
<th>μ₁</th>
<th>μ₂</th>
<th>y₁</th>
<th>y₂</th>
<th>v₁</th>
<th>v₂</th>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
<th>b</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>x₂</td>
<td>0</td>
<td>1</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>a₃</td>
<td>1</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>a₄</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

- W  -4  0  -7/8  -1  1  17/8  -1  -1  0  9/8  0  0  14

**MOST NEGATIVE VALUE**
<table>
<thead>
<tr>
<th>BV</th>
<th>x1</th>
<th>x2</th>
<th>μ1</th>
<th>μ2</th>
<th>y1</th>
<th>y2</th>
<th>v1</th>
<th>v2</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>b</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>0</td>
<td>0</td>
<td>3/4</td>
<td>1</td>
<td>-1</td>
<td>-1/4</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td>8/3</td>
</tr>
<tr>
<td>x2</td>
<td>0</td>
<td>1</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>x1</td>
<td>1</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>a4</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>-1/8</td>
<td>0</td>
<td>1</td>
<td>1/8</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>-W</td>
<td>0</td>
<td>0</td>
<td>-11/8</td>
<td>-1</td>
<td>1</td>
<td>13/8</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>5/8</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**MOST NEGATIVE VALUE**
<table>
<thead>
<tr>
<th>BV</th>
<th>x1</th>
<th>x2</th>
<th>μ1</th>
<th>μ2</th>
<th>y1</th>
<th>y2</th>
<th>v1</th>
<th>v2</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>b</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1/2</td>
<td>0</td>
<td>-6</td>
<td>-1/2</td>
<td>0</td>
<td>-6</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1/4</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>x1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>μ1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

-W 0 0 0 -1 1 1/4 1 10 0 2 0 11/8 2

**MOST NEGATIVE VALUE**
<table>
<thead>
<tr>
<th>BV</th>
<th>x₁</th>
<th>x₂</th>
<th>μ₁</th>
<th>μ₂</th>
<th>y₁</th>
<th>y₂</th>
<th>v₁</th>
<th>v₂</th>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
<th>b</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>μ₂</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1/2</td>
<td>0</td>
<td>-6</td>
<td>-1/2</td>
<td>0</td>
<td>-6</td>
<td>2</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>x₂</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1/4</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>x₁</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>μ₁</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>-W</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1.5</td>
<td>0</td>
<td>-25/8</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**MOST NEGATIVE VALUE**
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Basic variables</th>
<th>Solution</th>
<th>Objective value</th>
<th>Entering variable</th>
<th>Leaving variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((a1, a2, a3, a4))</td>
<td>((8, 16, 5, 3))</td>
<td>32</td>
<td>x2</td>
<td>a2</td>
</tr>
<tr>
<td>2</td>
<td>((a1, x2, a3, a4))</td>
<td>((8, 2, 3, 3))</td>
<td>14</td>
<td>x1</td>
<td>a3</td>
</tr>
<tr>
<td>3</td>
<td>((a1, x2, x1, a4))</td>
<td>((2, 2, 3, 0))</td>
<td>2</td>
<td>μ1</td>
<td>a4</td>
</tr>
<tr>
<td>4</td>
<td>((a1, x2, x1, μ1))</td>
<td>((2, 2, 3, 0))</td>
<td>2</td>
<td>μ2</td>
<td>a1</td>
</tr>
<tr>
<td>5</td>
<td>((μ2, x2, x1, μ1))</td>
<td>((2, 2, 3, 0))</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Applied the modified simplex technique to example, yields the sequence of iterations given in Table. The optimal solution to the original problem is \((x^*1, x^*2) = (3, 2)\).
Short Term Course

Advanced Optimization Techniques

Day 2 : (19-May-2015)
Artificial Neural Networks

Dr. Kusum Verma
Assistant Professor,
Department of Electrical Engineering, MNIT
A neuron has a cell body, a branching input structure (the dendrite) and a branching output structure (the axon).

- Axons connect to dendrites via synapses.
- Electro-chemical signals are propagated from the dendritic input, through the cell body, and down the axon to other neurons.
What are ANNs?

- Models of the brain and nervous system
  - Highly Parallel. Process information like the brain.
- Learning
- Very simple principles
- Very complex behaviours

ANNs can be viewed as parallel and distributed processing systems, which consists of a large number of simple and massively connected processors to generate performance similar to that of brain.
Properties of Artificial Neural Networks

- **Learning from experience**: able to solve complex difficult problems, with plenty of data that describe the problem.
- **Generalizing from examples**: Can interpolate from previous learning and give the correct response to unseen data.
- **Rapid applications development**: ANNs are generic machines and quite independent of knowledge about problem domain.
- **Adaptability**: Adapts to a changing environment, if is properly designed.
- **Computational efficiency**: Although the training of a neural network demands a lot of computer power, a trained network demands almost nothing in recall mode.
- **Non-linearity**: Not based on linear assumptions about the real world.
Appropriate Problem Domains for Neural Network Learning

- Input is high-dimensional discrete or real-valued
- Output is discrete or real-valued
- Output is a vector of values
- Form of target function is unknown
- Humans do not need to interpret the results
Applications

- Ability to model linear and non-linear systems without the need to make assumptions implicitly.

- Applied in almost every field of science and engineering. Few of them are:
  - Function approximation, or regression analysis, including time series and modelling.
  - Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
  - Data processing, including filtering, clustering, blind signal separation and compression.
  - Computational neuroscience and neurohydrodynamics
  - Forecasting and prediction
  - Estimation and control
Types of Problems

- Mathematical Modeling (Function Approximation)
- Classification
- Clustering
- Forecasting
- Vector Quantization
- Pattern Association
- Control
- Optimization
Mathematical Modeling
(Function Approximation)

- Modeling – often mathematical relationship between two sets of data unknown analytically
- No closed form expression available
- Empirical data available defining output parameters for each input
- Data is often noisy
- Need to construct a mathematical model that correctly generates outputs from inputs

Learning a mapping between an input and an output space from a set of input-output data is the core concern in diverse real world applications
Classification

- Assignment of objects to specific class
- Given a database of objects and classes of those objects

Examples-
In marketing: consumer spending pattern classification
In defence: radar and sonar image classification
In agriculture & fishing: fruit and catch grading
In medicine: ultrasound and electrocardiogram image classification, EEGs, medical diagnosis
Clustering

- Grouping together objects similar to one another
- Usually based on some “distance” measurement in object parameter space
- Objects and distance relationships available
- No prior info on classes or groupings
- Similar to statistical k-nearest neighbor clustering method
Forecasting

- Prediction of future events based on history
- Laws underlying behavior of system sometimes hidden; too many related variables to handle
- Trends and regularities often masked by noise
- Time series forecasting
- Weather, Stock market indices, machine performance
Vector Quantization

- Object space divided into several connected regions
- Objects classified based on proximity to regions
- Closest region or node is “winner”
- Form of compression of high dimensional input space
- Successfully used in many geological and environmental classification problems where input object characteristics is often unknown.
Pattern Association

- An associate network is a neural network with essentially a single functional layer that associates one set of vectors with another set of vectors.
- Auto-associative systems useful when incoming data is a corrupted version of actual object e.g. face detection, handwriting detection etc.
- May require several iterations of repeated modification of input.
Control

- Manufacturing, Robotic and Industrial machines have complex relationships between input and output variables.
- Output variables define state of machine.
- Input variables define machine parameters determined by operation conditions, time and human input.
- System may be static or dynamic.
- Need to map inputs to outputs for stable smooth operation.
Optimization

- Requirement to improve system performance or costs subject to constraints
- Maximize or Minimize
- Large number of variables affecting objective function (high dimensionality of problem)
- Design variables often subject to constraints
- Lots of local minima
- Neural nets can be used to find global optima
An ANN is composed of processing elements called or *perceptrons*, organized in different ways to form the network’s structure. Each of the perceptrons receives inputs, processes inputs and delivers a single output.

Note the three layers: input, intermediate (called the *hidden layer*) and output. Several hidden layers can be placed between the input and output layers.
The Artificial Neural Network

- Each ANN is composed of a collection of perceptrons grouped in layers.
- **Inputs** $x_i$ arrive through pre-synaptic connections.
- Synaptic efficacy is modeled using real weights $w_i$.
- The response of the neuron is a **nonlinear function** $f$ of its weighted inputs.
The Neuron

The neuron is the basic information processing unit of a NN. It consists of:

1. A set of synapses or connecting links, each link characterized by a weight: $W_1, W_2, \ldots, W_N$.

2. An adder function (linear combiner) which computes the weighted sum of the inputs:

   $$ u = \sum_{k=1}^{N} w_k x_k $$

3. Activation function (squashing function) $\varphi$ for limiting the amplitude of the output of the neuron:

   $$ y = \varphi(u + b) $$
Mathematically, the output expression of the network is given as

$$Y = F(S) = F\left[\sum_{K=1}^{N} X_K W_K + b\right]$$
Activation functions

- **Hard-Limit Transfer Function**
  \[ a = \text{hardlim}(n) \]

- **Linear Transfer Function**
  \[ a = \text{purelin}(n) \]

- **Log-Sigmoid Transfer Function**
  \[ a = \text{logsig}(n) \]

- **Satlin Transfer Function**
  \[ a = \text{satlin}(n) \]

- **Tan-Sigmoid Transfer Function**
  \[ a = \text{tansig}(n) \]

- **Radial Basis Function**
  \[ a = \text{radbas}(n) \]
Neural Network Types

- **Feed-forward**
  - Input signals travel through input layer, hidden layers, and output layer
  - Require training

- **Feedback**
  - Network pays attention to its own results and uses the results to adjust errors
  - Tend to be constructed rather than trained
Feed-forward Neural Networks

Single Layer Feed-forward NN

Input layer of source nodes

Output layer of neurons

Multi Layer Feed-forward NN

Input layer

Hidden Layer

3-4-2 Network

AOT'15, MNIT, Jaipur, 18-22 May 2015
Designing ANN models follows a number of systemic procedures. In general, there are five basic steps:

1. Collecting data,
2. Preprocessing data
3. Building the network
4. Train, and
5. Test performance of model
INTRODUCTION TO NN TOOLBOX

Features

- It supports a wide variety of feed-forward and recurrent networks, including perceptrons, radial basis networks, BP networks, learning vector quantization (LVQ) networks, self-organizing networks, Hopfield and Elman NWs, etc.

- It also supports the activation function types of bi-directional linear with hard limit (satlins) and without hard limit, threshold (hard limit), signum (symmetric hard limit), sigmoidal (log-sigmoid), and hyperbolic tan (tan-sigmoid).
Features

- In addition, it supports unidirectional linear with hard limit (satlins) and without hard limit, radial basis and triangular basis, and competitive and soft max functions. A wide variety of training and learning algorithms are supported.
Parameter setting

- Number of layers
- Number of neurons
  - too many neurons, require more training time
- Learning rate
  - from experience, value should be small $\sim 0.1$
- Momentum term
- ..
Simple Neuron

Input → Neuron without bias → $a = f(wp)$

Input → Neuron with bias → $a = f(wp+b)$
Neuron with Vector Input

\[ n = w_{1,1}p_1 + w_{1,2}p_2 + ... + w_{1,R}p_R + b \]

Where

\[ R = \text{number of elements in input vector} \]
Perceptrons

A = hardlim(Wp + b)

weights \( w_{1,1} = -1 \), \( w_{1,2} = 1 \) and a bias \( b = 1 \).

Where

\[ R = \text{number of elements in input vector} \]

Hard-Limit Transfer Function

\[ a = \text{hardlim}(n) \]

Where...

\( w_{1,1} = -1 \) and \( b = +1 \)

\( w_{1,2} = +1 \)
Network Architectures

Inputs Layer of Neurons

\[ a = f(Wp + b) \]

Where

\[ W = \begin{bmatrix}
    w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\
    w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{S,1} & w_{S,2} & \cdots & w_{S,R}
\end{bmatrix} \]

\[ R = \text{number of elements in input vector} \]

\[ S = \text{number of neurons in layer} \]
Multiple Layers of Neurons

\[ a^1 = f^1(IW^{1,1}p + b^1) \]
\[ a^2 = f^2(LW^{2,1}a^1 + b^2) \]
\[ a^3 = f^3(LW^{3,1}a^1 + b^3) \]
\[ a^3 = f^3(LW^{3,2}a^2 + b^3) \]
Other architectures

Radial Basis Functions

\[ a_1 = \text{radbas} \left( \| \mathbf{I} \mathbf{W}_{1,1} \| \mathbf{p}^\top \mathbf{b}_1 \right) \]

\[ a_2 = \text{purelin} \left( \mathbf{L} \mathbf{W}_{2,1} a_1 + \mathbf{b}_2 \right) \]

The transfer function for a radial basis neuron is

\[ \text{radbas}(n) = e^{-n^2} \]
Other architectures

Probabilistic Neural Networks

Input

Radial Basis Layer

Competitive Layer

Where...

$R = \text{number of elements in input vector}$

$a^1 = \text{radbas} (\| IW_{1,1} - \mathbf{b} \|)$

$a^2 = \text{compe} (LW_{2,1}, a^1)$

$a_i^1 = \text{ith element of } a^1$ where $IW_{1,1}$ is a vector made of the $i$th row of $IW_{1,1}$

$Q = \text{number of input/target pairs}$

$K = \text{number of classes of input data}$

$K = \text{number of neurons in layer 2}$

$= \text{number of neurons in layer 1}$
Other architectures

Generalized Regression Networks

\[
\begin{align*}
    a_1 &= \text{radbas}(\|IW_{1,1} - p\| \cdot b_1) \\
    a_2 &= \text{purelin}(n_2)
\end{align*}
\]

Where...

- \( R \) = no. of elements in input vector
- \( Q \) = no. of neurons in layer 1
- \( Q \) = no. of neurons in layer 2
- \( Q \) = no. of input/target pairs
Other architectures

Self Organising Maps (Kohonen)
Normalization

Normalization procedure before presenting the input data to the network is required since mixing variables with large magnitudes and small magnitudes will confuse the learning algorithm on the importance of each variable and may force it to finally reject the variable with the smaller magnitude.
Building the Network

- At this stage, the designer specifies the number of hidden layers, neurons in each layer, transfer function in each layer, training function, weight/bias learning function, and performance function.
Basic Types of Learning

- Neural Nets
  - Supervised Learning
  - Unsupervised Learning
**Error Back Propagation Algorithm**

Architecture of MLP indicating flow of input and error signals

\[ E = \frac{1}{2} \sum_{i=1}^{n} (O_i^{\text{actual}} - O_i^{\text{net}}) \]
- Initialize network with random weights
- For all training cases (called examples):
  - Present training inputs to network and calculate output
  - For all layers (starting with output layer, back to input layer):
    - Compare network output with correct output (error function)
    - Adapt weights in current layer

- Use gradient descent to minimize the error
  propagate deltas to adjust for errors
  backward from outputs
  to hidden layers
  to inputs
Backpropagation learning

1. Prop. to desired values
2. Backprop output layer

3. Hidden error values
4. and so on …
TRAINING THE NETWORK

- During the training process, the weights are adjusted to make the actual outputs (predicted) close to the target (measured) outputs of the network.
- Fourteen types of training algorithms for developing the MLP network.
- MATLAB provides built-in transfer functions
  - linear (purelin), Hyperbolic Tangent Sigmoid (tansig) and Logistic Sigmoid (logsig).
### Training Algorithm

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>trainlm</td>
<td>Levenberg-Marquardt</td>
</tr>
<tr>
<td>BFG</td>
<td>trainbfg</td>
<td>BFGS Quasi-Newton</td>
</tr>
<tr>
<td>RP</td>
<td>trainrp</td>
<td>Resilient Backpropagation</td>
</tr>
<tr>
<td>SCG</td>
<td>trainscg</td>
<td>Scaled Conjugate Gradient</td>
</tr>
<tr>
<td>CGB</td>
<td>traincgb</td>
<td>Conjugate Gradient with Powell/Beale Restarts</td>
</tr>
<tr>
<td>CGF</td>
<td>traincgf</td>
<td>Fletcher-Powell Conjugate Gradient</td>
</tr>
<tr>
<td>CGP</td>
<td>traincgp</td>
<td>Polak-Ribière Conjugate Gradient</td>
</tr>
<tr>
<td>OSS</td>
<td>trainoss</td>
<td>One Step Secant</td>
</tr>
<tr>
<td>GDX</td>
<td>traingdx</td>
<td>Variable Learning Rate Backpropagation</td>
</tr>
</tbody>
</table>
Classify Patterns with a Neural Network

- Use the `nprttool` GUI, as described in Using the Neural Network Pattern Recognition Tool.
- Use a command-line solution, as described in Using Command-Line Functions.
Pattern Classification Process

- Training Set
- Learning Algorithm
- Building The Classification Model
- Model Selection
- Testing and Validation
Welcome to the Neural Network Pattern Recognition Tool.

Solve a pattern-recognition problem with a two-layer feed-forward network.

Introduction

In pattern recognition problems, you want a neural network to classify inputs into a set of target categories.

For example, recognize the vineyard that a particular bottle of wine came from, based on chemical analysis (wine_dataset); or classify a tumor as benign or malignant, based on uniformity of cell size, clump thickness, mitosis (cancer_dataset).

The Neural Network Pattern Recognition Tool will help you select data, create and train a network, and evaluate its performance using cross-entropy and confusion matrices.

Neural Network

A two-layer feed-forward network, with sigmoid hidden and softmax output neurons (patternnet), can classify vectors arbitrarily well, given enough neurons in its hidden layer.

The network will be trained with scaled conjugate gradient backpropagation (trainscg).

To continue, click [Next].
Select Data

What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

- Inputs: [none]

Target data defining desired network output.

- Targets: [none]

Samples are:

- Matrix columns
- Matrix rows

Want to try out this tool with an example data set?

Load Example Data Set

Select inputs and targets, then click [Next].
Select a data set:

- Simple Classes
- Iris Flowers
- Breast Cancer
- Types of Glass
- Thyroid
- Wine Vintage

Description

Filename: cancer_dataset

Classify whether a breast tumor is benign or malignant based on nine characteristics of sample biopsies.

Samples may be classified using clustering (using only input data) or with Pattern Recognition or Fitting (with input and target data).

The data set consists of 699 samples.

"cancer_inputs" is a 9x699 matrix, whose rows are biopsy characteristics:

1. Clump Thickness
2. Uniformity of Cell Size
3. Uniformity of Cell Shape
4. Marginal Adhesion
5. Single Epithelial Cell Size
6. Bare Nuclei
7. Bland Chromatin
8. Normal Nucleoli
9. Mitoses

Import  Cancel
### Validation and Test Data
Set aside some samples for validation and testing.

<table>
<thead>
<tr>
<th>Select Percentages</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training:</strong></td>
<td>Three Kinds of Samples:</td>
</tr>
<tr>
<td>70%</td>
<td>Training:</td>
</tr>
<tr>
<td>489 samples</td>
<td>These are presented to the network during training, and the network is adjusted according to its error.</td>
</tr>
<tr>
<td><strong>Validation:</strong></td>
<td>Validation:</td>
</tr>
<tr>
<td>15%</td>
<td>These are used to measure network generalization, and to halt training when generalization stops improving.</td>
</tr>
<tr>
<td>105 samples</td>
<td><strong>Testing:</strong></td>
</tr>
<tr>
<td><strong>Testing:</strong></td>
<td>These have no effect on training and so provide an independent measure of network performance during and after training.</td>
</tr>
</tbody>
</table>

Change percentages if desired, then click [Next] to continue.

- Neural Network Start
- Welcome
**Train Network**

Train the network to classify the inputs according to the targets.

**Train Network**

Train using scaled conjugate gradient backpropagation (trainscg)

**Results**

<table>
<thead>
<tr>
<th>Samples</th>
<th>MSE</th>
<th>%ME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training: 489</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Validation: 105</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Testing: 105</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

**Notes**

- Training multiple times will generate different results due to different initial conditions and sampling.

- Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

- Percent Error indicates the fraction of samples which are misclassified. A value of 0 means no misclassifications, 100 indicates maximum misclassifications.

"Train network, then click [Next]."
Criteria that can be used to stop network training

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Stopping Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>min_grad</td>
<td>Minimum Gradient Magnitude</td>
</tr>
<tr>
<td>max_fail</td>
<td>Maximum Number of Validation Increases</td>
</tr>
<tr>
<td>time</td>
<td>Maximum Training Time</td>
</tr>
<tr>
<td>goal</td>
<td>Minimum Performance Value</td>
</tr>
<tr>
<td>epochs</td>
<td>Maximum Number of Training Epochs (Iterations)</td>
</tr>
</tbody>
</table>
Evaluate Network

Optionally test network on more data, then decide if network performance is good enough.

Iterate for improved performance

Try training again if a first try did not generate good results or you require marginal improvement.

Adjust Network Size

Increase network size if retraining did not help.

Not working? You may need to use a larger data set.

Import Larger Data Set

Optionally perform additional tests

Inputs:

Targets:

Samples are:

No inputs selected.

No targets selected.

Test Network

MSE

%E

Plot Confusion

Plot ROC

Select inputs and targets, click an improvement button, or click [Next]

Neural Network Start

Welcome

[Back]

[Next]

[Cancel]
% cancerInputs - input data.
% cancerTargets - target data.
inputs = cancerInputs;
targets = cancerTargets;
% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);
% Set up Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
% Train the Network
[net,tr] = train(net,inputs,targets);
% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);
% View the Network
view(net)
% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotconfusion(targets,outputs)
% figure, ploterrhist(errors)
network performance only on the test set

Best Validation Performance is 0.039514 at epoch 8

Confusion Matrix

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Output Class 1</th>
<th>Output Class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>446 63.8%</td>
<td>5 0.7%</td>
</tr>
<tr>
<td>2</td>
<td>12 1.7%</td>
<td>236 33.8%</td>
</tr>
<tr>
<td>1</td>
<td>97.4% 2.6%</td>
<td>97.9% 2.1%</td>
</tr>
<tr>
<td>2</td>
<td>95.2% 4.8%</td>
<td>97.6% 2.4%</td>
</tr>
</tbody>
</table>
more accurate results?

- Reset the initial network weights and biases to new values with `init` and train again.
- Increase the number of hidden neurons.
- Increase the number of training vectors.
- Increase the number of input values, if more relevant information is available.
- Try a different training algorithm.

Each time a neural network is trained, can result in a different solution due to different initial weight and bias values and different divisions of data into training, validation, and test sets.

As a result, different neural networks trained on the same problem can give different outputs for the same input. To ensure that a neural network of good accuracy has been found, retrain several times.
The R value is an indication of the relationship between the outputs and targets. If $R = 1$, this indicates that there is an exact linear relationship between outputs and targets. If $R$ is close to zero, then there is no linear relationship between outputs and targets.
Support Vector Machines

Dr. Kusum Verma
Assistant Professor,
Department of Electrical Engineering, MNIT
What is a Support Vector Machine?

- SVM is one of the supervised learning algorithms.
- It uses concepts from computational learning theory.
- It is an optimally defined surface, typically nonlinear in the input space, and linear in a higher dimensional space.
- Implicitly defined by a kernel function.
What is SVM used for?

- Regression and data-fitting
- Supervised and unsupervised learning
- Image Processing
- **Speech Recognition**
- **Pattern Recognition**
- Time-Series Analysis
- Radar Point Source Location
- Medical Diagnosis
- Process Faults Detection
Binary Linear Classifier

- If the training set is linearly separable: Binary classification can be viewed as the task of separating classes in feature space:

The hypersurface equation: \( f(x, w, b) = \text{sign}(w^T x + b) \)
Classification Margin

- Distance from an object to the hyperplane is:
  \[ r = \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} \]

- **Support Vectors** *(the closest objects to the hyperplane)*

- **Margin** *(the width that the boundary could be increased by before hitting a data point)*, \( \rho \).
  \[ \rho = \frac{2}{\|\mathbf{w}\|} \]

- The **maximum margin linear classifier** is the linear classifier with the maximum margin.

![Diagram](image-url)
Linear SVM Mathematically

What we know:

- $w \cdot x^+ + b = +1$
- $w \cdot x^- + b = -1$
- $w \cdot (x^+ - x^-) = 2$

\[ M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|} \]
Linear SVM Classifier Mathematically

- Assuming all data is at least distance 1 from the hyperplane, the following two constraints follow for a training set \( \{(x_i, y_i)\} \):
  
  \[ w^T x_i + b \geq 1 \quad \text{if} \quad y_i = 1, \quad w^T x_i + b \leq -1 \quad \text{if} \quad y_i = -1 \]

- For **support vectors**, the inequality becomes an equality; since each example’s distance from the hyperplane is: \( \frac{w^T x_i + b}{\|w\|} \), the **margin** is:
  
  \[ r = \frac{w^T x + b}{\|w\|} \]

Find \( w \) and \( b \) such that \[ \rho = \frac{2}{\|w\|} \] is maximized, for all \( \{(x_i, y_i)\} \)

\[ w^T x_i + b \geq 1 \quad \text{if} \quad y_i = 1; \quad \text{and} \quad w^T x_i + b \leq -1 \quad \text{if} \quad y_i = -1 \]

**A better formulation:**

Find \( w \) and \( b \) such that \[ \Psi(w) = \frac{1}{2} w^T w \] is minimized for \( \{(x_i, y_i)\} \), such that

\[ y_i (w^T x_i + b) \geq 1 \]
Solving the Optimization Problem

In most real applications, we need to optimize a **quadratic function** subject to **linear constraints**. The solution involves constructing a **dual problem** where a **Lagrange multiplier** $\lambda\lambda_i$ is associated with every constraint in the primary problem:

$$Q(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^T x_j$$

1. $\Sigma \lambda_i y_i = 0$,
2. $\lambda_i \geq 0$ for all $\lambda_i$

The solution:

$$w = \sum \lambda_i y_i x_i, \quad b = y_k - w^T x_k$$

for any $x_k$ such that $\lambda_k \neq 0$

The classifying decision function is:

$$f(x) = \text{sign}(\Sigma \lambda_i y_i x_i^T x + b)$$
Linear SVMs: Summary

- The classifier is a *separating hyperplane*.

- The most “important” training points are the support vectors; they define the hyperplane.

- Quadratic optimization algorithms can identify which training points $x_i$ are support vectors with non-zero Lagrangian multipliers $\lambda_i$.

- Both in the dual formulation of the problem and in the solution, training points appear only inside inner products.
Non-linear SVMs: Feature spaces

- What if the training set is not linearly separable?
- The original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:

\[ \Phi: x \rightarrow \phi(x) \]
Nonlinear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space.
- It does not need to represent the space explicitly, simply by defining a kernel function.
- The kernel function plays the role of the dot product in the feature space.
The “Kernel Functions”

• If every data point is mapped into high-dimensional space via some transformation \( \Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x}) \), the inner product becomes:
  \[
  K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j).
  \]

• The linear classifier relies on inner product between vectors
  \[
  K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j
  \]

• A kernel matrix (function) is some function that corresponds to an inner product into some feature space.

• For each \( K(\mathbf{x}_i, \mathbf{x}_j) \) checking that \( K(\mathbf{x}_i, \mathbf{x}_j) \) can be written in the form of \( \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j) \) or not.

Why use kernels?

• Make non-separable problem separable.
• Map data into better representational space
Examples of Kernel Functions

- **Linear:** 
  \[ K(x_i, x_j) = x_i^T x_j \]

- **Polynomial of power \( p \):** 
  \[ K(x_i, x_j) = (1 + x_i^T x_j)^p \]

- **Gaussian (radial-basis function network):**
  \[ K(x_i, x_j) = \exp\left( -\frac{|x_i - x_j|^2}{2\sigma^2} \right) \]

- **Sigmoid:** 
  \[ K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1) \]
Non-Linear SVM Classifier Mathematically

- Slack variables $\eta_i$ can be added to allow misclassification of noisy examples.

Find $w$ and $b$ such that

$$\Psi(W, \eta) = \frac{1}{2} W^T W + C \sum_{i=1}^{n} \eta_i$$

is minimized for all $\{(x_i, y_i)\}$, where

$$y_i (w^T \varphi(x_i) + b) \geq 1 - \eta_i \quad \text{and} \quad \eta_i \geq 0, \quad i=1,2,...,n.$$ 

$C$ is a user-specified positive number.

Using $\lambda_i$ and $\gamma_i$ as Lagrange multipliers, the unconstrained cost function becomes

$$L(w, b, \eta) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \eta_i - \sum_{i=1}^{n} \lambda_i \left( y_i (w^T \varphi(x_i) + b) - 1 + \eta_i \right) - \sum_{i=1}^{n} \gamma_i \eta_i$$
Solving the Optimization Problem

- The dual problem for non-linear SVM classification:

Find \( \lambda_1, \ldots, \lambda_n \) such that

\[
Q(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(x_i, x_j)
\]

is maximized subject to the constraints

\[
\begin{align*}
(1) & \quad \sum \lambda_i y_i = 0, \\
(2) & \quad 0 \leq \lambda_i \leq C \text{ for all } \lambda_i
\end{align*}
\]

- Again, \( x_i \) with non-zero \( \lambda_i \) will be support vectors. Solution to the dual problem is:

\[
\begin{align*}
\omega_o &= \sum_{i} \lambda_i y_i \phi(x_i) \\
\beta_o &= \left\{ \sum_{i \in I_o} \left[ 1 - y_i \omega_o^T \phi(x_i) \right] \right\} / \left( \sum_{i \in I_o} y_i \right)
\end{align*}
\]

where \( I_o = \{i; 0 < \lambda_i < C\} \)

The decision function is:

\[
f(x) = \text{sign}(\sum \lambda_i y_i K(x_i, x) + b)
\]
Model Selection

We select the best model as follows:

1. **Selection methods:**
   - Backward-Forward (BF)
   - Forward-Backward (FB)

2. **Quality Criteria:** We use the minimum description length (MDL), that is,

   \[
   L(X) = \frac{m \log(nk)}{2} + \frac{nk}{2} \log \left( \frac{1}{nk} \sum_{i=1}^{n} \varepsilon_i \left( \theta_{ik} \right)^2 \right) \quad (7)
   \]

   **Penalty for complexity**

   **Goodness of fit**

The model with the smallest MDL is the best model.
**Expected Risk, Structural Risk Minimization (SRM)**

- **Classification Problem**
  - Decision functions
  
  \[ f_\lambda(x) : x \rightarrow y, x \in \mathbb{R}^p, y \in \{-1, 1\} \]

  or

  \[ f_\lambda(x) : x \rightarrow y, x \in \mathbb{R}^p, y \in \{0, 1, \ldots, c - 1\} \]

- **Expected Risk:**

  \[ R(\lambda) = \int | f_\lambda(x) - y | \, P(x, y) \, dx \, dy \]

- **Ideal Goal**: Find the decision function \( f(x) \) minimize the expected risk

  - Empirical Risk Minimization:

    \[ R_{emp}(\lambda) = \frac{1}{n} \sum \left| f_\lambda(x_i) - y_i \right| \]

- To avoid the over-fitting problem, use SRM or minimum description length (MDL), where the empirical risk should be minimized
Model Validation

We validate the classification model using two methods:

1. **Internal Validation:**

   1. The correct classification rate (CCR)

   \[
   CCR = n^{-1} \sum_{k=1}^{c} CC_k, \quad (8)
   \]

   Where \( CC_k \) is the number of corrected observations in the class \( k \). The model with the highest CCR is the better performance.

   2. The average of squared classification error (SSCE)

   \[
   SSCE = n^{-1} \sum_{k=1}^{c} [n_k - CC_k]^2, \quad (9)
   \]

   The model with the smallest SSCE is the better performance.

Note: CCR and SSCE are computed for the training set.
Some Issues

• Choice of kernel
  - Gaussian or polynomial kernel is default
  - if ineffective, more elaborate kernels are needed
  - domain experts can give assistance in formulating appropriate
    similarity measures

• Choice of kernel parameters
  - e.g. $\sigma$ in Gaussian kernel
  - $\sigma$ is the distance between closest points with different classifications
  - In the absence of reliable criteria, applications rely on the use of a
    validation set or cross-validation to set such parameters.

• Optimization criterion – Hard margin v.s. Soft margin
  - a lengthy series of experiments in which various parameters are tested
The **Matlab** code for least squares support vector machine for arbitrary Kernel to compute $b$ and $\lambda$ is written as:

```matlab
function [lambda,b]=svmls(Gamma,Y,C);
% Gamma n x m is n input data points, n >> m
[n,m]=size(Gamma);
for i=1:n,
    Gamma(i,:)=Y(i)*Gamma(i,:);
end
B=inv(eye(m)/C+Gamma'*Gamma);
W=B*Gamma';
X=C*Y'-C*(Y'*Gamma)*W;
u=ones(n,1); b=(X*u)/(X*Y);
lambda=C*u-(C*Gamma)*(W*u)-b*X';
```
Advanced Optimization Techniques (AOT-15)

Simulated Annealing

Dr. Gunjan Soni
Assistant Professor, Dept. of Mech. Engg.
Simulated annealing (SA)

- Simulated Annealing is a method based on simulation of physical annealing.

- Annealing: Heat (metal or glass) and allow it to cool slowly, in order to remove internal stresses and toughen it.

- SA makes use of an iterative improvement procedure.

- It starts with an initial solution.

- SA avoids being trapped in local optima by accepting sometimes uphill moves.
Ball example – SA vs Greedy Algorithms

Greedy Algorithm gets stuck here! Locally Optimum Solution.

Simulated Annealing explores more. Chooses this move with a small probability (Hill Climbing).

Upon a large no. of iterations, SA converges to this solution.

Initial position of the ball
SA - Procedure

**SA Parameters**

- $Z_c = \text{objective function value for the current trial solution}$,

- $Z_n = \text{objective function value for the current candidate to be the next trial solution}$,

- $T = \text{temperature}$ (a parameter that measures the tendency to accept the current candidate to be the next trial solution if this candidate is not an improvement on the current trial solution).
SA - Procedure

**Move selection rule: For maximization problem**

- If $Z_n \geq Z_c$ always accept this candidate.
- If $Z_n < Z_c$ accept the candidate with the following probability:
  \[
  \text{Prob}\{\text{acceptance}\} = e^x, \quad x = \frac{Z_n - Z_c}{T}
  \]
  compare a random number between 0 and 1 to the probability of acceptance
  - If *random number* < Prob{acceptance}, accept a downward step.
  - Otherwise, reject the step.
SA - Procedure

Move selection rule: For minimization problem

- If $Z_n \leq Z_c$ always accept this candidate.
- If $Z_n > Z_c$ accept the candidate with the following probability:

$$\text{Prob}\{\text{acceptance}\} = e^x$$

where $x = (Z_c - Z_n) / T$

compare a random number between 0 and 1 to the probability of acceptance

- If random number $< \text{Prob}\{\text{acceptance}\}$, accept a downward step.
- Otherwise, reject the step.
Example

Let us consider the example of a small nonlinear programming problem (only a single variable). The problem is to

Maximize

\[ f(x) = 12x^5 - 975x^4 + 28000x^3 - 345000x^2 + 1800000x \]

Subject to

\[ 0 \leq x \leq 31 \]
N = number of constraints = 1
U = 31
L = 0
• Initial trial solution (ITS): $x = 15.5$ (between U and L limit)
• Neighborhood structure: Any feasible solution

SD\((j)\) = \((U(j) - L(j)) / 6\) for \(j = 1,2,\ldots,n\)

Then given the trial solution \((x(1), x(2),\ldots,x(n))\).
Reset \(x(j) = x(j) + N(0, SD\,(j))\), for \(j = 1,2,\ldots,n\)
• When variable $x(j)$ is midway between $U(j)$ and $L(j)$ any new feasible values is within 3 SD. It sets bound on movement of new value.

$x = 15.5$, thus

$Z(c) = f(15.5) = 3741121$

and $T(1) = 0.2 \times Z(c) = 748224$

$SD = (U-L)/6 = 5.167$

Obtain a random number ‘$r’ = 0.0735$

Find $p(Z \leq -1.45) = 0.0735$
Now, \(-1.45 = (0-x’)/5.167\), \(x’ = 7.5\)
\[x = 15.5 - 7.5 = 8\]
Hence, \(Z(n) = f(x) = 3055616\)

Here, \(Z(n) < Z(c)\), being a maximization problem, we will have to test whether to accept a downward step of not.

\[
\frac{(Z(n) - Z(c))}{T} = -0.916
\]

\(P\text{ (acceptance)} = e^{(-0.916)} = 0.4\)
Now, let the generated random number be 0.36
Since \(RN\) generated > \(P\text{(acceptance)}\), we accept a downward step.
SA - Procedure

SA Parameters

Two kinds of decisions have to be taken heuristically:

Generic decisions
• Can be taken without a deep insight into the particular problem
• Are tuned experimentally

Problem specific decisions
• Are related to the nature of the particular problem
• Need a good understanding of the problem
Generic Decisions

• Initial temperature

• Temperature schedule

• Cooling ratio

• Stopping criterion
SA Cooling Schedule

• Starting Temperature

• Final Temperature

• Temperature Decrement

• Iterations at each temperature
SA Cooling Schedule - Starting Temperature

- Starting Temperature
  - Must be *hot* enough to allow moves to *almost* neighbourhood state (else we are in danger of implementing hill climbing)
  - Must *not* be so hot that we conduct a random search for a period of time
  - Problem is finding a suitable starting temperature
SA Cooling Schedule - Temperature Decrement

- Temperature Decrement
  - Theory states that we should allow enough iterations at each temperature so that the system stabilises at that temperature
  - Unfortunately, theory also states that the number of iterations at each temperature to achieve this might be exponential to the problem size
SA Cooling Schedule - Temperature Decrement

- Temperature Decrement

- We can either do this by doing a large number of iterations at a few temperatures, a small number of iterations at many temperatures or a balance between the two
SA Cooling Schedule - Iterations

Iterations at each temperature

• A constant number of iterations at each temperature

• An alternative is to dynamically change the number of iterations as the algorithm progresses

• At lower temperatures it is important that a large number of iterations are done so that the local optimum can be fully explored

• At higher temperatures, the number of iterations can be less
## Iteration Table

<table>
<thead>
<tr>
<th>Iteration</th>
<th>T</th>
<th>Trial Solution Obtained</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x = 15.5</td>
<td>3,741,021.0</td>
</tr>
<tr>
<td>1</td>
<td>748,224</td>
<td>x = 17.557</td>
<td>1,427,533.956</td>
</tr>
<tr>
<td>2</td>
<td>748,224</td>
<td>x = 14.832</td>
<td>1,790,466.203</td>
</tr>
<tr>
<td>3</td>
<td>748,224</td>
<td>x = 17.681</td>
<td>4,188,641.364</td>
</tr>
<tr>
<td>4</td>
<td>748,224</td>
<td>x = 16.662</td>
<td>3,995,966.078</td>
</tr>
<tr>
<td>5</td>
<td>748,224</td>
<td>x = 18.444</td>
<td>4,299,788.258</td>
</tr>
<tr>
<td>6</td>
<td>374,112</td>
<td>x = 19.445</td>
<td>4,386,985.033</td>
</tr>
<tr>
<td>7</td>
<td>374,112</td>
<td>x = 21.432</td>
<td>4,302,136.329</td>
</tr>
<tr>
<td>8</td>
<td>374,112</td>
<td>x = 18.642</td>
<td>4,322,687.873</td>
</tr>
<tr>
<td>9</td>
<td>374,112</td>
<td>x = 22.432</td>
<td>4,113,901.493</td>
</tr>
<tr>
<td>10</td>
<td>374,112</td>
<td>x = 20.081</td>
<td>4,345,233.403</td>
</tr>
<tr>
<td>11</td>
<td>187,056</td>
<td>x = 20.383</td>
<td>4,393,306.255</td>
</tr>
<tr>
<td>12</td>
<td>187,056</td>
<td>x = 21.216</td>
<td>4,330,358.125</td>
</tr>
<tr>
<td>13</td>
<td>187,056</td>
<td>x = 21.354</td>
<td>4,313,392.276</td>
</tr>
<tr>
<td>14</td>
<td>187,056</td>
<td>x = 20.795</td>
<td>4,370,624.01</td>
</tr>
<tr>
<td>15</td>
<td>187,056</td>
<td>x = 18.895</td>
<td>4,348,060.727</td>
</tr>
<tr>
<td>16</td>
<td>93,528</td>
<td>x = 21.714</td>
<td>4,259,787.734</td>
</tr>
<tr>
<td>17</td>
<td>93,528</td>
<td>x = 19.463</td>
<td>4,387,360.1</td>
</tr>
<tr>
<td>18</td>
<td>93,528</td>
<td>x = 20.389</td>
<td>4,393,076.988</td>
</tr>
<tr>
<td>19</td>
<td>93,528</td>
<td>x = 19.83</td>
<td>4,398,710.575</td>
</tr>
<tr>
<td>20</td>
<td>93,528</td>
<td>x = 20.68</td>
<td>4,378,591.085</td>
</tr>
<tr>
<td>21</td>
<td>46,764</td>
<td>x = 20.031</td>
<td>4,399,955.913 ← Maximum</td>
</tr>
<tr>
<td>22</td>
<td>46,764</td>
<td>x = 20.184</td>
<td>4,398,462.299</td>
</tr>
<tr>
<td>23</td>
<td>46,764</td>
<td>x = 19.9</td>
<td>4,399,551.462</td>
</tr>
<tr>
<td>24</td>
<td>46,764</td>
<td>x = 19.677</td>
<td>4,395,385.618</td>
</tr>
<tr>
<td>25</td>
<td>46,764</td>
<td>x = 19.377</td>
<td>4,383,048.039</td>
</tr>
</tbody>
</table>
The local optima are at $x = 5$, $x = 20$, and $x = 31$, but only $x = 20$ is a global optimum.
Convergence of simulated annealing

Unconditional Acceptance

Move accepted with probability

\[ P = e^{\frac{-C}{\text{temp}}} \]

AOT'15, MNIT, Jaipur, 18-22 May 2015
Outline of a Basic SA

• **Initialization.** Start with a feasible initial trial solution.

• **Iteration.** Use the *move selection rule* to select the next trial solution. (If none of the immediate neighbours of the current trial solution are accepted, the algorithm is terminated.)
Outline of a Basic SA

• **Check the temperature schedule.** When the desired number of iterations have been performed at the current value of $T$, decrease $T$ to the next value in the temperature schedule and resume performing iterations at this next value.

• **Stopping rule.** When the desired *number of iterations* have been performed at the smallest value of $T$ in the temperature schedule (or when none of the immediate neighbours of the current trial solution are accepted), stop. Accept the best trial solution found at any iteration (including for larger values of $T$) as the final solution.
Solution using MATLAB

Coding the Objective Function

To code the Objective Function we create a MATLAB file named `simple_objective.m` with the following code in it:

```matlab
function y = simple_objective(x)
    y = -(12*x^5-975*x^4+28000*x^3-345000*x^2+1800000*x);
```
Solution using MATLAB

Open *Optimization Toolbox* by typing `optimtool` command

- **Solver:** Select `simulannealbnd - Simulated Annealing algorithm` solver

- **Objective function:** Select input (objective function) file name as `@simple_objective`
Solution using MATLAB

- **Start Point:** feed random start point (e.g. 15.5)

- **Bounds**
  - **Lower Bound:** feed lower bound valve (e.g. 0)
  - **Upper Bound:** feed upper bound valve (e.g. 31)

- **Options:** in options panel we can choose various attributes (stopping criteria, annealing parameters, acceptance criteria and plot functions etc.)

- **Run Solver:** run the SA solver using *start* button
Solution using MATLAB
Conclusions

• SA is based on neighbourhood search and allows uphill moves.

• It has a strong analogy to the simulation of cooling of material.

• Uphill moves are allowed with a temperature dependent probability.

• Generic and problem-specific decisions have to be taken at implementation.

• Experimental tuning is very important!
Distance: 43,499 miles
Temperature: 1,316
Iterations: 0
Thank You!
Genetic Algorithm

Dr. Rajesh Kumar

PhD, PDF (NUS, Singapore)
SMIEEE, FIETE, MIE (I),LMCSI, SMIAACSIT, LMISTE, MIAENG

Associate Professor, Department of Electrical Engineering
Adjunct Associate Professor, Centre of Energy

Malaviya National Institute of Technology, Jaipur, India, 302017
Tel: (91) 9549654481
rkumar@ieee.org, rkumar.ee@gmail.com
http://drrajeshkumar.wordpress.com/
Maximum and minimum of a smooth function is reached at a stationary point where its gradient vanishes.
Hill-climbing

Calculus based approach
But...
But...
More realistically?

• Noisy?

• Discontinuous?
It can live up to 70 years

But to reach this age, the eagle must make a hard decision
Scientists have tried to mimic the nature throughout the history.
The nontraditional optimization algorithms are

- Genetic Algorithms
- Neural Networks
- Ant Algorithms
- Simulated Annealing
Classes of Search Techniques

- Calculus-based techniques
  - Direct methods
  - Indirect methods
- Guided random search techniques
  - Evolutionary algorithms
  - Simulated annealing
- Enumerative techniques
  - Dynamic programming
- Search techniques
  - Evolutionary strategies
  - Genetic algorithms
    - Parallel
      - Centralized
      - Distributed
    - Sequential
      - Steady-state
      - Generational
- Fibonacci
- Newton
General Introduction to GA’s

• Genetic algorithms (GA’s) are a technique to solve problems which need optimization

• GA’s are a subclass of Evolutionary Computing

• GA’s are based on Darwin’s theory of evolution

• History of GA’s
  • Evolutionary computing evolved in the 1960’s.
  • GA’s were created by John Holland in the mid-70’s.
Biological Background (1) – The cell

- Every animal cell is a complex of many small “factories” working together
- The center of this all is the cell nucleus
- The nucleus contains the genetic information
Biological Background (2) – Chromosomes

- Genetic information is stored in the chromosomes
- Each chromosome is build of DNA
- Chromosomes in humans form pairs
- There are 23 pairs
- The chromosome is divided in parts: genes
- Genes code for properties
- The possibilities of the genes for one property is called: allele
- Every gene has an unique position on the chromosome: locus
Biological Background (3) – Genetics

- The entire combination of genes is called genotype.
- A genotype develops to a phenotype.
- Alleles can be either dominant or recessive.
- Dominant alleles will always express from the genotype to the phenotype.
- Recessive alleles can survive in the population for many generations, without being expressed.
Biological Background (4) – Reproduction

- During reproduction “errors” occur
- Due to these “errors” genetic variation exists
- Most important “errors” are:
  - Recombination (cross-over)
  - Mutation
Biological Background (5) – Natural selection

- The origin of species: “Preservation of favourable variations and rejection of unfavourable variations.”

- There are more individuals born than can survive, so there is a continuous struggle for life.

- Individuals with an advantage have a greater chance for survive: survival of the fittest.
Biological Background (6) – Natural selection

• Important aspects in natural selection are:
  • adaptation to the environment
  • isolation of populations in different groups which cannot mutually mate

• If small changes in the genotypes of individuals are expressed easily, especially in small populations, we speak of genetic drift

• Mathematical expresses as fitness: success in life
Genetic Algorithms

Notion of Genetic Algorithms

Genotype space = \{0, 1\}^L

Phenotype space

Encoding (representation)

Decoding (inverse representation)

10010001
10010010
010001001
011101001
Genetic Algorithms

Notion of Genetic Algorithms

Human

1 0 1 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 0

23 chromosomes

A fetus is formed by a Male(sperm) and female(egg).
\[ \begin{array}{c c c c c c c c}
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
+ & & & & & & & & \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0
\end{array} \]

\[ \begin{array}{c c c c c c c c}
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0
\end{array} \]
The evolutionary process can be expedited by improving the variety of the gene pool.

It is done via mutation.

Mutation Process
Simple Genetic Algorithm

{
initialize population;
evaluate population;
while Termination CriteriaNotSatisfied
{
select parents for reproduction;
perform recombination and mutation;
evaluate population;
}
}
Genetic algorithms are usually applied for maximization problems.

To minimize $f(x)$ ($f(x) > 0$) using GAs, consider maximization of

$$\frac{1}{1 + f(x)}$$
Example

Maximize $y^{1.3} + 10e^{-xy} + \sin (x-y) + 3$ in $R = [0,14] \times [0,7]$.

\[(0,5)=000101\]
\[(2,7)=001111\]
\[(6,1)=011001\]
\[(10,4)=101100\]
\[(12,0)=110000\]
\[(8,6)=100110\]
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
</tr>
</tbody>
</table>

Fitness:

\[
\begin{align*}
21.02 + 15.46 + 4.11 + 9.17 + 13.21 + 13.31 &= 0.276
\end{align*}
\]
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td></td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td></td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td></td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td></td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td></td>
</tr>
</tbody>
</table>

Fitness:

\[
\frac{21.02 + 15.46 + 4.11 + 9.17 + 13.21 + 13.31}{0.276} = 0.276
\]
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
</tr>
<tr>
<td>String</td>
<td>(x,y)</td>
<td>f(x,y)</td>
<td>Prob.</td>
<td>Cum.Prob</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
</tr>
<tr>
<td>String</td>
<td>(x, y)</td>
<td>f(x, y)</td>
<td>Prob.</td>
<td>Cum. Prob</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>000101</td>
<td>(0, 5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
</tr>
<tr>
<td>001111</td>
<td>(2, 7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
</tr>
<tr>
<td>011001</td>
<td>(6, 1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
</tr>
<tr>
<td>101100</td>
<td>(10, 4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
</tr>
<tr>
<td>110000</td>
<td>(12, 0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
</tr>
<tr>
<td>100110</td>
<td>(8, 6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
</tr>
</tbody>
</table>

0.269
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td></td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td></td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td></td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td></td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

AOT'15, MNIT, Jaipur, 18-22 May 2015
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td></td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td></td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td></td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>(x,y)</td>
<td>f(x,y)</td>
<td>Prob.</td>
<td>Cum.Prob</td>
<td>String</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td></td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td></td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td></td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

0.923
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td></td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td></td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>(x,y)</td>
<td>f(x,y)</td>
<td>Prob.</td>
<td>Cum.Prob</td>
<td>String</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

AOT'15, MNIT, Jaipur, 18-22 May 2015
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

Crossover: 0.707
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

**Crossover**
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

**Crossover**

String 000101, 100110
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

Crossover

AOT'15, MNIT, Jaipur, 18-22 May 2015
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

Crossover

AOT'15, MNIT, Jaipur, 18-22 May 2015
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum.Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

**Crossover:**
- 000101
- 100110
- 000101
- 001111
- 110000
- 101100

**Mutation:**
- 000101
- 100110
- 000111
- 001101
- 111100
- 100000

Changes if p>0.95
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
<th>Prob.</th>
<th>Cum. Prob</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
<td>100110</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
<td>000101</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
<td>101100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>Crossover</th>
<th>String</th>
<th>Mutation</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td></td>
<td>000110</td>
<td>000111</td>
<td>000110</td>
</tr>
<tr>
<td>100110</td>
<td>Crossover</td>
<td>100101</td>
<td>001101</td>
<td>100101</td>
</tr>
<tr>
<td>000101</td>
<td></td>
<td>000111</td>
<td>000111</td>
<td>000111</td>
</tr>
<tr>
<td>001111</td>
<td></td>
<td>001101</td>
<td>001101</td>
<td>001101</td>
</tr>
<tr>
<td>110000</td>
<td></td>
<td>111100</td>
<td>111100</td>
<td>111100</td>
</tr>
<tr>
<td>101100</td>
<td></td>
<td>100000</td>
<td>100000</td>
<td>100000</td>
</tr>
<tr>
<td>String</td>
<td>(x,y)</td>
<td>f(x,y)</td>
<td>Prob.</td>
<td>Cum.Prob</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>000101</td>
<td>(0,5)</td>
<td>21.02</td>
<td>0.276</td>
<td>0.276</td>
</tr>
<tr>
<td>001111</td>
<td>(2,7)</td>
<td>15.46</td>
<td>0.203</td>
<td>0.479</td>
</tr>
<tr>
<td>011001</td>
<td>(6,1)</td>
<td>4.11</td>
<td>0.054</td>
<td>0.533</td>
</tr>
<tr>
<td>101100</td>
<td>(10,4)</td>
<td>9.17</td>
<td>0.120</td>
<td>0.653</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
<td>0.173</td>
<td>0.826</td>
</tr>
<tr>
<td>100110</td>
<td>(8,6)</td>
<td>13.31</td>
<td>0.174</td>
<td>1.000</td>
</tr>
</tbody>
</table>

### Crossover

<table>
<thead>
<tr>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
</tr>
<tr>
<td>100110</td>
</tr>
<tr>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
</tr>
<tr>
<td>101100</td>
</tr>
</tbody>
</table>

**Crossover**

<table>
<thead>
<tr>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000110</td>
</tr>
<tr>
<td>100101</td>
</tr>
<tr>
<td>000111</td>
</tr>
<tr>
<td>001101</td>
</tr>
<tr>
<td>111100</td>
</tr>
<tr>
<td>100000</td>
</tr>
</tbody>
</table>

### Mutation

<table>
<thead>
<tr>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000110</td>
</tr>
<tr>
<td>001101</td>
</tr>
<tr>
<td>000111</td>
</tr>
<tr>
<td>001101</td>
</tr>
<tr>
<td>111100</td>
</tr>
<tr>
<td>100000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000110</td>
</tr>
<tr>
<td>001101</td>
</tr>
<tr>
<td>000111</td>
</tr>
<tr>
<td>001101</td>
</tr>
<tr>
<td>111100</td>
</tr>
<tr>
<td>100000</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
</tr>
<tr>
<td>011001</td>
</tr>
<tr>
<td>101100</td>
</tr>
<tr>
<td>110000</td>
</tr>
<tr>
<td>100110</td>
</tr>
</tbody>
</table>

**Crossover**

<table>
<thead>
<tr>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
</tr>
<tr>
<td>100110</td>
</tr>
<tr>
<td>000101</td>
</tr>
<tr>
<td>001111</td>
</tr>
<tr>
<td>110000</td>
</tr>
<tr>
<td>101100</td>
</tr>
</tbody>
</table>

**Mutation**

<table>
<thead>
<tr>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>000110</td>
</tr>
<tr>
<td>100101</td>
</tr>
<tr>
<td>000111</td>
</tr>
<tr>
<td>001101</td>
</tr>
<tr>
<td>111100</td>
</tr>
<tr>
<td>100000</td>
</tr>
</tbody>
</table>

Go to iter. 2
<table>
<thead>
<tr>
<th>String</th>
<th>(x,y)</th>
<th>f(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000110</td>
<td>(0,6)</td>
<td>23.17</td>
</tr>
<tr>
<td>001101</td>
<td>(2,5)</td>
<td>11.05</td>
</tr>
<tr>
<td>000111</td>
<td>(0,7)</td>
<td>25.43</td>
</tr>
<tr>
<td>001101</td>
<td>(2,5)</td>
<td>11.05</td>
</tr>
<tr>
<td>111100</td>
<td>(14,4)</td>
<td>9.24</td>
</tr>
<tr>
<td>110000</td>
<td>(12,0)</td>
<td>13.21</td>
</tr>
</tbody>
</table>

Previous Avg. = 12.71
New Avg. = 15.52
Previous Max. = 21.02
New Max. = 25.43
TSP Example: 30 Cities
Solution \( j(\text{Distance} = 800) \)
Best Solution (Distance = 420)
Benefits of Genetic Algorithms

- Concept is easy to understand
- Modular, separate from application
- Supports multi-objective optimization
- Good for “noisy” environments
- Always an answer; answer gets better with time
- Inherently parallel; easily distributed
Benefits of Genetic Algorithms (cont.)

- Many ways to speed up and improve a GA-based application as knowledge about problem domain is gained
- Easy to exploit previous or alternate solutions
- Flexible building blocks for hybrid applications
- Substantial history and range of use
When to Use a GA

- Alternate solutions are too slow or overly complicated
- Need an exploratory tool to examine new approaches
- Problem is similar to one that has already been successfully solved by using a GA
- Want to hybridize with an existing solution
- Benefits of the GA technology meet key problem requirements
# Some GA Application Types

<table>
<thead>
<tr>
<th>Domain</th>
<th>Application Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>gas pipeline, pole balancing, missile evasion, pursuit</td>
</tr>
<tr>
<td>Design</td>
<td>semiconductor layout, aircraft design, keyboard configuration, communication networks</td>
</tr>
<tr>
<td>Scheduling</td>
<td>manufacturing, facility scheduling, resource allocation</td>
</tr>
<tr>
<td>Robotics</td>
<td>trajectory planning</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>designing neural networks, improving classification algorithms, classifier systems</td>
</tr>
<tr>
<td>Signal Processing</td>
<td>filter design</td>
</tr>
<tr>
<td>Game Playing</td>
<td>poker, checkers, prisoner’s dilemma</td>
</tr>
<tr>
<td>Combinatorial Optimization</td>
<td>set covering, travelling salesman, routing, bin packing, graph colouring and partitioning</td>
</tr>
</tbody>
</table>
WRITING A FUNCTION

- Select **New** in the MATLAB File menu.
- Select **M-File**. This opens a new M-file in the editor.
- In the M-file, enter the following two lines of code:
  
  ```matlab
  function z = my_fun(x)
  z = x(1)^2 - 2*x(1)*x(2) + 6*x(1) + x(2)^2 -6*x(2);
  ```

- Save the M-file in a directory on the MATLAB path.
A function handle is a MATLAB value that provides a means of calling a function indirectly.

The syntax is

\[ \text{handle} = @\text{function name} \]

For example

\[ h = @\text{my\_fun} \]
USING THE TOOLBOX

There are two ways to use genetic algorithm with the toolbox:

- Calling the genetic algorithm function `ga` at the command line
- Using the genetic algorithm tool, a graphical interface
The syntax is
\[ [x \ fval] = \text{ga}( \text{@fitnessfun}, \ nvars, \ options) \]

Where

- \text{@fitnessfun} is a function handle to the fitness function
- \text{nvars} is the number of independent variables for the fitness function
- \text{options} is a structure containing options for the genetic algorithm
- \text{fval} is the final value of the fitness function
- \text{x} is the point at which final value is attained
USING GENETIC ALGORITHM TOOL

- To open we have to write following syntax at the command line

```bash
gatool
```
PAUSING AND STOPPING

- Click *Pause* to temporarily suspend the algorithm. To resume the algorithm using the current population at the time you paused, click *Resume*.

- Click *Stop* to stop the algorithm. The Status and results pane displays the fitness function value of the best point in the current generation at the moment you clicked *Stop*. 
SETTING STOPPING CRITERIA

- **Generations** -- The algorithm reaches the specified number of generations.
- **Time** -- The algorithm runs for the specified amount of time in seconds.
- **Fitness limit** -- The best fitness value in the current generation is less than or equal to the specified value.
- **Stall generations** -- The algorithm computes the specified number of generations with no improvement in the fitness function.
- **Stall time limit** -- The algorithm runs for the specified amount of time in seconds with no improvement in the fitness function.
The Plots pane, shown in the following figure, enables you to display various plots of the results of the genetic algorithm.
RESULTS

- For the function we wanted to minimise i.e $my\_fun$

\[ z = x(1)^2 - 2*x(1)*x(2) + 6*x(1) + x(2)^2 - 6*x(2); \]

The final result that will be obtained after using genetic algorithm will be

\[ x = [ 0.25836 \ 3.26145] \]
Control of Induction Motor Drives

- Optimization of both the structure and the associated parameters of four cascaded controllers for an IM drive. The GA tests and compares controllers having different orders. The GA optimizes both the number and the location of controller’s zeros and poles.

- While conventional design strategies firstly tune the current controller and then the speed controller (thus leading to a potentially sub-optimal couple of controllers), the GA simultaneously optimizes the two controllers, to obtain the best overall two cascaded-loops control systems.

- We developed a system based on MATLAB/SIMULINK environment and dSPACE control boards to run the optimization on-line, directly on the hardware.
Induction motor drive block diagram
Crossover, Mutation, and Selection

- **Crossover**
  - Binary flags: multi-point crossover
  - Real valued chromosomes: either heuristic or arithmetical crossover is applied with equal probability.

- **Mutation**
  - Binary flags: binary mutation
  - Real valued chromosomes: uniform mutation

- **Selection**
  - Tournament selection
The ability to work on heterogeneous solutions makes the considered GA similar to other evolutionary computation techniques, such as Genetic or Evolutionary Programming (GP or EP).

A significant difference between EP and GAs lays in the rate of application of the mutation.

In the preliminary investigation for optimal occurrence rate of operators, a much faster convergence was obtained with higher mutation rates.

Due to our final choice of emphasizing the rate of mutation (80%) with respect to crossover (20%), the algorithm can be viewed as a hybrid GA-EP evolutionary algorithm.
Choice of the fitness function

- The objective function is a weighted sum of five performance indices that are directly measured on system’s response to the a sequence of speed steps of different amplitude.

\[
\sum_{j=1}^{f} \alpha_j \cdot f_j
\]
Overview of the single indexes

- **Steady state speed response**

\[ f_1 = \sum_{j=1}^{n} \left| \omega_r(j) - \omega_r^{ref}(j) \right| \cdot g(j) \]

This index measures the speed error along segments of the speed response settling around the steady state.
Overview of the single indexes

- **Speed overshoot**

\[ f_2 = \frac{\omega_{r,\text{max}} - \omega_{r,\text{steady}}}{\omega_{r,\text{steady}}} \]

This index measures the speed overshoot when a speed step change is required.
Overview of the single indexes

- **Transient speed response duration**

\[ f_3 = \frac{n - n_t}{n} \]

This index measures the duration of transient conditions, that is the sum of all the settling times.
Overview of the single indexes

- Voltage reference oscillations for constant speed reference

\[
f_4 = \sum_{j=1}^{n} \left| v_a^{\text{ref}}(j) - v_a^{\text{mean}}(j) \right| \cdot g(j)
\]

This index accounts for undesired ripples and oscillations of the voltage that would increase losses and acoustic noise, contrasting with constant torque operations.
Overview of the single indexes

- **Current response**

\[ f_5 = \sum_{j=1}^{n} \left| i_a(j) - i_a^{ref}(j) \right| \]

This index measures the sum of absolute current errors
Experimental Setup

- Induction Motor
- PWM Inverter
- Current & Voltage Sensor
- PC with ds1104 Controller
- Load Bank

AOT'15, MNIT, Jaipur, 18-22 May 2015
On-line optimization with standard GA: Induction Motor Drive

Speed response at the end of the GA optimization
On-line optimization with standard GA: Induction Motor Drive

- Rotor flux response at the end of the GA optimization
Question: ‘If GAs are so smart, why ain’t they rich?’

Answer: ‘Genetic algorithms are rich - rich in application across a large and growing number of disciplines.’

- David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning
“Biogeography-Based Optimization”

Dr. Ajay Kumar Bansal

Director
Poornima Institute of Engineering & Technology,
Jaipur
Outline

1. Biogeography
2. Biogeography-Based Optimization
3. BBO algorithm
4. BBO Flow chart
5. Migration
6. Mutation
7. Comparison between BBO, PSO and GA
8. Benchmark Function and Result
Biogeography

The study of the geographic distribution of biological organisms

- Mauritius
- 1600s
Biogeography- cont....

- The biogeography concept specifies that the species migration from one island to another depending on the various factors.
- It also analyses that how species arise or become extinct.
- In biogeography, habitats are ecological areas which are geographically not directly connected to other areas and populated by particular plant or animal species.
Biogeography- cont....

- Animals and plants naturally arrange in a certain region or area.
- Major isolated land areas and islands often evolve their own distinct plants and animals.
- Distribution can change seasonally in response to the availability of resources, it does not always stay the same.
A population is a group of individuals of the same species living in an area.

Population ecology focuses on factors affecting how many individuals of a species live in an area.

Ecologists have long recognized global and regional patterns of distribution of organisms within the biosphere.

Biogeography is a good starting point for understanding what limits geographic distribution of species.
Biogeography- cont....

- Depending on the climate and environment of the different islands animals found a way to change for the better to live more properly.
- No two species can acquire the same niche; eventually one will become better adapted threw competition for food, shelter exc.
- The other will die out or be forced to change environments.
Biogeography- cont....

- Emigration occurs because of accumulation of random effects on a large number of species with large populations.
- When species emigrates from an island, it does not completely disappears; only a few representatives emigrate.
- In BBO it is assumed that emigration from an island results in extinction from that island.
- Species represent the independent variables of a function, and each island represents a candidate solution.
Biogeography- cont....

- Habitat suitability index (HSI) is the parameter that defines each habitat.
- A habitat is probable solution of the problem.
- The features that determine habitability are called suitability index variables (SIVs).
- Islands that are friendly to life are said to have a high habitat suitability index (HSI).
- SIVs are the independent variables and HSI is the dependent variable.
Species migrate between “islands” via flotsam, wind, flying, swimming, …
Biogeography- cont....

- Due to limited resources, populations may be evenly distributed to minimize competition, as is found in forests, where competition for sunlight produces an even distribution of trees.
- The more that the environment changes, the more the species change to adapt to their habitat and surroundings.
Island with high population
No more space to live
Species can leave

Species can come or leave

Island with low population
Some space left for new species

Island with zero population
Large space to live
No Species can leave

Large number of species can come

Biogeography-Based Optimization

AOT'15, MNIT, Jaipur, 18-22 May 2015
• Islands with high HSI support many species
• Islands with low HSI support only few species.
• Islands with high HSI have many species that *emigrate* to nearby habitats because of the large populations and the large numbers of species that they host.
• Emigration from an island with a high HSI does not occur because species *want* to leave their home
Biogeography- cont....

- Islands with a high HSI
  - have a high emigration rate
  - low immigration rate.

- Islands with a low HSI
  - high immigration rate because of their low populations.

- However, **species diversity** is correlated with HSI, so when more species arrive at a low HSI island, the island's HSI will tend to increase.
Biogeography- cont...

![Graph showing immigration and emigration rates versus number of species.]
Biogeography- cont...

As habitat suitability improves:

- The species count increases.
- Emigration increases (more species leave the habitat).
- Immigration decreases (fewer species enter the habitat).

- \( I \) is the maximum possible immigration rate.
- \( E \) is the maximum possible emigration rate.
- Maximum possible species in a habitat is \( S_{\text{max}} \).
- Condition where emigration rate \( \mu \) is equal to immigration rate \( \lambda \) is \( S_0 \).
Biogeography- cont...

- $I$ occurs when island is empty of any species and offer maximum opportunity to species on other islands for immigrating to settle on it; and as number of arrived species on that island increases, opportunity for settlement will decrease and thus immigration rate will decrease too.

- As $\lambda$ decreases, species density increases, so predation, competition and parasitism factors will increase; and as a result, emigration rate $\mu$ will increase, and reaches its maximum value $E$ when $\lambda$ reaches its minimum value.
Biogeography-Based Optimization

\[ \lambda = \text{the probability that the immigrating individual's solution feature is replaced} \]

\[ \mu = \text{the probability that an emigrating individual's solution feature migrates to the immigrating individual} \]
Biogeography-Based Optimization - cont...

\[ P_s = \text{probability that habitat contains } S \text{ species} \]

\[ P_s(t + \Delta t) = P_s(t)(1 - \lambda_s \Delta t - \mu_s \Delta t) + \\
\left( P_{s-1}(t) \lambda_{s-1} \Delta t + P_{s+1}(t) \mu_{s+1} \Delta t \right) \]

\[ \begin{align*}
S-1 \text{ species at time } t, \text{ and } \\
1 \text{ species immigrated}
\end{align*} \]

\[ \begin{align*}
S+1 \text{ species at time } t, \text{ and } \\
1 \text{ species emmigrated}
\end{align*} \]
The time $\Delta t$ must be small enough so that only one species will immigrate or emigrate. For small time, limit is taken as zero ($\Delta t \rightarrow 0$) and reduces to

$$
P_s = \begin{cases} 
-(\lambda^s + \mu^s) P_s + \mu^{s+1} P_{s+1} & S = 0 \\
-(\lambda^s + \mu^s) P_s + \mu^{s+1} P_{s+1} + \lambda^{s-1} P_{s-1} & 1 \leq S \leq S_{\max} - 1 \\
-(\lambda^s + \mu^s) P_s + \lambda^{s-1} P_{s-1} & S = S_{\max}
\end{cases}
$$

A linear and constant relationship is considered among habitats HSI, immigration and emigration rate

$$
\lambda_k = EK/P \\
\mu_k = I(1 - K/P)
$$

where $K$ represents total species of $k^{th}$ individual, $P$ represents total species, $E$ is maximum emigration rate and $I$ is maximum immigration rate.
BBO Algorithm

1. Define the mutation probability and the elitism parameter.
2. Initialize the population.
3. Calculate immigration rate and emigration rate for each island. Good solutions have high emigration rates and low immigration rates. Bad solutions have low emigration rates and high immigration rates.
4. Probabilistically choose the immigrating islands based on the immigration rates. Use roulette wheel selection based on the emigration rates to select the emigrating islands.
5. Migrate randomly selected SIVs (i.e., independent solution variables) based on the selected islands in the previous step.
6. Probabilistically perform mutation for each island.
7. Replace the worst islands in the population with the previous generation’s elite islands.
8. If the termination criterion is met, terminate; otherwise, go to step 3.
Biogeography-Based Optimization

BBO flow chart

Start

1. Generate Initial population
2. Calculate fitness of each habitat
3. Perform elitism by replacing bottom habitat
4. Evaluate each individual fitness
5. Calculate \( \lambda \) and \( \mu \) based on fitness value
6. Habitat selected on the basis of \( \mu \)?
   - Yes: Migration operation
   - No: Go to next step
7. Habitat selected on the basis of probability?
   - Yes: Mutation operation
   - No: Stop?
8. Result
9. Best Solution

AOT'15, MNIT, Jaipur, 18-22 May 2015
Migration

- Using emigration and immigration rates of each habitat, BBO probabilistically share the information between habitats.

- If any habitat is selected for modification, based on immigration rate each SIV in that habitat is modified probabilistically.

- If a SIV in the selected habitat is selected for modification, based on other habitats emigration rates the randomly selected SIV is migrate to the selected SIV probabilistically.
Migration

Select $H_i$ with probability $\alpha \lambda_i$
If $H_i$ is selected
  For $j = 1$ to $n$
    Select $H_j$ with probability $\alpha \mu_i$
    If $H_j$ is selected
      Randomly select an SIV $\sigma$ from $H_j$
      Replace a random SIV in $H_i$ with $\sigma$
  end
end
end
Mutation

- As nature’s habitat’s HSI can change suddenly due to apparently random events.
- Using species probabilities, mutation rates are calculated.
- Using mutation rates, low HSI as well as high HSI habitats are randomly modified.

For $j = 1$ to $m$

Use $\lambda_i$ and $\mu_i$ to compute the probability $P_i$.

Select SIV $H_i(j)$ with probability $\alpha P_i$.

If $H_i(j)$ is selected

Replace $H_i(j)$ with a randomly generated SIV.

end

end
# Comparison between BBO, PSO and GA

<table>
<thead>
<tr>
<th></th>
<th>BBO</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Develop</strong></td>
<td>2008</td>
<td>1995</td>
<td>Early 1970s</td>
</tr>
<tr>
<td><strong>Introduced by</strong></td>
<td>Dan Simon</td>
<td>J. Kennedy and R. Eberhart</td>
<td>John Henry Holland</td>
</tr>
<tr>
<td><strong>Optimization Technique (OT)</strong></td>
<td>Bio-inspired OT that used idea of probabilistically sharing features between solutions based on the solutions’ fitness values.</td>
<td>Robust Stochastic OT based on the movement and cooperation.</td>
<td>Intelligent OT that relies on the parallelism found in nature.</td>
</tr>
<tr>
<td><strong>End of generation</strong></td>
<td>BBO solutions survive forever.</td>
<td>survive forever.</td>
<td>GA solutions die out.</td>
</tr>
</tbody>
</table>
## Comparison between BBO, PSO and GA

<table>
<thead>
<tr>
<th></th>
<th>BBO</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grouped</strong></td>
<td>Does not form grouping of habitat having identical characteristic.</td>
<td>Are grouped into their similar characteristic.</td>
<td>Not necessarily have any built-in tendency</td>
</tr>
<tr>
<td><strong>Mutation</strong></td>
<td>Combination of two ideas of global recombination and uniform crossover, which are use the entire of the population as potential contributors to the next generation and use fitness based selection for each solution feature in each offspring.</td>
<td>PSO does not have genetic operators such as crossover or mutation. Particle update themselves with the internal velocity, they also have a memory that is important to the algorithm.</td>
<td>All GAs require some form of recombination, as this allows the creation of new solutions that have, by virtue of their parent success, a higher probability of exhibiting a good performance.</td>
</tr>
</tbody>
</table>
Benchmark Function

Rastrigin: \[ f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)] \]

nonseparable, regular, multimodal
Rastrigin function optimization result

$n=20$

Best chromosome = 52 52 52 52 52 52 52 52 52 52 52 52 52 52 52 52 52 52 52

Minimum cost = 0
Rosenbrock:  
\[ f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \]

nonseparable, regular, unimodal
Rosenbrock function optimization result
$n=20$

Best chromosome = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Minimum cost = 0
Reference


Thanks & Query
Short Term Course

Advanced Optimization Techniques

Day 3 : (20-May-2015)
Differential Evolution

Dr. Prerna Jain
Assistant Professor
Department of Electrical Engineering
Malviya National Institute of Technology Jaipur
Email: pjain.ee@mnit.ac.in, prerna7412@gmail.com.
Optimization

- Finding one or more feasible solutions
- Corresponds to extreme values of one or more objectives

Simple Definition:

“Consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set & computing the value of the function”

“More generally, finding "best available" values of some objective function given a defined domain (or a set of constraints), including a variety of different types of objective functions and different types of domains”
• Optimization is an extreme objective (minimization or maximization)

• Such extreme property has led into its great importance in engineering design, scientific experiments and business decision making

Rationale

• Classical optimization techniques
  - Weighted Sum Method
  - ε- Constraint Method
  - Weighted Metric Method
  - Benson’s Method
  - Value Function Method
  - Goal programming Methods, etc.

Direct Methods & Gradient methods
• **Shortcomings**
  - Use deterministic procedure
  - Direct search methods are slow
  - Gradient search methods not efficient in non differentiable, discontinuous, mixed real/integer problems
  - Large number variables
  - Mutiminima
  - Multiobjective
  - Suboptimal solution: No Global Optimization
  - Specific problems
  - No parallel computing
  - Constraint handling
  - ...........

  Not Suitable for many real world practical problems

Evolutionary Algorithms (Eas) mimic natural evolutionary principles to constitute search & stochastic optimization procedure

Alleviate many of these shortcomings
Ability of global optimization

Different Evolution is one such powerful technique
Historical Notes

Differential Evolution (DE): What and When


![Differential Evolution Diagram]
DE Basics

- Simple: Minimum control parameters
- Stochastic: Random
- Population based: Parallel computing
- Developed to optimize real parameter, real valued functions
- Genetic type of algorithm
- Fast
- Extremely well on a wide variety of problems
- General problem formulation is:

  For an objective function \( f : X \subseteq \mathbb{R}^D \rightarrow \mathbb{R} \) where the feasible region \( X \neq \emptyset \), the minimization problem is to find \( x^* \in X \) such that \( f(x^*) \leq f(x) \forall x \in X \)

  where:

  \[ f(x^*) \neq -\infty \]
DE Basics

- Algorithm: Typical Evolutionary Scheme

- Initialization of the population of candidate solutions
- Mutation: Expands the search space
- Recombination/Crossover: Reuses previously successful individuals
- Selection(Explicit): Mimics survival of the fittest
- Convergence: Predefined number of iteration
Problem Formulation

- Consider a system with real valued properties as objectives of the system to be optimized
  \[ g_m; m = 0, 1, 2, \ldots, P - 1 \]
- Real valued constraints
  \[ g_m; m = P, P + 1, \ldots, P + Q - 1 \]
- Properties: Need not to optimized but neither shall be degraded
- All the properties be dependent on the real valued parameters
  \[ x_j; j = 0, 1, \ldots, D - 1 \]
- Bounds of variables
  \[ x \in [x_{jl}, x_{jh}] \]
- Usually, incorporated into the \( g_m \) constraints
- Optimization of the system is to vary the D-dimensional parameter vector
  \[ \bar{x} = (x_0, x_1, \ldots, x_{D-1})^T \]
- Such that \( g_m \) objectives are optimized and \( g_m \) constraints are met
Problem Formulation

- Multiple objectives along with constraints are converted to single function usually via the weighted sum

\[ z(x) = \sum_{m=0}^{P+C-1} w_m \cdot g_m(x) \]

- And task is now

\[ \min z(x) \]
**DE Approach**

**Initialization**

- Parallel direct search method
- NP parameter vectors (Population members)

\[ x_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}] \]

- \( G \) is the generation number
- NP does not change during all the generations

- **Initialization**: Randomly select the initial parameter values uniformly on the intervals while satisfying other equality and inequality constraints for all the population members

\[ x_j \in [x_{jl}, x_{jh}] \]
DE Approach.....

Mutation

- Crucial idea behind DE is a new mutation scheme for generating trial parameter vectors
- Add difference vectors to base individual to produce mutant vector for each target vector in the current population in order to explore the search space
Mutation

- For each target vector
  \[ x_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}] \quad i = 1, 2, 3, \ldots, NP \]

- Associated Mutant vector
  \[ v_{i,G} = [v_{1,i,G}, v_{2,i,G}, \ldots, v_{D,i,G}] \quad i = 1, 2, 3, \ldots, NP \]

- Different Mutation Strategies

1. “DE/rand/1”
   \[ v_{i,G} = x_{r1,G} + F(x_{r1,G} - x_{r3,G}) \]

2. “DE/best/1”
   \[ v_{i,G} = x_{best,G} + F(x_{r1,G} - x_{r2,G}) \]

3. “DE/rand-to-best/1”
   \[ v_{i,G} = x_{r1,G} + F(x_{best,G} - x_{i,G}) + F(x_{r1,G} - x_{r2,G}) \]

4. “DE/best/2”:
   \[ v_{i,G} = x_{best,G} + F(x_{r1,G} - x_{r2,G}) + F(x_{r3,G} - x_{r4,G}) \]

5. “DE/rand/2”:
   \[ v_{i,G} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) + F(x_{r4,G} - x_{r5,G}) \]
Mutation..

- Indices $r_1, r_2, r_3, r_4, r_5$ are mutually exclusive integers randomly generated within the range $[1, NP]$ & different from $I$
- ‘F’ is scaling parameter: Real and constant factor which controls the amplification
- $x_{best,G}$ is the best individual vector at generation $G$
Recombination/Crossover Operation

- Applied to each pair of the target vector & its corresponding mutant vector

\[ x_{i,G} = [ x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G} ] \quad i = 1, 2, 3, \ldots, NP \]

\[ v_{i,G} = [ v_{1,i,G}, v_{2,i,G}, \ldots, v_{D,i,G} ] \quad i = 1, 2, 3, \ldots, NP \]

- to generate a trial vector

\[ u_{i,G} = [ u_{1,i,G}, u_{2,i,G}, \ldots, u_{D,i,G} ] \quad i = 1, 2, 3, \ldots, NP \]

\[ u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand \ j (0,1) \leq CR \ or \ (j = j_{rand}) \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad j = 1, 2, \ldots, D \]

- ‘CR’ is the crossover rate ; a user specified constant
- Controls the fraction of parameter values copied from the mutant vector
- Condition \( j \neq j_{rand} \) ensures that the trial vector \( u_{i,G} \) will differ from its corresponding target vector by at least one parameter.
Selection

- If values of any parameter of trial vector exceed the corresponding upper & lower bounds, they are uniformly and randomly reinitialized.
- Evaluate Fitness (Value of the objective function) of all trial vectors.
- Selection among the set of trial and corresponding target vectors is carried out to choose the best, on the basis of respective objective function values.

\[ x_{i,G+1} = \begin{cases} 
  u_{i,G} & \text{if } f(u_{i,G}) \leq f(x_{i,G}) \\
  x_{i,G} & \text{otherwise} 
\end{cases} \]

- Greedy scheme is the key for fast convergence.

Convergence

These steps are repeated generation after generation until some specific termination criteria are satisfied.
Algorithm 1: DE Algorithm [64]

Generate parent population $Pop$
Evaluate fitness for each individual in $Pop$

\textbf{while} Termination criteria is not satisfied \textbf{do}

\textbf{for} $i = 1$ to $N_p$ \textbf{do}

Select uniform randomly $r1 \neq r2 \neq r3 \neq i$

$j_{rand} = \text{randint}(1, D)$

\textbf{for} $j = 1$ to $D$ \textbf{do}

\textbf{if} $\text{randreal}_{j}(0,1) > CR$ or $j = j_{rand}$ \textbf{do}

$$U_i(j) = X_i(j) + F \times (X_{r2}(j) - X_{r3}(j))$$

\textbf{else}

$$U_i(j) = X_i(j)$$

\textbf{end if}

\textbf{end for}

\textbf{end for}

\textbf{for} $i = 1$ to $N_p$ \textbf{do}

Evaluate offspring $U_i$

\textbf{if} $U_i$ is better than $Pop_i$ \textbf{then}

$$Pop_i = U_i$$

\textbf{end if}

\textbf{end for}

\textbf{end while}
Advice:

- NP = 5 or 10 times of number of parameter in a vector
- If solutions get stuck
- F = 0.5 and then increase F or NP
- F = [0.4, 1] is a very effective range
- CR = 0.9 or 1 for a quick solution
Advantages

- Immediately accessible for practical applications
- Simple structure
- Ease of use
- Seed to get the solutions
- Robustness
Discussions

- Performance highly depends on chosen trial vector generation strategy & Control parameters values (Two important shortcomings)
- Inappropriate values may lead to premature convergence
- DE researchers have suggested many empirical guidelines
- Conflicting conclusions confuse scientists and engineers
- Many modifications have been suggested to avoid manual tuning
  - Fuzzy Adaptive DE
    - Use fuzzy logic controllers whose input incorporate the relative function values and individuals of successive generation to adapt the parameters for the mutation and crossover operations
  - Adapted pareto DE
    - Based on the idea of controlling the population diversity & implemented a multi-population approach
  - Self adapted DE
    - Encodes F and CR into the chromosome and uses a self adaptive control mechanism to change them
Discussions......

- Hybridization with other techniques in one such direction
- Trigonometric mutation operation hybridized
- Combination of DE with Estimation of Distribution Algorithm (EDA)
  Uses a probability model to determine promising regions in order to focus the search process on those areas
- Two level orthogonal crossover
- Local search into classical DE
- Attraction repulsion type concept of electromagnetism like algorithm to boost the mutation operation
- Opposition based learning to generate initial population
Discussions

- Delivers better & faster solution using its different crossover strategies
- Has good exploration ability of search space due to its unique mutation & stochastic crossover
- Increasing system complexity & size retards its capability to map entire unknown variables together
- Initially solutions move very fast towards optimal point but fails to perform at later stages during fine tuning
- Crossover operation causes many solutions to lose their strength at a later stage, whose fitness was initially good.
- DE has good exploration ability to find the region of global optima
- Poor exploitation ability for global optimization (Another important problem)
Discussions....

- Many modifications on the basic DE have been proposed to remedy the situation
  - DE in combination with sequential quadratic programming (DEC-SQP)
  - Hybrid Differential Evolution (HDE)
  - Variable scaling Hybrid Differential Evolution (VSHDE)
  - Self-Tuning HDE (STHDE)
  - Hybrid Differential Evolution with Biogeography Based Optimization (DE/BBO)
  - etc.
DE/BBO

- **BBO**
  - BBO population based biogeography inspired global optimization algorithm
  - Based on natural immigration & emigration of species in search of more suitable habitat, evolution & extinction of species
  - Areas well suited have high Habitat Suitability Index (HSI)
  - Variables that characterize habitability are called Suitability Index Variables (SIV)
  - Migration & Mutation are its two operations
  - Especially poor solutions tend to accept more useful information from good solutions
  - Unique attribute of BBO
  - This makes BBO good at exploiting information of current population
  - Renders it good exploitation property for global optimization

- **DE/BBO**
  - Combines exploration of DE & exploitation of BBO
  - Migration operator of BBO along with mutation, crossover & selection operators of DE
Habitat Migration in BBO

Algorithm 2: Habitat Migration [64]

for $i = 1$ to $N_p$ do

Select habitat $H_i$ with probability proportional to $\lambda_i$

if randreal($0,1) < \lambda_i$ do

for $j = 1$ to $N_p$ do

Select another habitat $H_j$ with probability proportional to $\mu_j$

if randreal($0,1) < \mu_j$ then

Randomly select an SIV from the habitat $H_j$

Replace a random SIV in $H_i$ with that selected SIV of $H_j$

end if

end for

end if

end for
DE/BBO

Hybrid Migration Operator

- Hybrid migration operator is most important step in DE/BBO
- Combines DE’s mutation & crossover with migration operation of BBO
- In this operator, an offspring incorporates new features from population members
- Fundamental idea of such hybridization is based on two considerations
- Destruction of good solution is minimized
- Poor solution can accept new features from good solutions
- Current population can be exploited efficiently
- Search space exploration is effective through crossover and mutation operator of DE
Algorithm 3: Hybrid Migration Operator [64]

for \( i = 1 \) to \( N_p \) do

Select uniform randomly \( r_1 \neq r_2 \neq r_3 \neq i \)

\( j_{\text{rand}} = \text{randint}(1, D) \)

for \( j = 1 \) to \( D \) do

if \( \text{randreal}(0,1) < \lambda_i \) do

if \( \text{randreal}(0,1) > CR \) or \( j = j_{\text{rand}} \) do

\( U_i(j) = X_{r_1}(j) + F \times (X_{r_2}(j) - X_{r_3}(j)) \)

else

Select another habitat \( X_k \) with probability proportional to \( \mu_k \)

\( U_i(j) = X_k(j) \)

end if

else

\( U_i(j) = X_i(j) \)

end if

end for

end if
DE/BBO…

Main Procedure

- DE/BBO algorithm is developed by integrating hybrid migration operator into DE
- Comparison with DE, DE/BBO needs little extra computational cost in sorting the population & calculating the migration rates
- Needs lesser time than BBO
- DE/BBO structure is simple
- Able to explore new search space with mutation operator of DE & to exploit the population information with migration operator of BBO
- Overcomes weak exploitation property of original DE algorithm
- Comparison with BBO, it has enhanced exploration due to DE mutation operator
Algorithm 4: DE/BBO Algorithm [64]

Generate initial population $Pop$
Evaluate fitness for each individual in $Pop$

while Termination criteria is not satisfied do
    For each individual, map fitness to the number of species
    Calculate immigration rate $\lambda_i$ and emigration rate $\mu_i$ for each individual
    Modify the population with hybrid migration operation of Algorithm 3
    for $i = 1$ to $N_p$ do
        Evaluate offspring $U_i$
        if $U_i$ is better than $Pop_i$ then
            $Pop_i = U_i$
        end if
    end for
end while
Applications till date

- Economic Load Dispatch
- Economic Emission Dispatch
- Strategic Bidding of a GENCO in Electricity Market
Thank You Queries ?
Particle Swarm Optimisation (PSO)
(To Follow the leader)
Introduction

- PSO is a population-based stochastic optimization technique developed by Eberhart and Kennedy in 1995.
- It is inspired by social behavior of bird flocking or fish schooling.
- It is inherently designed for continuous optimization problems.
- It provides a platform to many swarm based optimization techniques such as, TLBO, CSO, BA, GSO, etc.
Position Update (Steered the Vehicle)

A swarm consists of number of particles “or possible solutions” that fly through the feasible solution space to explore optimal solutions. Each particle updates its position based on its own best exploration; best swarm overall experience, and its previous velocity vector according to the following model:

\[
v_j^{k+1} = Wv_j^k + C_1 \times \text{rand}_1() \times \frac{p\text{best}_j - s_j^k}{\Delta t} + C_2 \times \text{rand}_2() \times \frac{g\text{best}_j - s_j^k}{\Delta t}
\]

(1)

**First term:** Inertia component that provides momentum

**Second term:** Cognitive component that provides direction due to personal best experience

**Last term:** Social component that provides direction due to swarm’s best experience

\[
s_j^{k+1} = s_j^k + v_j^{k+1} \times \Delta t
\]

(2)
\[ W(k) = \left( \frac{W_{\text{min}} - W_{\text{max}}}{\text{iter}_{\text{max}} - 1} \right)(k - 1) + W_{\text{max}} \]
Advantages and Limitations

**Advantages:**
1. Simplicity: Single control equation
2. CPU time: Shorter
3. Implementation: Easy

**Limitations:**
1. Sensitive to parameter tuning
2. Weak communication
3. Local trapping
4. Suitable only for continuous optimization
Some Important Terms

- Problem search space
- Problem dimensions/ Decision variables
- Search space reduction
- Local trapping
- Convergence
- Diversity
- Accuracy
- Efficiency
- Global optima
- Local optima
- Exploration
- Exploitation
- Promising region
- Communication
- Computational burden
- Solution quality
- Best fitness
- Worst fitness
- Mean fitness
- Standard deviation
- Coefficient of variance
- Velocity regulation
Major difficulty with standard PSO

Velocity Regulation of particles

- The velocity of particles remains uncontrolled during the computational process.
- Particles approaches the promising region with relatively higher velocities and thus eventually miss the global/near global optima.
- This causes poor convergence due to local trapping and thus degrades solution quality.
Improving standard PSO

- The velocity of particles should be regulated in such a way that it maintains a proper balance between exploration and exploitation of the search space.

- All particles should start with suitable higher velocities and then they must be continuously retarded till maximum iterations exhausted.

However, this is a difficult task!!!
Possible solutions

- Inertia weight update: linear, exponential, sigmoid, chaotic sequence, truncated sinusoidal, etc.
- Dynamic control of acceleration coefficients: linear control, exponential control, etc.
- Velocity control: constriction factor approach, empirical formula, etc.
- Improving cognitive/social behavior: splitting cognitive component, splitting social behavior, etc.
- Search space reduction
- Increased communication
- Hybridized approaches
Sample Result

![Graph showing best fuel cost over iteration count for different methods.](image)
Selected References


Thanks & Any Queries
Ant Colony Optimization

Dr. Anil Swarnkar
Assistant Professor
Department of Electrical Engineering
Malaviya National Institute of Technology Jaipur
Email: aswarnkar.ee@mnit.ac.in
Artificial Intelligence Techniques

Guide an underlying search to escape from being trapped in a local optima and to explore better areas of the solution space.

- Single solution approaches, concerning only one solution at a particular time during the search. Search is very much restricted to local regions, so called local.
  - Simulated Annealing,
  - Tabu Search,
  - Variable neighbourhood search, etc.
Artificial Intelligence Techniques

- Population based approaches concern a population of solutions at a time
  - Genetic algorithm
  - Particle Swarm Optimisation
  - Ant Colony Optimisation
  - BAT Algorithm
  - Harmony Search
  - Teaching Learning Based Optimization
  - Cuckoo Search
Maxima and Minima

Diagram 1

Diagram 2
Rastrigin's Function

\[ R(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2) \]
How AI works
Initialization
Generation/Iteration 60
Generation/Iteration 80
Generation/Iteration 95
Generation/Iteration 100
Ant Colony Optimisation
The Ant Colony Optimization (ACO) was initially proposed by Marco Dorigo in 1992. The search technique is inspired by the behavior of ants in finding paths from the nest to food and back to the nest without any visual clues.

ACO studies artificial systems that take inspiration from the behaviour of real ant colonies and which are used to solve discrete optimization problems.
Characteristics of Ants

- Almost blind.
- Incapable of achieving complex tasks alone.
- Rely on the *phenomena* of swarm intelligence for survival.
- Capable of establishing shortest-route paths from their colony to feeding sources and back.
- Use *stigmergic* communication via pheromone trails.
"Stigmergic?"

- Stigmergy, a term coined by French biologist Pierre-Paul Grasse, is interaction through the environment.
- Two individuals interact indirectly when one of them modifies the environment and the other responds to the new environment at a later time. This is stigmergy.
Characteristics of Ants

- Follow existing pheromone trails with high probability.

- What emerges is a form of autocatalytic behavior: the more ants follow a trail, the more attractive that trail becomes for being followed.

- The process is thus characterized by a positive feedback loop, where the probability of a discrete path choice increases with the number of times the same path was chosen before.
Ant Intelligence

An example with real Ants
Ant Intelligence

- There is a path along which ants are walking (for example from food source A to the nest E, and vice versa.
- Suddenly an obstacle appears and the path is cut off. So at position B the ants walking from A to E (or at position D those walking in the opposite direction) have to decide whether to turn right or left.
- The choice is influenced by the intensity of the pheromone trails left by preceding ants.
- A higher level of pheromone on the right path gives an ant a stronger stimulus and thus a higher probability to turn right. The first ant reaching point B (or D) has the same probability to turn right or left (as there was no previous pheromone on the two alternative paths).
Ant Intelligence

- Because path BCD is shorter than BHD, the first ant following it will reach D before the first ant following path BHD.

- The result is that an ant returning from E to D will find a stronger trail on path DCB, caused by the half of all the ants that by chance decided to approach the obstacle via DCBA and by the already arrived ones coming via BCD: they will therefore prefer (in probability) path DCB to path DHB.

- As a consequence, the number of ants following path BCD per unit of time will be higher than the number of ants following EHD.

- This causes the quantity of pheromone on the shorter path to grow faster than on the longer one, and therefore the probability with which any single ant chooses the path to follow is quickly biased toward the shorter one.

- The final result is that very quickly all ants will choose the shorter path.
Difference between real ants and artificial ants

- artificial ants will have some memory,
- they will not be completely blind,
- they will live in an environment where time is discrete.
Suppose that the distances between D and H, between B and H, and between B and D (via C) are equal to 1, and let C be positioned half the way between D and B. 

High trail levels are synonymous with short paths.
Working of Artificial Ants

- Suppose that 30 new ants come to B from A, and 30 to D from E at each time unit, that each ant walks at a speed of 1 per time unit, and that while walking an ant lays down at time $t$ a pheromone trail of intensity 1, which, to make the example simpler, evaporates completely and instantaneously in the middle of the successive time interval $(t + 1, t + 2)$.

- At $t = 0$ there is no trail yet, but 30 ants are in B and 30 in D. Their choice about which way to go is completely random. Therefore, on the average 15 ants from each node will go toward H and 15 toward C.
Working of Artificial Ants

• At $t = 1$ the 30 new ants that come to B from A find a trail of intensity 15 on the path that leads to H, laid by the 15 ants that went that way from E, and a trail of intensity 30 on the path to C, obtained as the sum of the trail laid by the 15 ants that went that way from B and by the 15 ants that reached B coming from D via C.

• The probability of choosing a path is therefore biased, so that the expected number of ants going toward C will be the double of those going toward H: 20 versus 10 respectively. The same is true for the new 30 ants in D which came from E.

• This process continues until all of the ants will eventually choose the shortest path.
Application to Traveling Salesman Problem (TSP)

- Given a set of \( n \) towns, the TSP can be stated as the problem of finding a minimal length closed tour that visits each town once.
- We call \( d_{ij} \) the length of the path between towns \( i \) and \( j \).
- An instance of the TSP is given by a graph \((N, E)\), where \( N \) is the set of towns and \( E \) is the set of edges between towns.
Application to Traveling Salesman Problem (TSP)

Let $b_i(t)$ ($i = 1, \ldots, n$) be the number of ants in town $i$ at time $t$

$$m = \sum_{i=1}^{n} b_i(t)$$

and let be the total number of ants.

Each ant is a simple agent with the following characteristics:

- it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge;
- to force the ant to make legal tours, transitions to already visited towns are disallowed until a tour is completed (this is controlled by a tabu list);
- when it completes a tour, it lays a substance called trail on each edge $(i,j)$ visited.
Let $\tau_{ij}(t)$ be the intensity of trail on edge $(i, j)$ at time $t$.

Each ant at time $t$ chooses the next town, where it will be at time $t + 1$.

Therefore, if we call an iteration of the AS algorithm the $m$ moves carried out by the $m$ ants in the interval $(t, t + 1)$, then every $n$ iterations of the algorithm (which we call a cycle) each ant has completed a tour.

At this point the trail intensity is updated according to the following formula:

$$\tau_{ij}(t + n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}$$
where $\rho$ is a coefficient such that $(1 - \rho)$ represents the evaporation of trail between time $t$ and $t + n$, 

$$\Delta \tau_{ij} = \sum_{k=1}^{m} \Delta \tau_{ij k}$$

where $\Delta \tau_{ij k}$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge $(i,j)$ by the $k$th ant between time $t$ and $t + n$; it is given by

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if kth ant uses edge } (i, j) \text{ in its tour (between time } t \text{ and } t + n) \\ 0 & \text{otherwise} \end{cases}$$

where $Q$ is a constant and $L_k$ is the tour length of the $k$th ant.
Constraint Handling

In order to satisfy the constraint that an ant visits all the $n$ different towns, we associate with each ant a data structure called the tabu list, that saves the towns already visited up to time $t$ and forbids the ant to visit them again before $n$ iterations (a tour) have been completed. When a tour is completed, the tabu list is used to compute the ant's current solution (i.e., the distance of the path followed by the ant). The tabu list is then emptied and the ant is free again to choose.
The transition probability from town \( i \) to town \( j \) for the \( k \)th ant as

\[
p_{ij}^k(t) = \begin{cases} 
\frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ik}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta} & \text{if } j \in \text{allowed}_k \\
0 & \text{otherwise}
\end{cases}
\]

Where visibility \( \eta_{ij} = 1/d_{ij} \).

This quantity is not modified during the run of the AS.
\(\alpha\) and \(\beta\) are parameters that control the relative importance of trail versus visibility. Therefore the transition probability is a trade-off between visibility (which says that close towns should be chosen with high probability, thus implementing a greedy constructive heuristic) and trail intensity at time \(t\) (that says that if on edge \((i, j)\) there has been a lot of traffic then it is highly desirable, thus implementing the autocatalytic process)
Algorithm (Initialization & Randomly place ants)

1. Initialize:
   - Set $t := 0$ \{t is the time counter\}
   - Set $NC := 0$ \{NC is the cycles counter\}
   - For every edge $(i, j)$ set an initial value $\tau_{ij}(t) = c$ for trail intensity and $\Delta \tau_{ij} = 0$
   - Place the $m$ ants on the $n$ nodes

2. Set $s := 1$ \{ s is the tabu list index\}
   - For $k := 1$ to $m$ do
     - Place the starting town of the $k$th ant in $\text{tabu}_k(s)$
Algorithm (Build tours)

3. Repeat until tabu list is full
   \{this step will be repeated \((n - 1)\) times\}
   Set \(s := s + 1\)
   For \(k := 1\) to \(m\) do
     Choose the town \(j\) to move to, with probability \(p_{ij}^{\uparrow k}(t)\)
     \{at time \(t\) the \(k\)th ant is on town \(i = \text{tabu}_k(s - 1)\}\)
     Move the \(k\)th ant to the town \(j\)
     Insert town \(j\) in tabu\(_k(s)\)
4. For $k := 1$ to $m$ do

Move the $k$th ant from $\text{tabu}_k(n)$ to $\text{tabu}_k(1)$

Compute the length $L_k$, of the tour described by the $k$th ant

Update the shortest tour found

For every edge $(i, j)$

For $k := 1$ to $m$ do

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } (i, j) \in \text{tour described by } \text{tabu}_k \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta \tau_{ij} := \Delta \tau_{ij} + \Delta \tau_{ij}^k;$$
Algorithm (Update trail)

5. For every edge \((i, j)\) compute \(\tau_{ij}(t + n)\) according to equation:

\[
\tau_{ij}(t + n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}
\]

Set \(t = t + n\)

Set \(NC = NC + 1\)

For every edge \((i, j)\) set \(\Delta \tau_{ij} = 0\)
Algorithm (Loop or exit)

6. If \((NC < NC_{MAX})\) and (not stagnation behavior) then
   Empty all tabu lists
   Goto step 2
else
   Print shortest tour
   stop

AOT'15, MNIT, Jaipur, 18-22 May 2015
A simple TSP example

\[ d_{AB} = 100; d_{BC} = 60 \ldots; d_{DE} = 150 \]
How to choose next city?

\[ p^k_{ij}(t) = \begin{cases} 
\frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta} & \text{if } j \in \text{allowed}_k \\
0 & \text{otherwise}
\end{cases} \]
Iteration 2

A
B
C
D
E

[A,D]
[B,C]
[C,B]
[D,E]

[E,A]
Iteration 4

[C, B, E, D]

E

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]

D

B

[A, DCE]
Iteration 5

[A,D,C,E,B]
[C,B,E,D,A]
[D,E,A,B,C]
[E,A,B,C,D]
[B,C,D,A,E]
$[A,D,C,E,B]$

$[B,C,D,A,E]$

$[C,B,E,D,A]$

$[D,E,A,B,C]$

$[E,A,B,C,D]$

$L_1 = 300$

$L_2 = 450$

$L_3 = 260$

$L_4 = 280$

$L_5 = 420$

$$
\tau_{i,j}^k = \begin{cases} 
\frac{Q}{L_k} & \text{if } (i, j) \in \text{bestTour} \\
0 & \text{otherwise}
\end{cases}
$$
Performance of AS compared to TS and SA on the OLIVER30 problem. Results are averaged over ten runs using integer distances.

<table>
<thead>
<tr>
<th></th>
<th>Best</th>
<th>Average</th>
<th>Std.dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
<td>420</td>
<td>420.4</td>
<td>1.3</td>
</tr>
<tr>
<td>TS</td>
<td>420</td>
<td>420.6</td>
<td>1.5</td>
</tr>
<tr>
<td>SA</td>
<td>422</td>
<td>459.8</td>
<td>25.1</td>
</tr>
</tbody>
</table>
Some inherent advantages

- Positive Feedback accounts for rapid discovery of good solutions
- Distributed computation avoids premature convergence
- The greedy heuristic helps find acceptable solution in the early stages of the search process.
- The collective interaction of a population of agents.
Disadvantages in Ant Systems

- Slower convergence than other Heuristics
- Performed poorly for TSP problems larger than 75 cities.
- No centralized processor to guide the AS towards good solutions
- Can be used for discrete optimization problems
Conclusion

- ACO is a relatively new metaheuristic approach for solving hard combinatorial optimization problems.
- Artificial ants implement a randomized construction heuristic which makes probabilistic decisions.
- The cumulated search experience is taken into account by the adaptation of the pheromone trail.
- ACO shows great performance with the “ill-structured” problems like network routing.
- In ACO local search is important to obtain good results.
References


Reference

Thank You
Teaching–learning-based optimization

Dr. Anil Swarnkar
Assistant Professor
Department of Electrical Engineering
Malaviya National Institute of Technology Jaipur
Email: aswarnkar.ee@mnit.ac.in
Introduction

- This method works on the effect of influence of a teacher on the output of learners in a class.
- Like other nature-inspired algorithms, TLBO is also a population-based method and uses a population of solutions to proceed to the global solution.
- The population is considered as a group of learners or a class of learners.
- The process of TLBO is divided into two parts: the first part consists of the ‘Teacher Phase’ and the second part consists of the ‘Learner Phase’.
- ‘Teacher Phase’ means learning from the teacher and ‘Learner Phase’ means learning by the interaction between learners.
Introduction

• Here, output is considered in terms of results or grades.
• The teacher is generally considered as a highly learned person who shares his or her knowledge with the learners.
• The quality of a teacher affects the outcome of the learners.
• It is obvious that a good teacher trains learners such that they can have better results in terms of their marks or grades.
Distribution of marks obtained by learners taught by two different teachers

Curve-2 represents better results than curve-1 and so it can be said that teacher T2 is better than teacher T1 in terms of teaching. The main difference between both the results is their mean (M2 for Curve-2 and M1 for Curve-1), i.e. a good teacher produces a better mean for the results of the learners. Learners also learn from interaction between themselves, which also helps in their results.
Model for the distribution of marks obtained for a group of learners.

- The figure shows a model for the marks obtained for learners in a class with curve-A having mean $M_A$.
- The teacher is considered as the most knowledgeable person in the society, so the best learner is mimicked as a teacher, which is shown by $T_A$.
- The teacher tries to disseminate knowledge among learners, which will in turn increase the knowledge level of the whole class and help learners to get good marks or grades.
- So a teacher increases the mean of the class according to his or her capability. In the figure, teacher $T_A$ will try to move mean $M_A$ towards their own level according to his or her capability, thereby increasing the learners’ level to a new mean $M_B$.
- Teacher $T_A$ will put maximum effort into teaching his or her students, but students will gain knowledge according to the quality of teaching delivered by a teacher and the quality of students present in the class.
- The quality of the students is judged from the mean value of the population.
- Teacher $T_A$ puts effort in so as to increase the quality of the students from $M_A$ to $M_B$, at which stage the students require a new teacher, of superior quality than themselves, i.e. in this case the new teacher is $T_B$.
- Hence, there will be a new curve-B with new teacher $T_B$.
- In TLBO, different design variables will be analogous to different subjects offered to learners and the learners’ result is analogous to the ‘fitness’, as in other population-based optimization techniques.
- The teacher is considered as the best solution obtained so far.
Teacher phase

- A **good teacher** is one who brings his or her learners up to his or her level in terms of knowledge.
- But in practice this is not possible and a teacher can only move the mean of a class up to some extent depending on the capability of the class.
- This follows a random process depending on many factors.
Teacher phase

Let $M_i$ be the mean and $T_i$ be the teacher at any iteration $i$. $T_i$ will try to move mean $M_i$ towards its own level, so now the new mean will be $T_i$ designated as $M_{new}$.

The solution is updated according to the difference between the existing and the new mean given by:

$$\text{Difference}_\text{Mean}_i = r_i (M_{new} - T_F M_i)$$

where $T_F$ is a teaching factor that decides the value of mean to be changed, and $r_i$ is a random number in the range $[0, 1]$.

The value of $T_F$ can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability as:

$$T_F = \text{round}[1 + \text{rand}(0, 1) \{2 - 1\}]$$

This difference modifies the existing solution according to the following expression:

$$X_{new,i} = X_{old,i} + \text{Difference}_\text{Mean}_i$$
Learner phase

- A learner learns something new if the other learner has more knowledge than him or her.
- Learner modification is expressed as:

\[ \text{For } i = 1: P_n \]
\[ \text{Randomly select two learners } X_i \text{ and } X_j, \text{ where } i \neq j \]
\[ \text{If } f(X_i) < f(X_j) \]
\[ X_{\text{new},i} = X_{\text{old},i} + r_i (X_i - X_j) \]
\[ \text{Else} \]
\[ X_{\text{new},i} = X_{\text{old},i} + r_i (X_j - X_i) \]
\[ \text{End} \]
\[ \text{End} \]
- Accept \(X_{\text{new}}\) if it gives a better function value.
Flow Chart

Start

Initialize number of students (population), termination criterion

Calculate the mean of each design variable

Identify the best solution (teacher)

Modify solution based on best solution:

\[ X_{\text{new}} = X_{\text{old}} + r(X_{\text{teacher}} - T_{\text{mean}}) \]

Reject No Is new solution better than existing? Yes Accept

Select any two solutions randomly \( X_i \) and \( X_j \)

Yes \( X_i \) better than \( X_j \) No

\[ X_{\text{new}} = X_{\text{old}} + r(X_i - X_j) \]

\[ X_{\text{new}} = X_{\text{old}} + r(X_j - X_i) \]

Reject No Is new solution better than existing? Yes Accept

Is termination criteria satisfied?

No

Yes

Final value of solutions
Implementation of TLBO for optimization

Step 1: Define the optimization problem and initialize the optimization parameters.

- Define the population size \( (P_n) \), number of generations \( (G_n) \), number of design variables \( (D_n) \), and limits of design variables \( (UL, LL) \).
- Define the optimization problem as: Minimize \( f(X) \).
- Marks in subjects \( X_i \) where \( i = 1, 2, \ldots, D_n \)
- where \( f(X) \) is the objective function, \( X \) is a vector for design variables such that

\[
LL_i \leq X_i \leq UL_i
\]
Step 2: Initialize the population.

- Generate a random population according to the population size and number of design variables.
- For TLBO, the population size indicates the number of learners and the design variables indicate the subjects (i.e. courses) offered.
- This population is expressed as

$$\text{population} = \begin{bmatrix}
  x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\
  x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{P_n,1} & x_{P_n,2} & \cdots & x_{P_n,D}
\end{bmatrix}$$
Step 3: Teacher phase.

- Calculate the mean of the population column-wise, which will give the mean for the particular subject as:
  \[ M_D = [m_1, m_2, \ldots, m_D] \]

- The best solution will act as a teacher for that iteration
  \[ X_{teacher} = X_{f(X)=min} \]

- The teacher will try to shift the mean from \( M_D \) towards \( X_{teacher} \), which will act as a new mean for the iteration. So,
  \[ M_{new,D} = X_{teacher} \]

- The difference between two means is expressed as
  \[ \text{Difference}_D = r \left( M_{new,D} - T_F M_D \right) \]

- The value of \( T_F \) is selected as 1 or 2. The obtained difference is added to the current solution to update its values using
  \[ X_{new,D} = X_{old,D} + \text{Difference}_D \]

- Accept \( X_{new} \) if it gives better function value
Step 4: Learner phase.

- The mathematical expression is explained earlier.

Step 5: Termination criterion.

- Stop if the maximum generation number is achieved; otherwise repeat from Step 3.
Reference

Testing on Benchmark Functions

Benchmark Function 1 (13 design variables, 9 linear inequality constraint)

\[ \min f(x) = 5 \sum_{i=1}^{4} x_i - 5 \sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \]

S.T. \[ g_1(x) = 2x_1 + 2x_2 + x_3 + x_4 - 10 \leq 0, \] 
\[ g_2(x) = 2x_1 + 2x_3 + x_4 + x_5 - 10 \leq 0, \] 
\[ g_3(x) = 2x_2 + 2x_3 + x_6 + x_7 - 10 \leq 0, \] 
\[ g_4(x) = -8x_1 + x_2 \leq 0, \] 
\[ g_5(x) = -8x_2 + x_3 \leq 0, \] 
\[ g_6(x) = -8x_3 + x_4 \leq 0, \] 
\[ g_7(x) = -2x_4 - x_5 + x_6 \leq 0, \] 
\[ g_8(x) = -2x_6 - x_7 + x_8 \leq 0, \] 
\[ g_9(x) = -2x_8 - x_9 + x_{10} \leq 0, \] 
\[ 0 \leq x_i \leq 1, \quad i = 1, 2, 3, \ldots, 9, \] 
\[ 0 \leq x_i \leq 100, \quad i = 10, 11, 12, \quad 0 \leq x_i \leq 1, \quad i = 13. \]
The optimum solution for this problem is at \( x^* = (1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1) \) with objective function value \( f(x^*) = -15 \).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Best</th>
<th>Mean</th>
<th>Worst</th>
<th>Function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ES</td>
<td>-15</td>
<td>-15</td>
<td>-15</td>
<td>240 000</td>
</tr>
<tr>
<td>PESO</td>
<td>-15</td>
<td>-14.710</td>
<td>-13</td>
<td>350 000</td>
</tr>
<tr>
<td>CDE</td>
<td>-13</td>
<td>-14.999996</td>
<td>-14.999993</td>
<td>100 100</td>
</tr>
<tr>
<td>CoDE</td>
<td>-15</td>
<td>-15</td>
<td>-15</td>
<td>248 000</td>
</tr>
<tr>
<td>ABC</td>
<td>-15</td>
<td>-15</td>
<td>-15</td>
<td>240 000</td>
</tr>
<tr>
<td>TLBO</td>
<td>-15</td>
<td>-15</td>
<td>-15</td>
<td>25 000</td>
</tr>
</tbody>
</table>
Benchmark Function 2 (10 design variables and one nonlinear equality constraint)

\[
\text{Max } f(x) = (\sqrt{n})^n \prod_{i=1}^{n} x_i \\
\text{S.T. } h(x) = \sum_{i=1}^{4} x_i^2 - 1 = 0 \\
\text{where } \quad n = 10 \quad \text{and} \quad 0 \leq x_i \leq 10 \quad (i = 1, 2, \ldots, n).
\]
The global maximum is at $x^* = (1/\sqrt{n}, 0/\sqrt{n}, 1/\sqrt{n}, \ldots)$ with objective function value $f(x^*) = 1$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Best</th>
<th>Mean</th>
<th>Worst</th>
<th>Function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ES</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>240 000</td>
</tr>
<tr>
<td>PESO</td>
<td>0.99393</td>
<td>0.764813</td>
<td>0.464</td>
<td>350 000</td>
</tr>
<tr>
<td>CDE</td>
<td>0.995413</td>
<td>0.788635</td>
<td>0.639920</td>
<td>100 100</td>
</tr>
<tr>
<td>ABC</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>240 000</td>
</tr>
<tr>
<td>TLBO</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100 000</td>
</tr>
</tbody>
</table>
Benchmark Function 3 (7 design variables and 4 nonlinear inequality constraints.)

\[
\begin{align*}
\text{Min } f(x) &= \ (x_1 - 10)^2 + 5(x_2 - 12)^2 + 12 + 3(x_4 - 11)^2 \\
&\quad + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7
\end{align*}
\]

S.T. 
\[
\begin{align*}
g_1(x) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0, \\
g_2(x) &= -282 + 7x_1 + 3x_2 + 10x_3 + x_4 - x_5 \leq 0, \\
g_3(x) &= -196 + 23x_1^2 + x_2^2 + x_6 - 8x_7 \leq 0, \\
g_4(x) &= 4x_1^2 + x_2^2 - 3x_4 - 2x_3^2 + 5x_6 - 11x_7 \leq 0,
\end{align*}
\]

where

\[-10 \leq x_i \leq 10 \quad (i = 1, 2, \ldots, 7).\]
The optimum solution is at \( x^* = (2.330499, 1.951372, -0.477514, 4.365726, -0.6244870, 1.1038131, 1.594227) \) with objective function value \( f(x^*) = 680.6300573 \).
Benchmark Function 4 (8 design variables and 3 nonlinear inequality and 3 linear inequality constraints.)

Min \( f(x) = x_1 + x_2 + x_3 \)
S.T. \( g_1(x) = -1 + 0.0025(x_4 + x_6) \leq 0, \)
\( g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0, \)
\( g_3(x) = -1 + 0.01(x_8 - x_5) \leq 0, \)
\( g_4(x) = -x_1x_6 + 833.333x_4 + 100x_1 - 83333.333 \leq 0, \)
\( g_5(x) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0, \)
\( g_6(x) = -x_3x_8 + 250000 + x_3x_5 - 2500x_5 \leq 0, \)
where
\( -100 \leq x_1 \leq 10000, \quad -1000 \leq x_i \leq 10000 \quad (i = 2, 3), \)
\( -100 \leq x_i \leq 10000 \quad (i = 4, 5, \ldots, 8). \)
The optimum solution is at $x^* = (579.3066, 1359.9709, 5109.9707, 182.0177, 295.601, 217.982, 286.165, 395.6012)$ with objective function value $f(x^*) = 7049.248021$. 

<table>
<thead>
<tr>
<th>Methods</th>
<th>Best</th>
<th>Mean</th>
<th>Worst</th>
<th>Function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ES</td>
<td>7051.903</td>
<td>7253.047</td>
<td>7638.366</td>
<td>240,000</td>
</tr>
<tr>
<td>PESO</td>
<td>7049.38</td>
<td>7205.5</td>
<td>7894.812</td>
<td>350,000</td>
</tr>
<tr>
<td>CDE</td>
<td>7049.2481</td>
<td>7049.2483</td>
<td>7049.2485</td>
<td>100,100</td>
</tr>
<tr>
<td>ABC</td>
<td>7053.904</td>
<td>7224.407</td>
<td>7604.132</td>
<td>240,000</td>
</tr>
<tr>
<td>TLBO</td>
<td>7049.2481</td>
<td>7083.6732</td>
<td>7224.4968</td>
<td>100,000</td>
</tr>
</tbody>
</table>
Benchmark Function 5 (3 design variables)

\[
\text{Max } f(x) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100}
\]

S.T. \( g(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 \leq 0 \)

where

\[
0 \leq x_i \leq 10 \quad (i = 1, 2, 3)
\]

\[
p, q, r = 1, 2, 3, 9.
\]
The optimum solution is at \( x^* = (5, 5, 5) \) with objective function value \( f(x^*) = 1 \).

### Table 5

Comparison of results obtained by different optimization methods for benchmark function 5.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Best</th>
<th>Mean</th>
<th>Worst</th>
<th>Function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ES</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>240 000</td>
</tr>
<tr>
<td>PESO</td>
<td>1</td>
<td>0.998875</td>
<td>0.994</td>
<td>350 000</td>
</tr>
<tr>
<td>CDE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100 100</td>
</tr>
<tr>
<td>CoDE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>248 000</td>
</tr>
<tr>
<td>ABC</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>240 000</td>
</tr>
<tr>
<td>TLBO</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>50 000</td>
</tr>
</tbody>
</table>
Thank You
Short Term Course

Advanced Optimization Techniques

Day 4 : (21-May-2015)
Grey Wolf Optimizer (GWO)

Dr. Rajesh Kumar
Associate Professor
Electrical Engineering
MNIT Jaipur
April 17, 2015
OUTLINE

1. About Grey Wolf
2. Developers of Algorithm
3. Wolf behaviour in nature
4. Algorithm development
5. Example
6. Advantages over other techniques
7. Application on Unit commitment problem
About Grey Wolf

- Wolf is characterised by powerful teeth, bushy tail and lives and hunts in packs. The average group size is 5-12.
- Their natural habitats are found in the mountains, forests, plains of North America, Asia and Europe.
- Grey wolf (Canis lupus) belongs to Canidae family.
- Grey wolves are considered as apex predators, meaning that they are at the top of the food chain.
Developers of Algorithm

Seyedali Mirjalili
Seyed Mohammad Mirjalili
Andrew Lewis
Wolf behaviour in nature

Social behaviour

- Hierarchy exits in pack
- $\alpha$ is the leader and decision maker.
- $\beta$ and $\delta$ assist $\alpha$ in decision making.
- Rest of the wolves ($\omega$) are followers.
Wolf behaviour in nature

Hunting behaviour

Group hunting behaviour is of equal interest in studying optimization.

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attacking the prey.
OUTLINE
About Grey Wolf
Developers of Algorithm
Wolf behaviour in nature
Algorithm development
Example
Advantages over other techniques
Application on Unit commitment problem

Social behaviour
Hunting behaviour

Approach, track and pursuit
OUTLINE
- About Grey Wolf
- Developers of Algorithm
- Wolf behaviour in nature
- Algorithm development
- Example
- Advantages over other techniques
- Application on Unit commitment problem

Social behaviour
Hunting behaviour

Pursuit

Dr. Rajesh Kumar
GREY WOLF OPTIMIZER (GWO)
OUTLINE
About Grey Wolf
Developers of Algorithm
Wolf behaviour in nature
Algorithm development
Example
Advantages over other techniques
Application on Unit commitment problem

Social behaviour
Hunting behaviour

Harass
Encircling
At the end, when the prey stops, wolves make a approximate regular polygon around it and lay down
Social hierarchy

In order to mathematically model the social hierarchy of wolves when designing GWO, we consider the fittest solution as the alpha ($\alpha$). Consequently, the second and third best solutions are named beta ($\beta$) and delta ($\delta$) respectively. The rest of the candidate solutions are assumed to be omega ($\omega$). In the GWO algorithm the hunting (optimization) is guided by $\alpha$, $\beta$, and $\delta$. The $\omega$ wolves follow these three wolves.
Encircling prey
Encircling Prey: Mathematical Modeling

The mathematical model of the encircling behaviour is represented by the equations:

\[ D = |CX_p - AX(t)| \]  \hspace{1cm} (1)

\[ X(t + 1) = X_p(t) - AD \]  \hspace{1cm} (2)
Encircling Prey: Mathematical Modeling

- A and C are coefficient vectors given by:
  \[ A = 2a r_1 a \]  \( (3) \)
  \[ C = 2r_2 \]  \( (4) \)

- t is the current iteration
- X is the position vector of a wolf
- \( r_1 \) and \( r_2 \) are random vectors \( \in [0, 1] \) and a linearly varies from 2 to 0
- More description in later slides
Hunting

- Grey wolves have the ability to recognize the location of prey and encircle them.
- The hunt is usually guided by the alpha. The beta and delta might also participate in hunting occasionally.
- However, in an abstract search space we have no idea about the location of the optimum (prey).
- In order to mathematically simulate the hunting behaviour, we suppose that the alpha, beta and delta have better knowledge about the potential location of prey.
Hunting

\[ \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)| \]  
\[ \vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot (\vec{D}_\delta) \]  
\[ \vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \]

where \( t \) indicates the current iteration, \( \vec{X}_\alpha(t), \vec{X}_\beta(t) \) and \( \vec{X}_\delta(t) \) are the position of the gray wolves \( \alpha, \beta \) and \( \delta \) at \( t^{th} \) iteration, \( \vec{X}(t) \) presents the position of the gray wolf at \( t^{th} \) iteration.

\[ \vec{A}(.) = 2 \vec{a} \cdot \text{rand}(0, 1) - \vec{a} \]  
\[ \vec{C}(.) = 2 \cdot \text{rand}(0, 1) \]

Where \( \vec{a} \) is the linear value varies from 2 to 0 according to iteration. \( \vec{A}(.) \) and \( \vec{C}(.) \) are the coefficient vector of \( \alpha, \beta \) and \( \delta \) wolves.
Attacking prey & Search for prey

Dr. Rajesh Kumar

Grey Wolf Optimizer (GWO)
Example

minimization of Korn function

\[ f(x_1, x_2) = \min\{(x_1 - 5)^2 + (x_2 - 2)^2\} \]
### Iteration 1

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.1686</td>
<td>4.4100</td>
<td>7.1739</td>
</tr>
<tr>
<td>2</td>
<td>6.2104</td>
<td>4.0935</td>
<td>5.8479</td>
</tr>
<tr>
<td>3</td>
<td>7.4231</td>
<td>8.3880</td>
<td>46.6773</td>
</tr>
<tr>
<td>4</td>
<td>2.8950</td>
<td>0.8703</td>
<td>5.7074</td>
</tr>
<tr>
<td>5</td>
<td>6.1062</td>
<td>3.7275</td>
<td>4.2079</td>
</tr>
<tr>
<td>6</td>
<td>6.3458</td>
<td>3.2158</td>
<td>3.2896</td>
</tr>
<tr>
<td>7</td>
<td>7.5690</td>
<td>6.1457</td>
<td>23.7866</td>
</tr>
<tr>
<td>8</td>
<td>6.2471</td>
<td>4.0456</td>
<td>5.7397</td>
</tr>
<tr>
<td>9</td>
<td>6.9965</td>
<td>4.5846</td>
<td>10.6663</td>
</tr>
<tr>
<td>10</td>
<td>4.7372</td>
<td>3.3048</td>
<td>1.7717</td>
</tr>
<tr>
<td>11</td>
<td>4.8148</td>
<td>3.4931</td>
<td>2.2637</td>
</tr>
<tr>
<td>12</td>
<td>5.9444</td>
<td>3.4433</td>
<td>2.9751</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>4.7372</td>
<td>3.3048</td>
<td>1.7717</td>
</tr>
<tr>
<td>$\beta$</td>
<td>4.8148</td>
<td>3.4931</td>
<td>2.2637</td>
</tr>
<tr>
<td>$\delta$</td>
<td>5.9444</td>
<td>3.4433</td>
<td>2.9751</td>
</tr>
</tbody>
</table>
Update process

$$\vec{D}_\alpha = |2.\text{rand}()[4.7372, 3.3048] - [6.1686, 4.4100]|$$
$$X_1 = [4.7372, 3.3048] - (2\vec{a}.\text{rand}(0, 1) - \vec{a})\vec{D}_\alpha$$

$$\vec{D}_\beta = |2.\text{rand}()[4.8148, 3.4931] - [6.1686, 4.4100]|$$
$$X_2 = [4.8148, 3.4931] - (2\vec{a}.\text{rand}(0, 1) - \vec{a})\vec{D}_\beta$$

$$\vec{D}_\delta = |2.\text{rand}()[5.9444, 3.4433] - [6.1686, 4.4100]|$$
$$X_3 = [5.9444, 3.4433] - (2\vec{a}.\text{rand}(0, 1) - \vec{a})\vec{D}_\delta$$

$$\vec{X}(1,:) = \frac{X_1 + X_2 + X_3}{3} = [4.0487, 2.6051]$$
## Iteration 2

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.0487</td>
<td>2.6051</td>
<td>1.2710</td>
</tr>
<tr>
<td>2</td>
<td>4.6492</td>
<td>3.0427</td>
<td>1.2103</td>
</tr>
<tr>
<td>3</td>
<td>5.4633</td>
<td>3.6633</td>
<td>2.9813</td>
</tr>
<tr>
<td>4</td>
<td>5.6096</td>
<td>3.5901</td>
<td>2.9001</td>
</tr>
<tr>
<td>5</td>
<td>4.6582</td>
<td>3.0302</td>
<td>1.1781</td>
</tr>
<tr>
<td>6</td>
<td>4.7476</td>
<td>3.3369</td>
<td>1.8509</td>
</tr>
<tr>
<td>7</td>
<td>4.2452</td>
<td>2.6600</td>
<td>1.0054</td>
</tr>
<tr>
<td>8</td>
<td>4.9026</td>
<td>3.2497</td>
<td>1.5712</td>
</tr>
<tr>
<td>9</td>
<td>4.5202</td>
<td>2.9588</td>
<td>1.1495</td>
</tr>
<tr>
<td>10</td>
<td>5.3971</td>
<td>3.5432</td>
<td>2.5392</td>
</tr>
<tr>
<td>11</td>
<td>4.1136</td>
<td>2.5382</td>
<td>1.0754</td>
</tr>
<tr>
<td>12</td>
<td>5.0927</td>
<td>3.1546</td>
<td>1.3418</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>4.2452</td>
<td>2.6600</td>
<td>1.0054</td>
</tr>
<tr>
<td>$\beta$</td>
<td>4.1136</td>
<td>2.5382</td>
<td>1.0754</td>
</tr>
<tr>
<td>$\delta$</td>
<td>5.0927</td>
<td>3.1546</td>
<td>1.3418</td>
</tr>
</tbody>
</table>
Iteration 3

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.4838</td>
<td>2.7843</td>
<td>0.8816</td>
</tr>
<tr>
<td>2</td>
<td>4.5634</td>
<td>2.8257</td>
<td>0.8725</td>
</tr>
<tr>
<td>3</td>
<td>4.5899</td>
<td>2.8395</td>
<td>0.8730</td>
</tr>
<tr>
<td>4</td>
<td>4.7486</td>
<td>2.9400</td>
<td>0.9467</td>
</tr>
<tr>
<td>5</td>
<td>4.6340</td>
<td>2.8684</td>
<td>0.8881</td>
</tr>
<tr>
<td>6</td>
<td>4.5957</td>
<td>2.8445</td>
<td>0.8767</td>
</tr>
<tr>
<td>7</td>
<td>4.5830</td>
<td>2.8366</td>
<td>0.8738</td>
</tr>
<tr>
<td>8</td>
<td>4.5787</td>
<td>2.8339</td>
<td>0.8729</td>
</tr>
<tr>
<td>9</td>
<td>4.5750</td>
<td>2.8321</td>
<td>0.8730</td>
</tr>
<tr>
<td>10</td>
<td>4.5724</td>
<td>2.8396</td>
<td>0.8727</td>
</tr>
<tr>
<td>11</td>
<td>4.5703</td>
<td>2.8295</td>
<td>0.8727</td>
</tr>
<tr>
<td>12</td>
<td>4.5696</td>
<td>2.8291</td>
<td>0.8727</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>4.5634</td>
<td>2.8257</td>
<td>0.8725</td>
</tr>
<tr>
<td>$\beta$</td>
<td>4.5696</td>
<td>2.8291</td>
<td>0.8727</td>
</tr>
<tr>
<td>$\delta$</td>
<td>4.5750</td>
<td>2.8321</td>
<td>0.8730</td>
</tr>
</tbody>
</table>
Flow chart
Advantages over other techniques

- Easy to implement due to simple structure.
- Less storage requirement than the other techniques.
- Convergence is faster due to continuous reduction of search space and Decision variables are very less ($\alpha$, $\beta$ and $\delta$).
- It avoids local optima when applied to composite functions also.
- only two main parameters to be adjusted ($a$ and $C$).
Unit Commitment Problem

Unit Commitment (UC) is a very significant optimization task, which plays an important role in the operation planning of power systems.

UCP is considered as two linked optimization decision processes, namely the unit-scheduled problem that determines on/off status of generating units in each time period of planning horizon, and the economic load dispatch problem.

UCP is a complex nonlinear, mixed-integer combinational optimization problem with 01 variables that represents on/off status.
Unit commitment problem

Table: Total costs of the BGWO method for test systems

<table>
<thead>
<tr>
<th>No. of Unit</th>
<th>Best Cost ($)</th>
<th>Average Cost ($)</th>
<th>Worst Cost ($)</th>
<th>Std. Deviation</th>
<th>CPU Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>563937.3</td>
<td>563976.6</td>
<td>564017.7</td>
<td>40.2</td>
<td>31.3</td>
</tr>
<tr>
<td>20</td>
<td>1124687.9</td>
<td>1124837.7</td>
<td>1124941.1</td>
<td>128.7</td>
<td>58.7</td>
</tr>
<tr>
<td>40</td>
<td>2248220.0</td>
<td>2248397.6</td>
<td>2248614.0</td>
<td>174.2</td>
<td>124.6</td>
</tr>
<tr>
<td>60</td>
<td>3367893.4</td>
<td>3367881.1</td>
<td>3367933.4</td>
<td>37.9</td>
<td>216.9</td>
</tr>
<tr>
<td>80</td>
<td>4492399.4</td>
<td>4492608.1</td>
<td>4492672.5</td>
<td>154.4</td>
<td>347.5</td>
</tr>
<tr>
<td>100</td>
<td>5612309.4</td>
<td>5612377.2</td>
<td>5612496.3</td>
<td>96.9</td>
<td>505.6</td>
</tr>
</tbody>
</table>
Performance Comparison

Table: Comparison With Other Algorithms

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>565825</td>
<td>1130660</td>
<td>2258503</td>
<td>3394066</td>
<td>4526022</td>
<td>5657277</td>
</tr>
<tr>
<td>GA</td>
<td>565825</td>
<td>1126243</td>
<td>2251911</td>
<td>3376625</td>
<td>4504933</td>
<td>5627437</td>
</tr>
<tr>
<td>EP</td>
<td>564551</td>
<td>1125494</td>
<td>2249093</td>
<td>3371611</td>
<td>4498479</td>
<td>5623885</td>
</tr>
<tr>
<td>MA</td>
<td>565827</td>
<td>1128192</td>
<td>2249589</td>
<td>3370820</td>
<td>4494214</td>
<td>5616314</td>
</tr>
<tr>
<td>GRASP</td>
<td>565825</td>
<td>1128160</td>
<td>2259340</td>
<td>3383184</td>
<td>4525934</td>
<td>5668870</td>
</tr>
<tr>
<td>LRPSO</td>
<td>565869</td>
<td>1128072</td>
<td>2251116</td>
<td>3376407</td>
<td>4496717</td>
<td>5623607</td>
</tr>
<tr>
<td>PSO</td>
<td>564212</td>
<td>1125983</td>
<td>2250012</td>
<td>3374174</td>
<td>4501538</td>
<td>5625376</td>
</tr>
<tr>
<td>IBPSO</td>
<td>563977</td>
<td>1125216</td>
<td>2248581</td>
<td>3367865</td>
<td>4491083</td>
<td>5610293</td>
</tr>
<tr>
<td>BFWA</td>
<td>563977</td>
<td>1124858</td>
<td>2248372</td>
<td>3367912</td>
<td>4492680</td>
<td>5612619</td>
</tr>
<tr>
<td>BGWO</td>
<td>563937.3</td>
<td>1124684.8</td>
<td>2248280.0</td>
<td>3367893.4</td>
<td>4492399.4</td>
<td>5612309.4</td>
</tr>
</tbody>
</table>
THANK YOU . . .

Mail: rkumar.ee@gmail.com
Fireworks Algorithm

Dr. Rajesh Kumar
Associate Professor
MNIT Jaipur
Outlines

1. Basic Fireworks Algorithm (FA)
2. FA Variant
3. Example
Developers of the Algorithm

Ying Tan and Yuanchu Zhu, Peking University Developed in 2010.
1.3 Introduction to Fireworks Algorithm

When a firework is set off, a shower of sparks will fill the local space around the firework.

- The explosion can be viewed as a search in the local Space around a firework.

![Fireworks爆炸示意图](image.png)

Fig. 1. Search Process of Fireworks Algorithm
from Nature to Mathematical Modeling

- Definition of firework

**Good firework:** firework can generate a big population of sparks within a small range.

**Bad firework:** firework that generate a small Population of sparks within a big range.

The basic FA algorithm is on the basis of simulating the process of the firework explosion illuminating the night sky.
2. Basic Firework Algorithm (FA)

2.1 Problem description
2.2 Flowchart of FA
2.3 Design of basic FA
   - Selection operators
   - Number of son sparks
2.4 Experimental results
2.1 Problem description

- Suppose the FA is designed for the general optimization problems:

\[
\text{Minimize } f(x) \in \mathbb{R}, \quad x_{\text{min}} \leq x \leq x_{\text{max}},
\]

Where \( x = x_1, x_2, \ldots, x_d \) denotes a location in the potential space, \( f(x) \) is an objective function, and \( x_{\text{min}} \) and \( x_{\text{max}} \) denote the bounds of the potential space.
2.2 FA’s Flowchart

1. Set N fireworks
2. Obtain the sparks, select N fireworks for next generation
3. Evaluate the sparks for next generation
4. Repeat

Options:
- N: Next iteration
- Y: Yes, continue
- N: No, stop
2.3 Design of basic Firework Algorithm

Number of sparks generated by each firework $X_i$ is defined as follows:

$$s_i = m * \frac{y_{max} - f(x_i) + \zeta}{\sum_{i=1}^{n} (y_{max} - f(x_i)) + \zeta}$$
2.3 Design of basic Firework Algorithm

To avoid overwhelming effects of splendid fireworks, bounds are defined

\[
\hat{s}_i = \begin{cases} 
\text{round}(a \times m) & \text{if } s_i \leq am \\
\text{round}(b \times m) & \text{if } s_i < am, a < b < 1 \\
\text{round}(s_i) & \text{otherwise}
\end{cases}
\]
2.3 Design of basic Firework Algorithm

In contrast to the design of sparks number, the amplitude of a good firework explosion is smaller than that of a bad one. Amplitude of defined by:

$$f(x_i) - y_{min} + \xi = \frac{\prod_{i=1}^{n} (f(x_i) - y_{min})}{\text{BIG RANGE}}$$

$$= \frac{\prod_{i=1}^{n} \text{LITTLE SPARKS}}{\text{SMALL RANGE}}$$

$$= \frac{\prod_{i=1}^{n} \text{GOOD}}{\text{BAD}}$$
2.3 Design of basic Firework Algorithm

**Generating Sparks.** In explosion, sparks may undergo the effects of explosion from random z directions (dimensions).

---

**Algorithm 1. Obtain the location of a spark**

1. Initialize the location of the spark: \( x_i \);
2. \( z = \text{round}(d \cdot \text{rand}(0, 1)) \);
3. Randomly select \( z \) dimensions of \( x_j \);
4. Calculate the displacement: \( \hat{x}_j = A_i \cdot \text{rand}(-1, 1) \);
5. For each dimension \( \hat{x}_k \in \{ \text{pre-selected} \ z \ \text{dimensions of} \ \hat{x}_j \} \)
   - \( \hat{x}_k = \hat{x}_k + h \);
   - If \( \hat{x}_k < x_k^{\text{min}} \) or \( \hat{x}_k > x_k^{\text{max}} \) then
     - Map \( \hat{x}_k \) to the potential space: \( \hat{x}_k = x_k^{\text{min}} + |\hat{x}_k| \% (x_k^{\text{max}} - x_k^{\text{min}}) \);
   - End if
6. End for
2.3 Design of basic Firework Algorithm

To keep the diversity of sparks, we design another way of generating sparks----Gaussian explosion.

**Algorithm 2.** Obtain the location of a spark:

```plaintext
Initialize the location of the spark: \( \hat{x}_j = \hat{x} \),
\( z = \text{round}(d \cdot \text{rand}(0, 1)) \);
Randomly select \( z \) dimensions of \( \hat{x}_j \);
Calculate the coefficient of Gaussian explosion for each dimension \( \hat{x}_k^j \in \) pre-selected \( z \)
\( \hat{x}_k^j = \hat{x}_k^j \cdot g; \)
if \( \hat{x}_k^j < x_{k_{\text{min}}}^j \) or \( \hat{x}_k^j > x_{k_{\text{max}}}^j \) then
map \( \hat{x}_k^j \) to the potential space: \( \hat{x}_k^j = \cdot \)
end if
end for
```
2.3 Design of basic Firework Algorithm

Classroom

Sparse

KEEP DIVERSITY!

Crowd
2.3 Design of basic Firework Algorithm

To avoid the crowd of the sparks, we borrow the idea of density to design the selection operators.

\[ R(x_i) = \sum_{j \in K} d(x_i, x_j) = \sum_{j \in K} ||x_i - x_j|| \]

Where

\[ p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_i)} \]
Example of FWA:

\[ f(x) = \min\{x^2(1) + x^2(2)\} \]

Boundary limits:

\[ 0 \leq x \leq 5 \]
Random selection of location

<table>
<thead>
<tr>
<th>Location number</th>
<th>X(1)</th>
<th>X(2)</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.7</td>
<td>-1.2</td>
<td>15.4324</td>
</tr>
<tr>
<td>2</td>
<td>4.13</td>
<td>-4.02</td>
<td>33.2173</td>
</tr>
<tr>
<td>3</td>
<td>-2.21</td>
<td>0.46</td>
<td>5.0957</td>
</tr>
</tbody>
</table>
Number of spark generation in each explosion calculation:

\[ S_1 = 5 \times \frac{33.2173 - 15.4324 + \varepsilon}{\sum_{i=1}^{3} (33.2173 - f(x_i)) + \varepsilon} \approx 2 \]

Accordingly:

\[ S_2 \approx 1 \]
\[ S_3 \approx 3 \]

**Note:** Here \( \varepsilon \) is the minimum value assignee very less to avoid infeasible solution and Variable \( S_i \) is \( 1 \leq S_i \leq 3 \).
Amplitude of spark calculation

\[ A_1 = 2 \times \frac{15.4324 - 5.0957 + \epsilon}{\sum_{i=1}^{3} f(x_i) - 5.0957 + \epsilon} \approx 0.54 \]

Accordingly:

\[ A_2 \approx 1.46 \]

\[ A_2 \approx 0.25 \]
<table>
<thead>
<tr>
<th>Location number</th>
<th>X(1)</th>
<th>X(2)</th>
<th>f(x)</th>
<th>No of Spark</th>
<th>Amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.7</td>
<td>1.32</td>
<td>15.4324</td>
<td>2</td>
<td>0.54</td>
</tr>
<tr>
<td>2</td>
<td>4.13</td>
<td>-4.02</td>
<td>33.2173</td>
<td>1</td>
<td>1.46</td>
</tr>
<tr>
<td>3</td>
<td>-2.21</td>
<td>0.46</td>
<td>5.0957</td>
<td>3</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Obtaining location of Spark

\[ h = 0.57 \times \text{rand}(-1,1) \approx 0.49 \]

\[ x_1' = [-3.7, 1.32] + h \approx [-3.21, 1.81] \]

Note: Dimension is selected randomly.
<table>
<thead>
<tr>
<th>Bit(x(1))</th>
<th>Bit(x(1))</th>
<th>X(1)</th>
<th>X(2)</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-3.21</td>
<td>1.81</td>
<td>13.5802</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-3.72</td>
<td>0.92</td>
<td>14.7409</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3.084</td>
<td>-4.03</td>
<td>25.7538</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-2.21</td>
<td>0.429</td>
<td>5.0613</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-2.002</td>
<td>0.667</td>
<td>4.4545</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-2.064</td>
<td>0.46</td>
<td>4.4713</td>
</tr>
</tbody>
</table>

**Note**: Bit(x) shows the dimension selected in random process i.e. 1 selected and 0 is not selected.
Obtaining the location of Gaussian spark

\[ g = Gaussian(1,1) \]

\[ \tilde{x}_1' = [0, 1.32] * g \approx [0, 2.1486] \]

Gaussian spark generation:

\[ \tilde{x}_1' = [-3.70, 2.1486] \]
<table>
<thead>
<tr>
<th></th>
<th>Bit(x(1))</th>
<th>Bit(x(2))</th>
<th>X(1)</th>
<th>X(2)</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-3.70</td>
<td>2.1486</td>
<td>18.3065</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.5631</td>
<td>-0.5495</td>
<td>0.619</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2.21</td>
<td>0.4956</td>
<td>5.1297</td>
</tr>
</tbody>
</table>
Selection of next position of spark

- Select the fittest solution for the first location of next iteration.
- Rest of the N-1 location are chosen accordingly:
  - Find the relative distance between all locations of spark (all spark) using x value (Note: not using f(x) in selection, only \( x_i \) is used).
  - Randomly select next value according to there relative distance.

<table>
<thead>
<tr>
<th>X(1)</th>
<th>X(2)</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5631</td>
<td>-0.5495</td>
<td>0.6190</td>
</tr>
<tr>
<td>3.0843</td>
<td>-4.03</td>
<td>25.7538</td>
</tr>
<tr>
<td>4.13</td>
<td>-4.03</td>
<td>33.21</td>
</tr>
</tbody>
</table>
Thank you

Any Question?
Development of Directed Bee Colony Optimization Algorithm

Dr. Rajesh Kumar

PhD, PDB (NUS, Singapore)
SMIEEE, FIETE, LMCSI, MIE (I), SMIACSIT, LMISET, MIAENG

Associate Professor, Department of Electrical Engineering
Adjunct Associate Professor, Centre of Energy and Environment
Malaviya National Institute of Technology, Jaipur, India, 302017
Tel: (91) 9549654481
rkumar@ieee.org, rkumar.ee@gmail.com
http://drrajeshkumar.wordpress.com
Goal of Optimization

Find values of the variables that minimize or maximize the objective function while satisfying the constraints.
Mathematical Formulation of Optimization Problems

minimize the objective function

\[ \min f(x), \quad x = (x_1, x_2, \ldots, x_n) \]

subject to constraints

\[ c_i(x) \geq 0 \]

\[ c_i(x) = 0 \]

Example

\[ \min \left[ (x_1 - 2)^2 + (x_2 - 1)^2 \right] \]

subject: \( x_1^2 - x_2^2 \leq 0 \)

\( x_1 + x_2 \leq 2 \)
Classification of Optimization Problems

- Single variable
- Multi-variable
- Constrained
- Non-constrained
- Single objective
- Multi-objective
- Linear
- Non-linear
Local and Global Optimizers

- which of the minima is found depends on the starting point
- such minima often occur in real applications
Major Techniques used in Optimization
Motivation

- The new optimization algorithm inspired by group decision-making process of honey bees for best nest site which is analogous to the optimization process.
Working principle of bee in nature

• Initially some bees go for searching the nectar sources.

• Then the bee loads the nectar and returns to the hive to relinquish and performs the waggle dance.

• The waggle dance gives information about the quality, distance and direction of flower patch.

• The recruited bees will evaluate the quality of source upon exploring it.

• If the patch is still good enough for nectar, then they perform waggle dance to convey the same which enables quiescent bees in recruitment of more bees to explore the patch.
Decision Process

- Two methods used by bee swarms to reach to a decision for finding out the best nest site.
- Consensus: Widespread agreement among the group is taken into account.
- Quorum: The decision for best site happens when a site crosses the quorum (threshold) value.
- In DBC, both consensus and quorum have been mimicked, compared and presented for finding out the optimum solution.
Directed Bee Colony Algorithm

- 1/10\textsuperscript{th} of total bees (known as scouts) go for searching next site.
- Information of best site through waggle dance
- Decision making Process:
  - Consensus
  - Quorum
- Inform other bees through piping
Honey Bee Algorithm

- Waggle dance
- Calculation of angle b/w sun and flowers
- waggle dance occurs on the surface of a swarm rather than on the combs inside a hive.
Honey Bee Algorithm

- Bee runs through a figure-eight pattern on a vertical comb
- Angle of the run indicates the direction to the food source
- Duration of the waggle run relates to the distance to the food source
The proposed Directed Bee Colony Optimization algorithm

• **DBC** is a computational system in which several bees work, interact with each other and in unison take decision to achieve goals.

• **Assumptions made in the proposed algorithm:**
  - Bees live and act in a given environment.
  - Bees attempt to achieve particular goals or perform particular tasks.
  - Information exchange process accompanies no losses.
  - Bees will never die and hence number of bees remains constant during the process.
DBC Algorithm

- In DBC, all bees live in an environment i.e. FSR. An environment is organized in a structure as shown below for d(1,2,3) parameters:

Fig. 1. Domain of the objective function with one independent parameter.

Fig. 2. Domain of the objective function with two independent parameters.

Fig. 3. Domain of the objective function with three independent parameters.
• A point inside each volume is chosen as the starting point for the search process, which in proposed approach is the midpoint of that volume through various investigations.

• The midpoint of total cluster can be calculated using (1)

\[
\left[ \frac{W_{1i} + W_{1f}}{2}, \frac{W_{2i} + W_{2f}}{2}, ..., \frac{W_{di} + W_{df}}{2} \right]
\]  

(1)

• If there is one parameter only one bee explores the search.

• If the parameters are \(W_{1i} = 1, W_{1f} = 6\) then 5 bees are sent for exploration for the axis given by two parameters \(d_1\) and \(d_2\).

• If the parameters are \(W_{1i} = 1, W_{1f} = 5\) and \(W_{2i} = 1, W_{2f} = 5\) then bees are sent for exploration.
Bee search methodology

• In the proposed optimization, the analogy has been derived with the search approach adopted by bees and the tendency of remembering only four locations at a time.

• The author has used a very popular optimization technique (Fig. 4) usually known as Nelder–Mead method (NM) based on geometric operations (reflection, expansion, contraction and shrinking).
Main steps of the simplex algorithm

1: Initial Simplex

2: Center of gravity
(without the worst point)

3: Reflection
(\text{Default: } \rho = 1.00)

4a: Contraction
(\text{Default: } \gamma = 0.50)

4b: Expansion
(\text{Default: } \chi = 2.00)

5: Shrinkage
(\text{Default: } \sigma = 0.50)
Communication of the information through waggle dance

- The individual optimal solutions of bees are communicated to the centralized system that chooses best preferable solution.

- **Definition 5.** A food quality index $FQI : FSR \to \mathcal{R}$ is a measure of the quality of the solution that is represented by the bee food search procedure.

- For optimal minimum cases it selects the best optimal solution which can mathematically expressed as

  $FQI(i) = \min(FV(i))$  \hspace{1cm} (2)

where, $FV(i)$ represent the different search value equivalent to food value obtained by a bee.
Global Optimal Solution Selection

• When bee encounters multiple global optimal solutions with same fitness value it selects the optimal solution nearer to the starting point (Fig. 5).

• This is due to the natural phenomenom used by bees when it find nectar site with same fitness value.

Fig. 5. Solution report by bee to hive: (a) with x constraints and (b) without constraints.
Bee based decision processes-consensus and quorum

- **Consensus:** Once exploration and waggle dance (transmission of data) is finished the global optimized point is chosen by comparing the Fitness Values of all the optimized points in the optimum vector table i.e. global best, gbest as in case of PSO [10,11].

- **Point with the lowest Fitness Value is selected as the global optimized point** \((X_G)\).

- **Quorum:** In quorum method, if the number of bees providing an optimized solution reaches quorum threshold \((\varepsilon_q)\) then that optimum solution is considered as the final solution and further search of optimum result is stopped and hence saves time.
Flow Chart for DBC

Fig. 6. Flowchart for the proposed DBC algorithm.
### Table 1

<table>
<thead>
<tr>
<th>Function</th>
<th>Function range</th>
<th>$\overrightarrow{x}$</th>
<th>$f_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere $f_1(\overrightarrow{x}) = \sum_{i=1}^{n} x_i^2$</td>
<td>$-100 \leq x_i \leq 100$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel's Problem 2.22 $f_2(\overrightarrow{x}) = \sum_{i=1}^{n} x_i \sin(\sqrt{</td>
<td>x_i</td>
<td>})$</td>
<td>$-10 \leq x_i \leq 10$</td>
</tr>
<tr>
<td>Rosenbrock $f_3(\overrightarrow{x}) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$</td>
<td>$-100 \leq x_i \leq 100$</td>
<td>$1, 1, ..., 1$</td>
<td>0</td>
</tr>
<tr>
<td>Rotated hyperrhombus $f_4(\overrightarrow{x}) = \sum_{i=1}^{n} x_i^2$</td>
<td>$-10 \leq x_i \leq 10$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Generalized Schwefel's Problem 2.26 $f_5(\overrightarrow{x}) = -\sum_{i=1}^{n} x_i \sin(\sqrt{</td>
<td>x_i</td>
<td>})$</td>
<td>$-512 \leq x_i \leq 512$</td>
</tr>
<tr>
<td>Rastrigin $f_6(\overrightarrow{x}) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10)$</td>
<td>$-5.12 \leq x_i \leq 5.12$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ackley $f_7(\overrightarrow{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$</td>
<td>$-32 \leq x_i \leq 32$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Griewank $f_8(\overrightarrow{x}) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}$</td>
<td>$-600 \leq x_i \leq 600$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>Function</th>
<th>Function range</th>
<th>$\overrightarrow{x}$</th>
<th>$f_{\text{bias}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shifted sphere (CE2005 F1) $f_{\text{sh}}(\overrightarrow{x}) = \sum_{i=1}^{n} x_i^2 + f_{\text{bias}}$ where $z = x - a$</td>
<td>$-100 \leq x_i \leq 100$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Shifted Schwefel's Problem 1.22 (CE2005 F2) $f_{\text{sh}}(\overrightarrow{x}) = \sum_{i=1}^{n} x_i^2$</td>
<td>$-10 \leq x_i \leq 10$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Shifted Rosenbrock (CE2005 F6) $f_{\text{sh}}(\overrightarrow{x}) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) + f_{\text{bias}}$ where $z = x - a + 1$</td>
<td>$-30 \leq x_i \leq 30$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Shifted Rastrigin (CE2005 F9) $f_{\text{sh}}(\overrightarrow{x}) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10) + f_{\text{bias}}$ where $z = x - a$</td>
<td>$-5 \leq x_i \leq 5$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Shifted rotated high conditional elliptic (CE2005 F3) $f_{\text{sh}}(\overrightarrow{x}) = \sum_{i=1}^{n} (10^a x_i^{i-1} - 10z_i^2) + f_{\text{bias}}$ where $z = (x - a)M$; $M$ is the linear transformation matrix</td>
<td>$-100 \leq x_i \leq 100$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Shifted rotated Griewank's function without bounds (CE2005 F7) $f_{\text{sh}}(\overrightarrow{x}) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}$ where $z = (x - a)M$; $M$ is the linear transformation matrix</td>
<td>$-100 \leq x_i \leq 100$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rotated hybrid composition function CF1 $f_{\text{rot}}(\overrightarrow{x})$ marked as F1 in CE2005 benchmark problem set</td>
<td>$-5 \leq x_i \leq 5$</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Rotated hybrid composition function CF2 $f_{\text{rot}}(\overrightarrow{x})$ marked as F18 in CE2005 benchmark problem set</td>
<td>$-5 \leq x_i \leq 5$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>
Impact of bee decision making on the performance of DBC

- The decision making processes consensus and quorum are compared (Table 3).
- Performance can be increased by switching between the two decision making processes.
- Two choices of parameters can be used: number of bees and quorum (threshold) value.
- As the quorum value is increased the optimal solution approaches towards the global value at an expense of time (Fig. 7).
Choice of parameters:

- Effect of number of bees: More the number of bees, more is the computational time taken for processing but with increased accuracy towards the optimal result (Fig. 8).
Comparison with other similar algorithms: Case study-I: normal benchmark problems

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Functions</th>
<th>Sphere</th>
<th>Schwefel's 2.22</th>
<th>Rosenbrock</th>
<th>Rotated-hyper</th>
<th>Schwefel's 2.26</th>
<th>Rastrigin</th>
<th>Ackley</th>
<th>Griewank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA</td>
<td>1.87E-04</td>
<td>1.7152E-01</td>
<td>9.46128E-03</td>
<td>2.82441E-11</td>
<td>-8.3796E+02</td>
<td>1.3906E+00</td>
<td>1.13E+00</td>
<td>7.678E-03</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>3.20E-05</td>
<td>7.2851E-02</td>
<td>2.1562E+00</td>
<td>5.4379E-11</td>
<td>8.1603E-08</td>
<td>8.2424E-01</td>
<td>4.070E-01</td>
<td>4.1207E-03</td>
</tr>
<tr>
<td></td>
<td>EA</td>
<td>5.26E-11</td>
<td>3.127E-06</td>
<td>4.325E-05</td>
<td>0</td>
<td>-8.3796E+02</td>
<td>2.04E-08</td>
<td>7.342E-06</td>
<td>1.23E-08</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>8.25E-11</td>
<td>2.456E-06</td>
<td>3.654E-05</td>
<td>4.12E-12</td>
<td>4.65E-06</td>
<td>2.79E-08</td>
<td>5.63E-06</td>
<td>2.08E-08</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>4.390E-06</td>
<td>1.7082E-06</td>
<td>2.7301E-06</td>
<td>5.1592E-11</td>
<td>4.3287E-06</td>
<td>2.2156E-06</td>
<td>4.543E-06</td>
<td>3.5012E-06</td>
</tr>
<tr>
<td></td>
<td>BFO</td>
<td>1.429E-05</td>
<td>1.983E-05</td>
<td>1.735E-04</td>
<td>2.7634E-11</td>
<td>-8.3796E+02</td>
<td>2.59E-02</td>
<td>6.324E-06</td>
<td>9.975E-01</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>9.603E-06</td>
<td>1.986E-05</td>
<td>2.134E-04</td>
<td>7.864E-11</td>
<td>1.304E-08</td>
<td>3.06E-02</td>
<td>5.34E-06</td>
<td>8.142E-01</td>
</tr>
<tr>
<td></td>
<td>HS</td>
<td>6.017E-12</td>
<td>2.4000E-06</td>
<td>2.5648E-06</td>
<td>6.5685E-11</td>
<td>-8.3796E+02</td>
<td>1.5391E-09</td>
<td>6.306E-06</td>
<td>6.9944E-14</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>9.574E-12</td>
<td>1.708E-06</td>
<td>4.700E-06</td>
<td>7.864E-11</td>
<td>2.304E-12</td>
<td>2.080E-09</td>
<td>4.542E-06</td>
<td>3.501E-14</td>
</tr>
<tr>
<td></td>
<td>DE</td>
<td>1.754E-13</td>
<td>4.136E-07</td>
<td>2.869E-06</td>
<td>0</td>
<td>-8.3796E+02</td>
<td>3.10E-12</td>
<td>5.637E-12</td>
<td>9.42E-03</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>3.318E-13</td>
<td>3.132E-07</td>
<td>3.86E-06</td>
<td>2.31E-12</td>
<td>1.82E-14</td>
<td>6.56E-12</td>
<td>3.72E-12</td>
<td>1.33E-02</td>
</tr>
<tr>
<td></td>
<td>ABC</td>
<td>4.30E-17</td>
<td>0</td>
<td>2.325E-06</td>
<td>0</td>
<td>-8.3796E+02</td>
<td>4.37E-17</td>
<td>9.64E-17</td>
<td>4.04E-17</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>1.07E-17</td>
<td>1.34E-17</td>
<td>2.56E-05</td>
<td>1.23E-13</td>
<td>0</td>
<td>1.10E-17</td>
<td>5.24E-17</td>
<td>1.12E-17</td>
</tr>
<tr>
<td></td>
<td>DBC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8.81E-18</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Case study-I: normal benchmark problems

- DBC performed either better or at same level with the other algorithms at lower dimensions.
- The t-tests show that DBC perform better than most of algorithms and in a statistically significant manner.
- The uniqueness of the solution by the proposed algorithm is evident from the zero standard deviation (SD) which is novel character of proposed algorithm and can be recommended for real time applications.
Case study-II: scalability study

- The dimension of the functions is increased from 2 to 30 and the performance with different algorithms is noted.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Functions</th>
<th>Sphere</th>
<th>Schwefel's 2.22</th>
<th>Rosenbrock</th>
<th>Rotated-hyper</th>
<th>Schwefel's 2.26</th>
<th>Rastrigin</th>
<th>Ackley</th>
<th>Griewank</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>AE</td>
<td>1.255E-02</td>
<td>8.058E-01</td>
<td>1.6628E+02</td>
<td>4.8340E+00</td>
<td>-8.914E+03</td>
<td>1.0438E+01</td>
<td>1.099E+00</td>
<td>1.2342E+00</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>4.3E-03</td>
<td>6.52E-01</td>
<td>5.951E+01</td>
<td>2.5505E+00</td>
<td>7.25E+02</td>
<td>2.6386E+00</td>
<td>2.496E-01</td>
<td>1.1045E-01</td>
</tr>
<tr>
<td>EA</td>
<td>AE</td>
<td>1.789E-03</td>
<td>1.7207E-02</td>
<td>3.1318E+01</td>
<td>1.5891E+02</td>
<td>-1.166E+04</td>
<td>7.1789E-01</td>
<td>1.047E-02</td>
<td>4.6366E-03</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>2.77E-04</td>
<td>1.70E-03</td>
<td>1.74E+01</td>
<td>4.2515E+00</td>
<td>2.34E+00</td>
<td>9.22E-01</td>
<td>9.08E-04</td>
<td>3.96E-03</td>
</tr>
<tr>
<td>PSO</td>
<td>AE</td>
<td>1.115E-02</td>
<td>1.0612E+00</td>
<td>4.0254E+02</td>
<td>1.412E+03</td>
<td>-1.0652E+04</td>
<td>3.2476E+00</td>
<td>1.492E-06</td>
<td>1.1151E-02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>1.421E+02</td>
<td>5.853E-01</td>
<td>6.336E+02</td>
<td>1.511E+03</td>
<td>6.63E+02</td>
<td>6.9521E+00</td>
<td>1.861E-06</td>
<td>1.4209E-02</td>
</tr>
<tr>
<td>BFO</td>
<td>AE</td>
<td>4.143E-01</td>
<td>3.1097E+00</td>
<td>1.265E-02</td>
<td>3.562E-02</td>
<td>-1.1631E+04</td>
<td>1.1592E+01</td>
<td>1.991E+01</td>
<td>9.147E+03</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>1.288E-01</td>
<td>1.908E-01</td>
<td>1.46E+01</td>
<td>2.54E-02</td>
<td>1.416E+01</td>
<td>2.250E+01</td>
<td>3.796E-01</td>
<td>7.62E-03</td>
</tr>
<tr>
<td>HS</td>
<td>AE</td>
<td>1.87E-04</td>
<td>1.7152E+01</td>
<td>3.4029E+02</td>
<td>4.297E+03</td>
<td>1.342E+02</td>
<td>7.157E-02</td>
<td>7.971E+00</td>
<td>1.2079E+02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>3.2E-05</td>
<td>7.2851E+02</td>
<td>2.6669E+02</td>
<td>1.362E+03</td>
<td>1.2539E+04</td>
<td>1.3906E+00</td>
<td>7.450E+00</td>
<td>1.9128E+02</td>
</tr>
<tr>
<td>DE</td>
<td>AE</td>
<td>8.2E-14</td>
<td>1.5E-09</td>
<td>3.246E+01</td>
<td>6.8E-11</td>
<td>-1.108E+04</td>
<td>6.92E+00</td>
<td>9.7E-08</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>5.9E-14</td>
<td>9.9E-10</td>
<td>2.98E+01</td>
<td>7.4E-11</td>
<td>5.74E+02</td>
<td>3.88E+01</td>
<td>4.2E-08</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>2.194E-16</td>
<td>7.8E-11</td>
<td>2.914E+01</td>
<td>4.31E-12</td>
<td>1.042E+01</td>
<td>7.876E-13</td>
<td>6.52E-10</td>
<td>3.54E-02</td>
</tr>
<tr>
<td>DBC</td>
<td>AE</td>
<td>2.423E-26</td>
<td>5.3291E-14</td>
<td>2.6347E+03</td>
<td>7.6377E-24</td>
<td>-1.2567E+04</td>
<td>0</td>
<td>2.753E-14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Case study-III: shifted and shifted rotated benchmark problems

- A set of CEC 2005 benchmark problems shifted, shifted rotated and hybrid composite is evaluated for DBC algorithm and compared with the counterparts.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Functions</th>
<th>Shifted sphere</th>
<th>Shifted Schwefel's 1.2</th>
<th>Shifted Rosenbrock</th>
<th>Shifted Rastrigin</th>
<th>Shifted rotated high conditional elliptic function</th>
<th>Rotated hybrid composition function CFI</th>
<th>Rotated hybrid composition function CF2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>AE</td>
<td>9.75E-04</td>
<td>1.68</td>
<td>31.56</td>
<td>5.61E-04</td>
<td>1.06E+06</td>
<td>6.93E-01</td>
<td>144.63</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>6.17E-04</td>
<td>1.25</td>
<td>33.95</td>
<td>3.86E-04</td>
<td>5.93E-05</td>
<td>4.31E-01</td>
<td>20.38</td>
</tr>
<tr>
<td>EA</td>
<td>AE</td>
<td>1.74E+03</td>
<td>2.61E+03</td>
<td>9.13E+03</td>
<td>3.67E+01</td>
<td>6.84E+03</td>
<td>5.50E+03</td>
<td>213.00</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>1.02E+02</td>
<td>1.36E+02</td>
<td>3.63E+03</td>
<td>4.17E+02</td>
<td>2.64E+02</td>
<td>5.35E+03</td>
<td>18.76</td>
</tr>
<tr>
<td>PSO</td>
<td>AE</td>
<td>0.00</td>
<td>0.00</td>
<td>39.89</td>
<td>5.26</td>
<td>5.64E+04</td>
<td>4.36E-02</td>
<td>103.51</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.00</td>
<td>0.00</td>
<td>113.23</td>
<td>2.74</td>
<td>3.27E-04</td>
<td>1.96E-02</td>
<td>8.77</td>
</tr>
<tr>
<td>EPSO</td>
<td>AE</td>
<td>5.5775E-04</td>
<td>1.0545E-02</td>
<td>2.0754E+02</td>
<td>3.1543E+01</td>
<td>7.2254E-05</td>
<td>1.6035E+00</td>
<td>1.3963E+02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>8.9835E-04</td>
<td>1.1203E-02</td>
<td>7.2116E+02</td>
<td>8.0409E+00</td>
<td>6.1746E-06</td>
<td>9.8414E-01</td>
<td>4.4296E+01</td>
</tr>
<tr>
<td>BFO</td>
<td>AE</td>
<td>9.82E-01</td>
<td>3.62</td>
<td>70.97</td>
<td>15.62</td>
<td>1.66E+05</td>
<td>8.4537</td>
<td>165.04</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>2.44E-01</td>
<td>8.60E-01</td>
<td>1.75E-03</td>
<td>2.04</td>
<td>3.21E-04</td>
<td>2.01E-01</td>
<td>109.19</td>
</tr>
<tr>
<td>DE</td>
<td>AE</td>
<td>0.00</td>
<td>0.00</td>
<td>7.53E-09</td>
<td>3.76E-06</td>
<td>1.37E+05</td>
<td>0.00</td>
<td>1.30E-14</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.00</td>
<td>0.00</td>
<td>8.73E-09</td>
<td>6.16E-06</td>
<td>0.00</td>
<td>0.00</td>
<td>7.11</td>
</tr>
<tr>
<td>DDE</td>
<td>AE</td>
<td>6.84E-16</td>
<td>1.89E-08</td>
<td>0.00</td>
<td>1.25</td>
<td>1.46E+15</td>
<td>2.89E-03</td>
<td>1.03E-15</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>5.72E-16</td>
<td>1.00E-08</td>
<td>0.00</td>
<td>6.99E+00</td>
<td>2.46E+15</td>
<td>8.48E-04</td>
<td>1.30E-14</td>
</tr>
<tr>
<td>LA-DE</td>
<td>AE</td>
<td>5.684E-14</td>
<td>3.431E-13</td>
<td>3.136E-13</td>
<td>3.997E+00</td>
<td>5.339E+02</td>
<td>3.299E-01</td>
<td>1.83E+02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>4.988E-14</td>
<td>8.311E-12</td>
<td>1.211E-12</td>
<td>6.298E+00</td>
<td>1.489E+00</td>
<td>2.488E-01</td>
<td>1.489E+00</td>
</tr>
<tr>
<td>HS</td>
<td>AE</td>
<td>6.440E+00</td>
<td>3.883E+03</td>
<td>3.401E+03</td>
<td>8.710E-01</td>
<td>1.500E+07</td>
<td>4.080E+03</td>
<td>3.448E+02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>2.777E+00</td>
<td>8.086E+03</td>
<td>5.272E+03</td>
<td>8.086E+00</td>
<td>4.456E+06</td>
<td>5.733E+00</td>
<td>2.581E+02</td>
</tr>
<tr>
<td>EHS</td>
<td>AE</td>
<td>4.343E-11</td>
<td>1.667E+09</td>
<td>4.183E-01</td>
<td>1.81E-06</td>
<td>5.024E+01</td>
<td>1.007E-05</td>
<td>1.103E+02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>3.866E-12</td>
<td>2.943E-10</td>
<td>1.7360E-02</td>
<td>5.927E-07</td>
<td>8.008E+01</td>
<td>2.064E-06</td>
<td>2.029E+00</td>
</tr>
<tr>
<td>ABC</td>
<td>AE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.16E-17</td>
<td>3.96E-11</td>
<td>1.024E+02</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Conclusion

- The nature inspired algorithms are analyzed and problems are identified there by making a way for the proposal of new DBC optimization.

- **Upshot of the proposed algorithm:** Generates better optimal solutions as compared to its counterparts.

- One has to choose between consensus and quorum, can go for consensus for better accuracy.

- In case of quorum better solution can be obtained by increasing the threshold.

- The results show that DBC performs better and also removes the randomness in the algorithm when compared to other algorithms at lower and higher dimensions by generating unique solution.

- Other classes of improved, mutated, directed and hybrid algorithms like CLPSO, DM-GA, HSDM, EPSDE are still to be compared and will be agenda for future research.
Advanced Optimization Techniques (AOT-15)

MARKOV CHAINS

Dr. Gunjan Soni
Assistant Professor, Dept. of Mech. Engg.
DEFINITION OF A MARKOV CHAIN

• Let $X_1$ be a random variable that characterizes the state of a system at discrete points in time $t=1,2,3,\ldots$. The family of random variables $\{X_1\}$ form a stochastic process.

• The number of states in a stochastic process may be finite or infinite.

Example 1.1 (Machine Maintenance)

The condition of a machine at the time of monthly preventive maintenance is characterized as fair, good, or excellent. For a month $t$, the stochastic process for this situation can be represented as:

$$X_t = \begin{cases} 
0, & \text{ poor} \\
1, & \text{ fair} \\
2, & \text{ good} 
\end{cases} \quad \text{for } T=1,2,3,\ldots$$
The random variable $X_t$ is finite because it represent three state: poor (0), fair(1), good (2).

Example 1.2 (Job Shop)
Jobs arrive randomly at job shop at the average rate of 5 jobs per hour. The arrival process follows a Poisson distribution which, theoretically, allow any number of jobs between zero and infinity to arrive at the shop during the time interval $(0, t)$. The infinite-state process describing the number of arriving jobs is

$$X_t = 0, 1, 2, \ldots, \text{to } 0$$
- A stochastic process is a Markov Process if the occurrence of a future state depends only on the immediately preceding states.
- For a given chronological time $t_0, t_1, \ldots, t_n$, the family of random variables $\{X_{tn}\} = \{x_1, x_2, \ldots, x_n\}$ is said to be a Markov process if it possesses the following property:

$$P\{X_{t_n} = x_n \mid X_{t_{n-1}} = x_{n-1}, \ldots, X_{t_0} = x_0\} = P\{X_{t_n} = x_n \mid X_{t_{n-1}} = x_{n-1}\}$$

- In a Markovian process with $n$ exhaustive and mutually exclusive states (outcomes), the probabilities at a specific point in time $t=0,1,2,...$ are usually written as

$$p_{ij} = P\{X_t = j \mid X_{t-1} = i\}, (i, j) = 1,2\ldots,n, t = 0,1,2\ldots T$$

This is known as **one-step transition probability** of moving from state $i$ at $t-1$ to state $j$ at $t$.
By definition:

\[ \sum_{j} P_{ij} = 1, i = 1, 2, 3 \ldots n \]

\[ P_{ij} \geq 0, (i, j) = 1, 2, 3 \ldots n \]

A convenient way for summarizing the one step transition probabilities is to use the following matrix notation:

\[
P = \begin{bmatrix}
p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\
p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\
& & & \ddots & \\
p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mn}
\end{bmatrix}
\]

The matrix P is Markov chain. All the transition probabilities \( P_{ij} \) are fixed and independent over time.
Example 1.3 (The Gardener Problem)
Every year, at the beginning of the gardening season (March-September), a gardener uses a chemical test to check soil condition. Depending on the outcomes of the test, productivity for the new season falls in one of the three states: (1) Good (2) Fair (3) Poor. Over the year, Gardener has observed that the last year's soil condition impacts current year's productivity and that the situation can be described by the Markov chain:

\[
P = \begin{bmatrix}
0.2 & 0.5 & 0.3 \\
0 & 0.5 & 0.5 \\
0 & 0 & 1
\end{bmatrix}
\]
The transition probabilities shows that the soil condition can either deteriorate or stay same but never improve.

- **state 1 Good**: there is 20% chance soil will not change next year, 50% chance it will become fair and 30% chance it will deteriorate.
- **state 2 Fair**: next year’s productivity may remain fair with 50% probability or become poor with probability 50%.
- **state 3 poor**: equal condition next year with probability 1.

\[
P_1 = \begin{pmatrix} .30 & .60 & .10 \\ .10 & .60 & .30 \\ 0 & .40 & .55 \end{pmatrix}
\]

The use of fertilizer now allows improvement in the deteriorating condition. 10% chance soil change from fair to good, 5% chance from poor to good and 40% chance from poor to fair.
ABSOLUTE AND n-STEP TRANSITION PROBABILITIES

Given the initial probabilities \( a^{(0)} = \{a_j^{(0)}\} \) of starting in state \( j \) and the transition matrix \( P \) of a Markov chain, the absolute probabilities \( a^{(n)} = \{a_j^{(n)}\} \) of being in state \( j \) after \( n \) transition (\( n > 0 \)) are computed as follows

\[
\begin{align*}
    a^{(1)} &= a^{(0)} P \\
    a^{(2)} &= a^{(1)} P = a^{(0)} PP = a^{(0)} P^2 \\
    a^{(3)} &= a^{(2)} P = a^{(0)} P^2 P = a^{(0)} P^3 \\
    & \vdots \\
    & \vdots \\
    a^{(n)} &= a^{(0)} P^n, \quad n = 1, 2, \ldots
\end{align*}
\]
The matrix $P^n$ is known as the n-step transition matrix. From these calculations:

$$P^n = P^{n-1}P$$

or

$$P^n = P^{n-m}P^m, 0 < m < n$$

These are known as **Chapman-Kolomogorov** equation.
Example 2.1 The following transition matrix applies to the gardener problem with fertilizer

\[ P_1 = \begin{bmatrix} 0.30 & 0.60 & 0.10 \\ 0.10 & 0.60 & 0.30 \\ 0.05 & 0.40 & 0.55 \end{bmatrix} \]

The initial condition of the soil is good – that is \( a^{(0)} = (1,0,0) \). Determine the absolute probabilities of the three state of the system after 1, 8 and 16 gardening seasons.

\[
P^2 = \begin{bmatrix} 0.30 & 0.60 & 0.10 \\ 0.10 & 0.60 & 0.30 \\ 0.05 & 0.40 & 0.55 \end{bmatrix} \times \begin{bmatrix} 0.30 & 0.60 & 0.10 \\ 0.10 & 0.60 & 0.30 \\ 0.05 & 0.40 & 0.55 \end{bmatrix} = \begin{bmatrix} 0.1550 & 0.5800 & 0.2650 \\ 0.1050 & 0.5400 & 0.3550 \\ 0.0825 & 0.4900 & 0.4275 \end{bmatrix}
\]
\[
P^4 = \begin{bmatrix}
.1550 & .5800 & .2650 \\
.1050 & .5400 & .3550 \\
.0825 & .4900 & .4275
\end{bmatrix} \times \begin{bmatrix}
.1550 & .5800 & .2650 \\
.1050 & .5400 & .3550 \\
.0825 & .4900 & .4275
\end{bmatrix} = \begin{bmatrix}
.10679 & .53295 & .36026 \\
.10226 & .52645 & .37129 \\
.09950 & .52193 & .37857
\end{bmatrix}
\]

\[
P^8 = \begin{bmatrix}
.10679 & .53295 & .36026 \\
.10226 & .52645 & .37129 \\
.09950 & .52193 & .37857
\end{bmatrix} \times \begin{bmatrix}
.10679 & .53295 & .36026 \\
.10226 & .52645 & .37129 \\
.09950 & .52193 & .37857
\end{bmatrix} = \begin{bmatrix}
.101753 & .525514 & .372733 \\
.101702 & .525435 & .372863 \\
.101669 & .525384 & .372863
\end{bmatrix}
\]

\[
P^{16} = \begin{bmatrix}
.101753 & .525514 & .372733 \\
.101702 & .525435 & .372863 \\
.101669 & .525384 & .372863
\end{bmatrix} \times \begin{bmatrix}
.101753 & .525514 & .372733 \\
.101702 & .525435 & .372863 \\
.101669 & .525384 & .372863
\end{bmatrix} = \begin{bmatrix}
.101659 & .52454 & .372881 \\
.101659 & .52454 & .372881 \\
.101659 & .52454 & .372881
\end{bmatrix}
\]
\[
\begin{bmatrix}
.30 & .60 & .10 \\
.10 & .60 & .30 \\
.05 & .40 & .55
\end{bmatrix}
\]

\[
\alpha^{(1)} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} .30 & .60 & .10 \\
.10 & .60 & .30 \\
.05 & .40 & .55 \end{bmatrix} = \begin{bmatrix} .30 & .60 & .10 \end{bmatrix}
\]

\[
P^8 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix}
.101753 & .525514 & .372733 \\
.101702 & .525435 & .372863 \\
.101669 & .525384 & .372863
\end{bmatrix} \cdot \begin{bmatrix} .101753 & .525514 & .372733 \\
.101702 & .525435 & .372863 \\
.101669 & .525384 & .372863 \end{bmatrix}
\]

\[
P^{16} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix}
.101659 & .52454 & .372881 \\
.101659 & .52454 & .372881 \\
.101659 & .52454 & .372881
\end{bmatrix} = \begin{bmatrix} .101659 & .52454 & .372881 \end{bmatrix}
\]

The row of \( P^8 \) and the vector of absolute probabilities \( P^8 \) are almost identical. The result more pronounced for \( P^{16} \). It shows as the number of transition increase, the absolute probabilities are independent of initial \( \alpha^{(0)} \). In this case the resulting probabilities are known as \textit{steady-state probabilities}. 
CLASSIFICATION OF THE STATES IN A MARKOV CHAIN

The states of a Markov chain can be classified based on the transition probabilities $p_{ij}$.

1. A state $j$ is **absorbing** if it returns to itself with certainty in one transition— that is $p_j$.

2. A state $j$ is **transient** if it can reach another state but cannot itself be reached back another state. Mathematically, this will happen if

   $\lim_{n \to \infty} p_{ij}^n = 0$ for all $i$.

3. A state $j$ is **recurrent** if the probability of being revisited from other states is 1. This happen if, and only if, the state is not transient.
4. A state $j$ is **periodic** with period $t > 1$ if a return is possible only in $t, 2t, 3t, \ldots$. This means that $p_{jj}^n = 0$ whenever $n$ is not divisible by $t$.

State 1 and 2 are transient because they cannot be reentered once the system is “trapped” in states 3 and 4.

\[
P = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & .3 & .7 \\
0 & 0 & .4 & .6 \\
\end{pmatrix}
\]
State 3 and 4 play role of an absorbing state, constitute of closed set. All states of a closed set must communicate, which means that it is possible to go from any state to any other state in the set in one or more transition

\[ p_{ij}^n > 0 \quad \text{for all} \quad i \neq j \quad \text{and} \quad n \geq 1 \]

State 3 and 4 can both be absorbing states if \( p_{33} = p_{44} = 1 \)

A closed Markov chain is said to be ergodic if all its state are recurrent and aperiodic. In this case, the absolute probabilities after \( n \) transitions, \( a^n = a^0 P^n \) always converge uniquely to a limited (steady state) distribution as \( n \to \infty \) that is independent of initial probabilities \( a^0 \).
Example 3.1 (Absorbing and Transient states)
Consider the gardener Markov chain with no fertilizer

\[
P = \begin{bmatrix}
0.2 & 0.5 & 0.3 \\
0 & 0.5 & 0.5 \\
0 & 0 & 1
\end{bmatrix}
\]

States 1 and 2 are transient because they reach state 3 but can never be reached back. State 3 is absorbing because \( p_{33} = 1 \). These classification can also be seen when \( \lim_{n \to \infty} p_{ij}^n = 0 \) is computed.

\[
P_{100} = \begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 1 \\
0 & 0 & 1
\end{bmatrix}
\]

Which shows in long run, the probability of ever reentering transient state 1 and 2 is zero whereas the probability of being “trapped” in absorbing state 3 is certain.
Example 3.2 (Periodic states)
Periodicity of a state can be test by computing $p^n$ and observing the values of $p^n_{jj}$ for $n=2,3,4,...$. These values will be positive only at the corresponding period of state.

$$
P = \begin{pmatrix}
0 & .6 & .4 \\
0 & 1 & 0 \\
.6 & .4 & 0
\end{pmatrix}
$$

$$
P^2 = \begin{pmatrix}
.24 & .76 & 0 \\
0 & 1 & 0 \\
0 & .76 & .24
\end{pmatrix}
$$

$$
P^3 = \begin{pmatrix}
0 & .904 & .0960 \\
0 & 1 & 0 \\
.144 & .856 & 0
\end{pmatrix}
$$

$$
P^4 = \begin{pmatrix}
.0576 & .9424 & 0 \\
0 & 1 & 0 \\
0 & .9424 & .0576
\end{pmatrix}
$$
Continuing with $n = 6, 7, \ldots$, $p^n$ shows that $p_{11}$ and $p_{33}$ are positive for even values of $n$ and zero otherwise. This means that the period for states 1 and 3 is 2.
Thank You
Multi Agent System for Micro Grid

Dr. Rajesh Kumar

PhD, PDF (NUS, Singapore)
SMIEEE, FIETE, MIE (I),LMCSI, SMIACSIT, LMISTE, MIAENG

Associate Professor, Department of Electrical Engineering
Adjunct Associate Professor, Centre of Energy
Malaviya National Institute of Technology, Jaipur, India, 302017
Tel: (91) 9549654481
rkumar@ieee.org, rkumar.ee@gmail.com
http://drrajeshkumar.wordpress.com/
Multi Agent System for Self Healing System
Multi Agent System for Self Healing System
Multi Agent System for Self Healing System

Smart Grid
Multi Agent System for Self Healing System

Electrical Infrastructure

"Intelligence" Infrastructure
Multi Agent System for Self Healing System

Smart Grid Applications

- Distributed Generation and Alternate Energy Sources
- Self-Healing Wide-Area Protection and Islanding
- Asset Management and On-Line Equipment Monitoring
- Real-time Simulation and Contingency Analysis
- Participation in Energy Markets
- Demand Response and Dynamic Pricing

Shared Information – Continuously Optimizing – Intelligent Responses!
Multi Agent System for Self Healing System
Smart Grid

The smart grid is the collection of all technologies, concepts, topologies, and approaches that allow the silo hierarchies of generation, transmission, and distribution to be replaced with an end-to-end, organically intelligent. Fully integrated environment where the business processes, objectives, and needs of all stakeholders are supported by the efficient exchange of data, services, and transactions.
Key characteristics of a smart grid

- Distributed generation – not only traditional large power stations, but also individual PV panels, micro-wind, etc.
- Grid optimization – system reliability, operational efficiency and asset utilization and protection
- Demand response – utilities offer incentives to customers to reduce consumption at peak times
- Advanced metering infrastructure (AMI) – smart meters
- Grid-scale storage
- Plug-in hybrid electric vehicles (PHEVs) and vehicle to grid (V2G)
Multi Agent System for Self Healing System

Micro Grid
Multi Agent System for Self Healing System

AOT'15, MNIT, Jaipur, 18-22 May 2015
Definition of Agent

1. An agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments. [Yoav Shoham, 1993]

2. An entity is a software agent if and only if it communicates correctly in an agent communication language. [Genesereth and Ketchpel, 1994]

3. Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions. [Hayes-Roth, 1995]
5. An agent is anything that can be viewed as (a) Perceiving its environment, and (b) Acting upon that environment [Russell and Norvig, 1995]

6. A computer system that is situated in some environment and is capable of autonomous action in its environment to meet its design objectives. [Wooldridge, 1999]
Agents: A working definition

An agent is a computational system that interacts with one or more counterparts or real-world systems with the following key features to varying degrees:

• Autonomy
• Reactiveness
• Pro-activeness
• Social abilities

e.g., autonomous robots, human assistants, service agents

The need is for automation and distributed use of online resources
Single agent-based systems differ from multi-agent systems:

• Environment: agents need to take into account others who may interfere with their plans and goals. They need to coordinate with others in order to avoid conflicts.

• Knowledge/expertise/skills: these are distributed

• Design: agents need not be homogeneous and may be designed and implemented using different methodologies and languages

• Interaction: agents interact following rules of interaction (interaction protocols)
Multi-Agent Learning Problem:

Agent tries to solve its learning problem, while other agents in the environment also are trying to solve their own learning problems.
Planning of Coordination Agent

- Centralized Multi-agent Planning: A coordinating agent gets receipt of all local plans.

- Distributed Multi-agent planning: each agent is provided with model of other agent.
Multi-agent systems

Influence area

Environment

Interactions
Coherence

- Coherence, i.e. how well the system behaves as a unit, is important in MASs

- Coherence can be measured in different ways depending on the problem domain and system.

- Usually measured from an external observer’s perspective who ascertains whether or not a system appears to be behaving in a coherent way — not necessarily something that the agents themselves are aware of.
MAS applications

They are particularly suitable for:

- Building systems to solve complex problems that cannot be solved by any one agent on its own.

- Dealing with problems that involve many problem-solving methods, require different types of expertise and knowledge or where there are multiple viewpoints.

- Creating systems where dynamic reorganization is required.

- Tasks in which the information resources are distributed.
MAS advantages

The use of MAS technology offers a number of advantages:

- Extensibility and flexibility
- Robustness and reliability
- Computational efficiency and speed
- Development and maintainability
- Reusability
- Reduced costs
MAS challenges

A number of issues with regards to their design and implementation:

• Efficient and effective interaction protocols
• Task and problem formulation, decomposition, task allocation to individual agents and subtask synthesis
• Agent identification, search and location in open systems
• Coherent and stable system behaviour while avoiding harmful interactions
• Representation of information about the state of the environment, as well as other agents, their actions and knowledge
• Team and organization formation
• Efficient planning and learning algorithms
Closed MAS

• Static design with pre-defined components and functionalities
• The properties of the system are known in advance:
  – common language
  – each agent can be developed as an expert
  – agents are cooperative
  – multiple developers can work towards the development of the system at the same time
• Example: a MAS in an organization
Advantages

• Distributed load and expertise
• Simplicity and predictability, since
  – components are known
  – interaction language and protocols are known
  – agents usually are cooperative
  – agents share architecture and software

Disadvantages

• Maintenance costs can be high
• May be less fault tolerant
• Difficult to inter-operate with other systems
Open MAS

• The system has no prior static design, only single agents within

• Agents are not necessarily aware of others – a mechanism for identifying, searching and locating others is required

• Agents may be non-cooperative, malicious or not trustworthy

• Example: open electronic marketplaces
Advantages

• Single agent or groups are designed separately (modular)
• Flexible and fault tolerant
• Evolutionary design
• Easier to maintain
• Dynamic, open society

Disadvantages

• Overall behaviour of the system not predictable
• Protocols, languages, ontologies may vary across agents
• Malicious behaviour difficult to avoid
Interaction

**Definition:** An interaction occurs when two or more agents are brought into a dynamic relationship through a set of reciprocal actions (Ferber 1999)

- Interactions develop as a result of a series of actions whose consequences influence the future behavior of agents
- May be direct or indirect, intended, or unintended
- Interaction assumes:
  - Agents that are capable of acting and/or communicating
  - Situations that promote interaction
  - Dynamic elements allowing for local and temporary relationships among agents
Elements of interactions

• Goals: The agents’ objectives and goals are important. Goals of different agents can be conflicting.

• Resources: agents require resources to achieve their objectives. Resources are not infinite, other agents may want to use them as well. Conflicts may arise.

• Expertise/skills/capabilities. Agents may lack the necessary skills, expertise of capabilities for accomplishing one or more of their tasks. They may require the ‘help’ of others.
Agent characterization

- **Self-interested/antagonistic agents**: have incompatible goals with others and they are interested in maximizing their own utility, not necessarily that of the society as a whole.

- **Cooperative/nonantagonistic agents**: have usually compatible goals; they act to maximize their own utility in conjunction with that of the entire system.
Agent communication

Why is communication important?

• Communication is required in MASs where agents have to cooperate, negotiate etc., with other agents
• Agents need to communicate in order to understand and be understood by others
• Diversity introduces heterogeneity

• Can we use natural language as an Agent Communication Language (ACL)? No, natural language is ambiguous
How do agents acquire intelligence?

Cognitive agent

The model of human intelligence and human perspective of the world ➔ characterise an intelligent agent using symbolic representations and mentalistic notions:

- **knowledge** - John knows humans are mortal
- **beliefs** - John took his umbrella because he believed it was going to rain
- **desires, goals** - John wants to possess a PhD
- **intentions** - John intends to work hard in order to have a PhD
- **choices** - John decided to apply for a PhD
- **commitments** - John will not stop working until getting his PhD
- **obligations** - John has to work to make a living
FIPA Agent Model
Multi Agent System for Self Healing System

MAS - Micro Grid
Intelligent Agent

Multi Agent System for Self Healing System
Security Architecture

Sending Agent

Provides encryption service

Msg Encoding Agent

Authorizes agent to provide encryption service

Security Manager

Issues channel encryption order

Channel Encoding Agent

Encrypts Channel

Encrypted Secure Channel

Receiving Agent

Sends Encrypted Msg

Receives Encrypted Msg

AOT'15, MNIT, Jaipur, 18-22 May 2015
System Structure
Centralized Multi-agent Self-healing Power System
Multi Agent System for Self Healing System

Power system model designed in Sim Power System and interfacing with Agent system
Multi Agent System for Self Healing System
Multi Agent System for Self Healing System

Schematic diagram MAS

Normal State

start

Simulink

Circuit parameter

Fault State

Control signal to CB

Fault sensing

Agent System

Normal State

end
Multi Agent System for Self Healing System

Agent Interaction Diagram
Fault current from wind farm without and with SFCL in case of fault in Distribution Grid.
Three phase Fault current and voltage
Multi Agent System for Self Healing System

Three phase Fault Current and voltage from wind farm with SFCL
Multi Agent System for Self Healing System

Distribution Grid Fault

- Reduction in Fault Current
- Improved voltage profile
Multi Agent System for Self Healing System

Transmission Line Fault

Reduction in fault current

Improved voltage profile
Customer Grid Fault

[Graphs illustrating the reduction in fault current and improved voltage profile]
Effect of MAS based SFCL for different types of fault

<table>
<thead>
<tr>
<th>Different Types of Fault</th>
<th>Current without SFCL (A)</th>
<th>Current with SFCL (A)</th>
<th>Voltage without SFCL (kV)</th>
<th>Voltage with SFCL (kV)</th>
<th>No: Cycles to compensate without SFCL</th>
<th>No: Cycles to compensate with SFCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Grid Fault</td>
<td>1985</td>
<td>1634</td>
<td>5.5</td>
<td>7.7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Transmission Line Fault</td>
<td>2240</td>
<td>1670</td>
<td>0</td>
<td>7.7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Customer Grid Fault</td>
<td>1687</td>
<td>1589</td>
<td>6.4</td>
<td>7.7</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table: Current reduction and Voltage profile Improvement
MAS Implementation for Critical Load Securing in PV Based Microgrid

- Critical loads are the loads in which continuity of supply should be maintained. (e.g. ICU of an HOSPITAL)
- In this work, Intelligent Distributed Autonomous Power Systems (IDAPS) comprises of solar photovoltaic as distributed energy resources (DER) and loads, as well as their control algorithms, has been developed.
  1. Once an upstream fault is detected, Island the microgrid from the main grid.
  2. During emergency condition, secure critical loads and shed non-critical loads according to the given priority.
  3. After clearing the upstream fault resynchronize the microgrid to the main grid.
Simulation Setup

Figure: Single line diagram of power system model designed in Simulink/sim power system and interfacing with agent system
Multi Agent System for Self Healing System

Proposed MAS

Figure: Schematic diagram of agent
Multi Agent System for Self Healing System

Grid Connected Mode of Operation

Figure: Test circuit for the grid connected operation
Multi Agent System for Self Healing System

Grid Connected Mode of Operation

Figure: Simulation result illustrating smooth voltage and current output at the interconnection point (bus B2) with the change in DER power output.
Islanded Mode of Operation

Figure: Test circuit for the islanded operation
Islanded Mode of Operation

Figure: Simulation result illustrating smooth voltage and current output at the interconnection point
Securing Critical Loads During an Emergency

- **During the Grid-Connected Mode (before 5PM).**
- **During the Transition (at 5PM).**
- **During the Islanded Mode (5PM-8PM).**

Figure: The variation of 60Hz voltage and current waveforms measured at the interconnection point before and after the upstream outage applied to the circuit
Securing Critical Loads During an Emergency

- **During Resynchronization to the Main Grid (at 8PM)**

Figure: Simulation results during the synchronization of the microgrid to the main grid
Conclusions

- **Self-healing**
  
  Agent protocols have been developed and a microgrid multi agent System simulation that demonstrates performance during a casualty requiring self-healing has been implemented. The power system self-healing is difficult, but the case is made for combining multi-agent control with super conducting fault current limiter and micro grids to this end.
Acknowlegdement

I thank my researcher students for their research contributions used in the presentation
Multi Agent System for Self Healing System

Thanks

Q & A
Short Term Course

Advanced Optimization Techniques

Day 5 : (22-May-2015)
Scientific Paper Writing

Dr. Rajesh Kumar

PhD, PDF (NUS, Singapore)
SMIEEE, FIETE, MIE (I), SMIACSIT, LMISTE, MIAENG

Associate Professor, Department of Electrical Engineering
Associate Professor, Centre of Energy and Environment
Malaviya National Institute of Technology, Jaipur, India, 302017
Tel: (91) 8764231105
rkumar@ieee.org, rkumar.ee@mnit.ac.in
http://www.mnit.ac.in
http://drrajeshkumar.wordpress.com
• National scientific journal
• International scientific journal
• (general or important)
• SCI/ EI/ ISTP
• SCI: Scientific Citation Index
• EI: Engineering Index
• ISTP: Index of Scientific & Technical Proceedings
Outline

• Brief introduction of scientific paper
• 2. Paper writing
• 3. Paper submitting
• 4. Common mistakes appeared in paper
• 5. Brief introduction of “final examination”
Directed Bee Colony Optimization Algorithm

Rajesh Kumar

Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur 302017, India

Abstract

The paper presents a new optimization algorithm inspired by group decision-making process of honey bees. The honeybees search for the best nest site among many possible sites taking care of both speed and accuracy. The nest site selection is analogous to finding the optimality in an optimization process. Such similarities between two processes have been used to cultivate a new algorithm by learning from each other. Various experiments have been conducted for better understanding of the algorithm. A comprehensive experimental investigation on the choice of various parameters such as number of bees, starting point for exploration, choice of decision process etc. has been made, discussed and used to formulate a more accurate and robust algorithm. The proposed Directed Bee Colony algorithm (DBC) has been tested on various benchmark optimization problems. To investigate the robustness of DBC, the scalability study is also conducted. The experiments conducted clearly show that the DBC generally outperformed the other approaches. The proposed algorithm has exceptional property of generating a unique optimal solution in comparison to earlier nature inspired approaches and therefore, can be a better option for real-time online optimization problems.

Keywords:
- Genetic Algorithm (GA)
- Evolutionary Algorithm (EA)
- Particle Swarm optimization (PSO)
- Harmony Search (HS)
- Differential Evolution (DE)
- Artificial Bee Colony (ABC)
Introduction

1. Introduction

Nature is a vast source of information which is being used by various species for their survival. The colonies of social insects and species can adapt themselves according to the change in environment. The colonies of ant and bees are the best examples of such cases. The group decision-making process used by bees, ants, birds, fishes and swarms for searching out the best food resources among the various possible solutions are the best and robust examples of swarm-based decision methods. These group decisions processes can be mimicked and used for finding out solution for various optimization problems as well as their applications in various engineering fields \[32,37\].
Main Text

2. Working principle of bee in nature

In the last couple of years, the researchers have paid great attention to study of the behavior of social insects in an attempt to use the Swarm Intelligence concept for developing various artificial systems. Group decisions taken by natural swarms are robust in all senses, as these not only give the best and accurate result but also save a lot of swarm’s energy.
4. Experiments and results

In order to determine the impact of control parameters, the author has conducted full factorial experiments for the proposed DBC algorithm for the nine benchmark problems. These benchmark problems are summarized in Tables 1 and 2. The benchmark functions in this section provide a balance between multi-modal with many local minima and functions with only a few local minima as well as easy and difficult functions.

This section is divided into three parts. First, impact of bee decision making on the performance of DBC is studied with experimental results. In the second part, the impact of the number of bees and starting point on the performance of DBC is compiled and discussed. In the third section, the algorithm has been tested
5. Conclusion

The paper first analyzes the nature inspired algorithms and discusses their known problems of consistency in solution and of premature phenomenon. A new optimization algorithm DBC has been proposed and described. The upshot of the proposed algorithm is that it generates better optimal solutions as compared to its counterparts. The algorithm is based on natural swarm group decision method by bees to find next site. One has to select
References


Structure of Research Article

1. Title
2. Authors’ name (including the author’s affiliation)
3. Abstract (including 3~8 keywords)
4. Main text
5. Acknowledgement
6. References
7. Appendix
Paper Writing- Title

• Short sentence, phrase, generally less than 20 words
• The research object (or problem) should be included in the title
• Reflect the main outcome and the approach used in your research work

Title: Meaningful and brief

Directed Bee Colony Optimization Algorithm

is better than

A Novel Nature Inspired Algorithm Incorporating Bee Behavior with Deterministic Frame work for Optimization Problems
Bee Inspired Optimization Algorithm is Proposed with Modifications
Paper Writing - Author’s Name

Contribution: The name of the authors, with all initials, the institute or organization, with full address.

Example:

Rajesh Kumar  
Department of Electrical Engineering  
Malaviya National Institute of Technology  
Jaipur, India, 302017
Paper Writing - Abstract

• A short summary of a longer article
• Written after the paper is completed, although it is intended to be read first
• Appears on a separate page just after the title page and therefore just before the essay itself
Paper Writing - Abstract

• Decide whether to read an entire article
• Remember key findings on a topic
• Understanding a text
• Index articles
The paper presents a new optimization algorithm inspired by the group decision-making process of honey bees. The honeybees search for the best nest site among many possible sites, taking care of both speed and accuracy. The nest site selection is analogous to finding the optimality in an optimization process. Such similarities between two processes have been used to cultivate a new algorithm by learning from each other. Various experiments have been conducted for a better understanding of the algorithm. A comprehensive experimental investigation on the choice of various parameters such as number of bees, starting point for exploration, choice of decision process etc. has been made, discussed and used to formulate a more accurate and robust algorithm. The proposed Directed Bee Colony algorithm (DBC) has been tested on various benchmark optimization problems. To investigate the robustness of DBC, the scalability study is also conducted. The experiments conducted clearly show that the DBC generally outperformed the other approaches. The proposed algorithm has exceptional property of generating a unique optimal solution in comparison to earlier nature inspired approaches and therefore, can be a better option for real-time online optimization problems.
Paper Writing - Main Text

Introduction:

• What is the problem and why is it interesting?
• Who are the main contributor?
• What did they do?
• What novel thing will you reveal?
Paper Writing - Main Text

Nature is a vast source of information which is being used by various species for their survival. The colonies of social insects and species can adapt themselves according to the change in environment. The colonies of ant and bees are the best examples of such cases. The group decision making process used by bees, ants, birds, fishes and swarms for searching out the best food resources among the various possible solutions are the best and robust examples of swarm-based decision methods. These group decisions processes can be mimicked and used for finding out solution for various optimization problems as well as their applications in various engineering fields [32,37].

A lot of classical methods have been developed and are being used for optimization problem. Golden section search, Fibonacci search, Newton's method and secant method are some one dimensional search methods. Gradient methods, Newton's method, Conjugate direction method and Neural Networks are commonly used for unconstrained optimization [4]. These methods are problem specific and use gradients. Consequently, they are applicable to a much smaller classes of optimization [26].
In past decade a vast research has been carried out on the development of nature-inspired optimization algorithms which include Genetic Algorithm (GA), Evolutionary Algorithm (EA), Particle Swarm optimization (PSO), Harmony Search (HS), Differential Evolution (DE), Bacterial Foraging Optimization (BFO) and Artificial Bee Colony (ABC) [7,16,1,37]. These algorithms are population based algorithms, a class of meta-heuristics. Population based algorithms trust on iteratively updating a population of candidate solution, while foraging algorithm differs at method espoused for updating and search patterns [32].

Evolution Algorithm (EA) is inspired by the theory of evolution by means of natural selection. Specifically, the technique is inspired by macro-level or the species-level process of evolution (phenotype, hereditary, variation) and is not concerned with the genetic mechanisms of evolution as in case of GA [15,16]. Tournament feature selection with directed mutation has been developed and concluded that such hybrid algorithms overcome some disadvantages of existing one and proves better solvers [9].
2. Working principle of bee in nature

In the last couple of years, the researchers have paid great attention to study of the behavior of social insects in an attempt to use the Swarm Intelligence concept for developing various artificial systems. Group decisions taken by natural swarms are robust in all senses, as these not only give the best and accurate result but also save a lot of swarm’s energy.

In nature, initially some bees go for searching the nectar sources. Upon arrival on nectar site, bee loads a lot of nectar and returns to hive relinquishing the nectar or pollen to store bee and go to “dance floor” to perform waggle dance [29]. This dance gives precise information to other bees regarding the quality, distance and direction of flower patch. Each individual’s knowledge of outside
Paper Writing- Main Text

• Experimental procedure or theoretical analysis
• Usually a list of all materials or equipments you used for the experiment
• The theoretical model used in your analysis
• list all steps in the correct order
• Experimental paper: equipment, materials, method
• Modeling paper: assumptions, mathematical tools, method
• Computational paper: inputs, computational tools, method
• Explain what is especially different about your method
• Give sufficient detail that the reader
• Don’t mix method with results or discussion
Paper Writing - Main Text

achieve goals. Following are the assumptions made in the proposed algorithm:

• Bees live and act in a given environment.
• Bees attempt to achieve particular goals or perform particular tasks.
• Information exchange process accompanies no losses.
• Bees will never die and hence number of bees remains constant during the process.

Definition 1. An exploration region \( \mathcal{X} \in \mathbb{R}^d \), \( d = 1 \ldots n \) is a bounded region that represents a feasible solution to some problem.

Definition 2. A grid on FSR, \( G = \{ E_i \}_{i=1}^N \) is defined as a set of elements, \( E_i \), such that \( E_i \in \mathbb{R}^d \), \( E_i \cap \bar{E}_j = \emptyset \), \( i \neq j \) and \( \cup E_i = \text{FSR} \), where \( \bar{E}_i \) denotes the interior of \( E_i \) and \( \emptyset \) is the empty set.

Definition 3. A sub exploration region \( E_i = \text{FIV}^m \) is a vector of \( m \) integers that represents feasible solutions to the problem.
FLOWCHART FOR DIRECTED BEE COLONY ALGORITHM

1. Define parameters:
   - f(x): Objective Function
   - N: Number of Variables
   - X_U (l): Lower bound
   - X_L (l): Upper bound
   - Step Size
   - BEEE: Number of Bees
   - Starting Point X_0

2. Method to be used
   - Quorum
   - Consensus

3. For BEEE:
   - If BEEE ≤ 1, obtain the optimum solution for each bee
   - Otherwise, obtain the optimum solution for each bee

4. For communication:
   - Communicate solution through Waggle Dance
   - Quorum of ith bee: Quorum Threshold
   - Global optimum solution = Min(Solution of all Bees)

5. Global optimum solution = Solution of ith bee

Fig. 6: Flowchart for the proposed DBC algorithm.
4. Experiments and results

In order to determine the impact of control parameters, the author has conducted full factorial experiments for the proposed DBC algorithm for the nine benchmark problems. These benchmark problems are summarized in Tables 1 and 2. The benchmark functions in this section provide a balance between multi-modal with many local minima and functions with only a few local minima as well as easy and difficult functions.

This section is divided into three parts. First, impact of bee decision making on the performance of DBC is studied with experimental results. In the second part, the impact of the number of bees and starting point on the performance of DBC is compiled and discussed. In the third section, the algorithm has been tested
Paper Writing - Main Text

4.1. Impact of bee decision making on the performance of DBC

Experiments have been carried out to compare the two decision making process in the bees for both consensus as well

4.2. Impact of parameters on the performance of DBC

The solution provided by DBC are not same throughout, it depends upon its step size and the range of parameters. Different

4.3. Comparison with other similar algorithms

The proposed DBC algorithm has been tested for a number of standard optimization problems of different classes and results show the robustness and advantage of proposed algorithm over
Paper Writing- Main Text

- Discussion:
- Extract principles, relationships, generalizations.
- Present analysis, model or theory.
- Show relationship between the results and analysis, model or theory.
Figures

Flow charts show methods, procedures.

Graphs plot data

Schematics show how equipment works, or illustrate a mechanism or model

Drawings and photographs illustrate equipment, microstructures etc
Paper Writing - Main Text

**Fruits Picked**

<table>
<thead>
<tr>
<th>Kind of Fruit</th>
<th>Number Picked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apples</td>
<td>6</td>
</tr>
<tr>
<td>Bananas</td>
<td>10</td>
</tr>
<tr>
<td>Oranges</td>
<td>14</td>
</tr>
<tr>
<td>Limes</td>
<td>12</td>
</tr>
</tbody>
</table>

**Joe's Fruit Stand Sales**

<table>
<thead>
<tr>
<th>Fruit</th>
<th>Average Sales/Day</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apples</td>
<td>108.42</td>
<td>7.5%</td>
</tr>
<tr>
<td>Bananas</td>
<td>279.39</td>
<td>19.3%</td>
</tr>
<tr>
<td>Oranges</td>
<td>546.27</td>
<td>37.8%</td>
</tr>
<tr>
<td>Grapes</td>
<td>375.3</td>
<td>25.9%</td>
</tr>
<tr>
<td>Limes</td>
<td>137.61</td>
<td>9.5%</td>
</tr>
</tbody>
</table>
Paper Writing- Main Text
### Table 1: A comparison of FIFA World Cup 2002, 2006 and 2010 – the use of medication per match and per tournament.

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2006*</th>
<th>2002*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(n=2944) (%)</td>
<td>(n=736) (%)</td>
<td>(n=2944) (%)</td>
</tr>
<tr>
<td>Any medication</td>
<td>1418 (48.2%)</td>
<td>528 (71.7%)</td>
<td>1257 (47.7%)</td>
</tr>
<tr>
<td>NSAIDs</td>
<td>1020 (34.6%)</td>
<td>403 (54.8%)</td>
<td>855 (29.0%)</td>
</tr>
<tr>
<td>Injections*</td>
<td>96 (3.3%)</td>
<td>54 (7.3%)</td>
<td>103 (3.5%)</td>
</tr>
<tr>
<td>Analgesics</td>
<td>189 (6.4%)</td>
<td>109 (14.8%)</td>
<td>98 (3.7%)</td>
</tr>
<tr>
<td>β-2 agonists</td>
<td>58 (2.0%)</td>
<td>20 (2.7%)</td>
<td>31 (1.1%)</td>
</tr>
<tr>
<td>Antihistamines</td>
<td>64 (2.2%)</td>
<td>42 (5.7%)</td>
<td>106 (3.6%)</td>
</tr>
<tr>
<td>Any supplement</td>
<td>1019 (34.6%)</td>
<td>353 (48.0%)</td>
<td>1041 (35.4%)</td>
</tr>
</tbody>
</table>

* Corticosteroid and local anaesthetic injections only.  
NSAID, non-steroidal anti-inflammatory drugs.
Paper Writing- Main Text

Conclusions

• Must have:
  • Summarize the study work
  • State outcome (or conclusions) of the research

• May have:
  • Make suggestions of further work
  • Draw together the most important results and their consequences.
  • List any reservations or limitations.
  • Don’t duplicate the Abstract as the Conclusions or vice versa.
  • Abstract is an overview of the entire paper
  • Conclusion is a summing up of the advances in knowledge
optimal results. A set of benchmark functions have been used to test DBC in comparison with GA, EA, PSO, BFO, HS, DE and ABC for both lower dimension and higher dimension. Experimental results prove the robustness and accuracy of DBC over other search based approaches. The results show that DBC performs better and also removes the randomness in the algorithm. All other evolutionary algorithms generate different solutions on different runs whereas DBC generates unique optimum solution. Hence it gives a better option to optimize real-time and on-line optimization problems. Although DBC has proven better algorithm over presented algorithms but other classes of improved, mutated, directed and hybrid algorithms like EPSO, DM-GA, HSDM, EPSDE are still to be compared and will be agenda for future research.
Acknowledgement

• The author states and expresses the acknowledge for the financial support.
• Thank people who have helped you with ideas, technical assistance, materials or finance.
• Keep it simple, give full names and affiliation, and don’t get sentimental.
References

• Provide a reference list at the end of the article or chapter to supplement textual notes; the reference list gives complete citation for all works cited.

• References must be complete: name, initials, year, title, journal, volume, start page and finish page.


Appendix

• Generally, the appendix is adopted where the content does not directly relate to the discussed topic but useful for understanding the study work.

• Equation derivation and experimental data often included in appendix
Paper Writing

1. The design
2. The market – who are your readers?
3. The concept – making a concept sheet
4. Embodiment – the first draft
5. Detail – grammar, spelling, punctuation, style
# Market for Technical Writing

<table>
<thead>
<tr>
<th>What Writing?</th>
<th>Who are the readers?</th>
<th>How will they use it?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thesis</td>
<td>Examiners</td>
<td>To judge and rank your work</td>
</tr>
<tr>
<td>Paper</td>
<td>Referees</td>
<td>To check originality, quality, suitability</td>
</tr>
<tr>
<td></td>
<td>Scientifically literate public</td>
<td>To extract information</td>
</tr>
<tr>
<td>Research proposal</td>
<td>The funding body and….. Its referees</td>
<td>To judge if your aims match the priorities of the funding body To judge quality and promise of the work</td>
</tr>
<tr>
<td>Popular article</td>
<td>Intelligent but uninformed public</td>
<td>To be introduced to a new field To be entertained</td>
</tr>
</tbody>
</table>
Look Professional

- Grammar: words, sentence structure
- Punctuation: full stop, comma, question mark
- Style: clear, define everything

Consistent in
- Font size
- Margins
- Spacing
- Table size and format
- Figure size and format
- Neat and Clean!
Acknowledgement

I thank Wei-Ze Wang, Ph. D
School of Mechanical & Power Engineering
East China University of Science and Technology
for his research, insight and academic contributions used in the presentation
Thanks

Q & A
Technical Writing Using LaTeX

Dinesh Gopalani
Deptt. Of CSE, MNIT Jaipur
dgopalani.cse@mnit.ac.in
What is LaTeX?

- LaTeX is a document markup language used for writing documents.
- LaTeX was originally written in the early 1980s by Leslie Lamport at SRI International.
- It has become the standard for writing Technical and Scientific documents.
- It is freely available for most of the platforms.
Why LaTeX?

• We may concentrate on Contents of the paper rather than its visual appearance.
• The output produced is equivalent to that of published books.
• Facilitates for the automatic numbering of chapters, sections, theorems, equations etc
• Complex Mathematical equations and formulae can be written with ease.
Why LaTeX? (Contd.)

• If you write a research paper and wish to get it published. Each publisher will have its own layout of how it wants things. If you write it in Latex you only need to change the class/style file to set it into their format.

There is no need of rewriting the paper.
Paper in different Formatting Styles
LaTeX – The Steps ...

Your favourite text editor

Your favourite text editor

Text file with embedded instructions

file.tex

Device Independent Document

latex

file.tex

file.dvi

Some other printer format (eg. Postscript)

Acrobat Reader

PDF Document

dvi2pdf

file.dvi

file.pdf

AOT'15, MNIT, Jaipur, 18-22 May 2015
How to Write LaTeX Code?

- Commands start with a backslash (\).
- Parameters are given in curly brackets \{\}.
- Optional parameters are given in square brackets [ ].
- Environments (blocks with a certain type of content) are of the form:

  \begin{environment_type}
  environment content
  \end{environment_type}
A Simple Document

\documentclass{article}
\begin{document}
...
\end{document}
A Simple Document

\documentclass{article}
\begin{document}

This specifies the type of the document: article, report, book, letter, etc.

Here we write the contents.

\end{document}
Standard Document Classes

- **Article**: Ideal for a short research paper (divided into sections, subsections, etc).
- **Book**: Class to be used to typeset a book (chapters, sections, etc).
- **Report**: Similar to the Book class but for single-sided printing.
- Other classes include **letter, slides**, etc.
Organization of the Paper

• A paper is typically split into the following logical parts:
  – A title
  – An abstract
  – A number of sections
  – A number of subsections in each section
  – A number of subsubsections in each subsection
  – A number of references
(Also we may have mathematical equations, figures and tables, etc.)
The Title

\documentclass{article}
\begin{document}

\title{Technical Writing Using LaTeX}
\author{Dinesh Gopalani}
\date{May 22, 2015}

\maketitle

\end{document}
Technical Writing Using \LaTeX

Dinesh Gopalani

May 22, 2015
The Abstract

• Used to give an overview of the content of the document.
• Is usually typeset with wider margins than the main text.
• Specified using the abstract environment:
  \begin{abstract}
  ...
  \end{abstract}
Sections and Subsections

\documentclass{article}
\begin{document}
\section{This is a Section}
\subsection{This is a Subsection}
This is body of subsection.
\subsection{This is another Subsection}
\section{This is another Section}
\end{document}
There is a Section

This is body of subsection.

1. This is a Subsection

This is body of subsection.

1.1 This is a Subsection

1.2 This is another Subsection

2. This is another Section
Figures and Tables

• The figure environment is used to include a floating figure in the text.
• Similarly the table environment can be used to insert a floating table.
• A caption can be added to both using the \caption{} command.
• The two environments are similar except for the caption title, and whether they appear in a list of figures, or the list of tables.
Figures

\begin{figure}
\includegraphics{latex.jpeg}
\caption{A Book on Latex}
\end{figure}
Figures (Contd.)

\begin{figure}
\includegraphics{latex.jpeg}
\caption{A Book on Latex}
\end{figure}

Figure 1. A Book on Latex
Figures (Contd.)

\begin{figure}
\includegraphics{latex.jpeg}
\caption{A Book on \LaTeX}
\end{figure}

Number is assigned automatically

Figure 1. A Book on \LaTeX
Tables

• To draw Tables, the tabular environment is used.

• Parameters give the information about the column layout. (l – Left, c – Centre, r – Right Alignments, | - Vertical Line, etc.)

• Separate lines using \ and columns using the Ampersand (&) symbol.

• \hline draws a Horizontal Line.
Tables (Contd.)

\begin{tabular}{|l|c|r|}
\hline
Name & Pos & Score \\
Suresh Sharma & 3rd & 90 \\
Pradeep Agarwal & 2nd & 94 \\
Amit Verma & 1st & 98 \\
\hline
\end{tabular}
\begin{tabular}{|l|c|r|}
\hline
Name & Pos & Score \\
\hline
Suresh Sharma & 3rd & 90 \\
\hline
Pradeep Agarwal & 2nd & 94 \\
\hline
Amit Verma & 1st & 98 \\
\hline
\end{tabular}

Left Alignment for First Column

Center Alignment for Second Column

Right Alignment for Third Column
\begin{tabular}{|l|c|r|}
\hline
Name & Pos & Score \\
\hline
Suresh Sharma & 3rd & 90 \\
Pradeep Agarwal & 2nd & 94 \\
Amit Verma & 1st & 98 \\
\hline
\end{tabular}
Mathematical Symbols and Equations

• All mathematics must appear in maths mode.
• The following symbols can be produced using the commands:
  \leq \times \pi \infty

• If a mathematical expression appears in a line of normal text, use a dollar symbol $ to start and to end the mathematics.
Mathematical Symbols and Equations (Contd.)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>LaTeX Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neq$</td>
<td>\neq</td>
</tr>
<tr>
<td>$\leq$</td>
<td>\leqslant</td>
</tr>
<tr>
<td>$\geq$</td>
<td>\geqslant</td>
</tr>
<tr>
<td>$\approx$</td>
<td>\approx</td>
</tr>
<tr>
<td>$\equiv$</td>
<td>\equiv</td>
</tr>
<tr>
<td>$\cong$</td>
<td>\cong</td>
</tr>
<tr>
<td>$\simeq$</td>
<td>\simeq</td>
</tr>
<tr>
<td>$\partial$</td>
<td>\partial</td>
</tr>
<tr>
<td>$\infty$</td>
<td>\infty</td>
</tr>
<tr>
<td>$\nabla$</td>
<td>\nabla</td>
</tr>
<tr>
<td>$\aleph$</td>
<td>\aleph</td>
</tr>
<tr>
<td>$\ell$</td>
<td>\ell</td>
</tr>
<tr>
<td>$\vee$</td>
<td>\vee</td>
</tr>
<tr>
<td>$\wedge$</td>
<td>\wedge</td>
</tr>
<tr>
<td>$\forall$</td>
<td>\forall</td>
</tr>
<tr>
<td>$\exists$</td>
<td>\exists</td>
</tr>
<tr>
<td>$\pm$</td>
<td>\pm</td>
</tr>
<tr>
<td>$\mp$</td>
<td>\mp</td>
</tr>
<tr>
<td>$\times$</td>
<td>\times</td>
</tr>
<tr>
<td>$\div$</td>
<td>\div</td>
</tr>
<tr>
<td>$\cup$</td>
<td>\cup</td>
</tr>
<tr>
<td>$\cap$</td>
<td>\cap</td>
</tr>
<tr>
<td>$\in$</td>
<td>\in</td>
</tr>
<tr>
<td>$\notin$</td>
<td>\notin</td>
</tr>
<tr>
<td>$\rightarrow$</td>
<td>\rightarrow</td>
</tr>
<tr>
<td>$\iff$</td>
<td>\iff</td>
</tr>
<tr>
<td>$$</td>
<td>$</td>
</tr>
<tr>
<td>£</td>
<td>\pounds</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>{</td>
<td>{</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
<tr>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>\ast</td>
<td>\ast</td>
</tr>
<tr>
<td>\dag</td>
<td>\dag</td>
</tr>
<tr>
<td>\ddag</td>
<td>\ddag</td>
</tr>
<tr>
<td>\bullet</td>
<td>\bullet</td>
</tr>
<tr>
<td>\wr</td>
<td>\wr</td>
</tr>
<tr>
<td>\copyright</td>
<td>\copyright</td>
</tr>
<tr>
<td>\maltese</td>
<td>\maltese</td>
</tr>
</tbody>
</table>
A method closure $\langle \varsigma(x_i)b_i, S_{e_i} \rangle$ is a pair consisting of method together with a stack that is used for the reduction of the method body.
A method closure $\langle \varsigma(x_i)b_i,S_{e_i}\rangle$ is a pair consisting of method together with a stack that is used for the reduction of the method body.
Mathematical Symbols and Equations (Contd.)

- If mathematical formulae are to appear on a separate line, start the mathematics using \[, and end it with \].
- For Equations, use equation environment.

\begin{equation}
\Gamma, x_i:\Sigma_s^\prime(iota_i)\vdash b_i:\Sigma_b^\prime(iota_i), \forall i \in 1..q
\end{equation}
Mathematical Symbols and Equations (Contd.)

• If the mathematical formulae are to appear on a separate line, start the mathematics using \[, and end it with \].

• For Equations, use the equation environment.

\begin{equation}
\Gamma, x_i : \Sigma_a^1(\tau_i) \vdash b_i : \Sigma_b^1(\tau_i), \forall i \in 1..q
\end{equation}
References (Bibliography)

- To create the bibliography, use the thebibliography environment.
- Items in the bibliography are added using the \bibitem{label} command. The label is used to refer to the entry.
- Citing a bibliography item in the main text can be done using the \cite{label} or \cite{label1, label2,...} command to obtain citations such as [2] or [7,4].
References (Bibliography) (Contd.)

\begin{thebibliography}
\bibitem{label1} bibliographic information
\bibitem{label2} bibliographic information
...
\end{thebibliography}
References (Bibliography) (Contd.)

\begin{thebibliography}
\bibitem{label1} bibliographic information
\bibitem{label2} ...
\end{thebibliography}

References


Table of Contents

• To add a table of contents with chapters, sections, etc. the command `\tableofcontents` is used.

• We may also include a list of figures and a list of tables by using `\listoffigures` and `\listoftables` respectively.
Table of Contents (Contd.)

\documentclass[12pt]{report}
\begin{document}
\tableofcontents
\listoffigures
\listoftables
....
\end{document}
Table of Contents (Contd.)

\documentclass[12pt]{report}
\begin{document}
\tableofcontents
\listoffigures
\listoftables
\listofnotations
\end{document}
Labels and Cross References

• To name a numbered object (figure, section, chapter, etc) the command \label{label-name} is used.

• Now in the text to insert the number of the object named using \label command, \ref{label-name} is used at that place.
Labels and Cross References (Contd.)

The calculus consists of objects and aspects, and define the operations. Table \ref{syntax-un-asp} summarizes syntax for the aspect calculus. The conventions used for representing the syntax are based on standard Backus-Naur Form (BNF) \cite{cho.56, knu.64}.

\begin{tabular}...
\end{tabular}

\label{syntax-un-asp}

\begin{tabular}...
\end{tabular}
The calculus consists of objects and aspects, and define the operations. Table \ref{syntax-un-asp} summarizes syntax for the aspect calculus. The conventions used for representing the syntax are based on standard Backus-Naur Form (BNF) [25, 60].
Final Words ....

- Your journey with LaTeX already started by attending this session.
- Enjoy your writing with LaTeX
- Spread the joy of using LaTeX.