Genetic Algorithms: Solution Representation

Andrew Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527
Tel: 319 - 335 5934 Fax: 319-335 5669
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Set Covering Problem

\[
\begin{align*}
\text{Min} & \quad \sum cjx_j \\
\sum a_{ij} & \geq 1 \quad \text{for all } i, j \\
x_j & = 0, 1 \quad \text{for all } j
\end{align*}
\]

• Scheduling
• Process planning

Reference 1

Set Covering Problem

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
5 & 3 & 2 & 4 & 8
\end{bmatrix}
\]

Decision variable
Matrix
Cost

Mathematical programming representation

\[
x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 0
\]

Genetic representation

\[
[1, 0, 1, 0, 0]
\]

Bin Packing Problem

Placing n objects into at most n bins

• Container packing
• Space shuttle spatial layout
• CD ROM packing

Example 2
Bin Packing Problem

\[
\begin{align*}
\text{Min} & \sum y_j \\
\sum w_j x_{ij} & \leq cy_i \quad \text{all } i \\
\sum x_{ij} &= 1 \quad \text{all } j \\
y_i &= 0, 1 \\
x_{ij} &= 0, 1
\end{align*}
\]

Where:
- \( w_j \) = weight of object \( j \)
- \( c \) = bin capacity
- \( x_{ij} = 1 \) if object \( j \) assigned to bin \( i \), \( = 0 \) otherwise
- \( y_i = 1 \) if bin \( i \) is used, \( = 0 \) otherwise

Bin-based representation

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Chromosome:
1. Object 1 in Bin 1, Object 2 in Bin 4, Object 3 in Bin 2
2. ..., Object 6 in Bin 2

Object-based representation

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Chromosome:
1. Bin 1 - the first three objects
2. Bin 2 - the next three objects
3. Bin 3 - the last object

Group-based representation

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Using the above equivalence scheme, the chromosome ADBCBE implies A = \{1\}, B = \{3, 6\}, C = \{4\}, D = \{2\}, and E = \{5\}
Group-based Representation

ADBCEB implies $A = \{1\}$, $B = \{3, 6\}$, $C = \{4\}$, $D = \{2\}$, and $E = \{5\}$,
represented as

3, 6 5 4 2 1 Object

B E C D A Bin

Advantages

• Most suitable of all three methods
• Genes represent both objects and groups (bins)

• Chromosomes of variable length

Clustering Problem

Multitude of formulations, ranging from
• binary matrix to
• mathematical programming
implies

Many possible solution representations
in genetic programming

Order-based Representation 1 (2)

Consider matrix with
• three clusters, and
• seven features (or objects)

Note: number of object clusters = number of feature clusters

\[
\begin{bmatrix}
1 & 2 & 7 & 4 & 3 & 16 & 25 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

Chromosome

Order-based Representation 2 (2)

Mutation:
• Random permutation

Crossover*:
• Partially mapped crossover (PMX)
• Order crossover (OX)
• Cycle crossover (CX)
• Recombination crossover (ER)

* Reference 1
Project Scheduling 1(7)

Project Activity Network

Project Scheduling 2(7)

Solution representation

Position: Activity ID
Value: Priority of activity

Project Scheduling 3(7)

Topologically sorted graph

Project Scheduling 4(7)

Level 1: 1
Level 2: 2, 3, 4
Level 3: 5, 6
Level 4: 7

Project Scheduling 5(7)

Topologically sorted graph

Project Scheduling 6(7)

Topologically sorted graph

Triangularization algorithm
http://www.icaen.uiowa.edu/%7Eankusiak/process-model.html

Triangularization algorithm
http://www.icaen.uiowa.edu/%7Eankusiak/process-model.html
Project Scheduling 7(7)

Topologically sorted graph

Position-based Crossover

Random genes from one parent are transferred to a child
Missing genes are filled left-to-right from the other parent

Swap Mutation

Parent 1 3 1 7 6 4 5 2
Child 6 5 7 1 4 2 3

Local Search-based Mutation

Parent chromosome 3 1 7 6 4 5 2
Pivot gene 3 4 7 6 1 5 2
Neighborhood (4, 1) (4, 7)
(4, 2) (4, 6)

Fitness Function

Objective function = Min finish time of the last network activity

Min problem transformed to a Max problem so that the fitter individual correspond higher value of the fitness function

Fitness Function

Objective function = Min finish time of the last network activity

Min problem transformed to a Max problem so that the fitter individual correspond higher value of the fitness function

Fitness function g

The selection method based on the fitness function g changes with γ from proportional to random
References

