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

Min $\Sigma c_j x_j$
$\Sigma a_{ij} \geq 1$ for all $i, j$

$x_j = 0, 1$ for all $j$

Applications
- Scheduling
- Process planning

Reference 1

Set Covering Problem

Example 1

$[x] \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}$

Decision variable

$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \end{bmatrix}$

Matrix

$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 \end{bmatrix}$

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

Cost

Mathematical programming representation

$x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 0$

Genetic representation

$[1, 0, 1, 1, 0]$
Bin Packing Problem

Model
Min $\sum y_j$
$\sum w_j x_{ij} \leq c_i$ all i
$\sum x_{ij} = 1$ all j
$y_i = 0, 1$
$x_{ij} = 0, 1$

$w_j =$ 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

• Equal length chromosomes; use of standard
generic operators
• Redundant representation, e.g.,
$1 2 3 4 5$ and $3 4 5 2 2$ are identical solutions
(for equal capacity containers)

In both cases items 5 and 6 are stored in one container;

Object-based Representation

• Equal length chromosomes; use of standard
generic operators

Redundant representation, e.g.,
$1 2 3 4 5 \mid 6$, $3 2 1 4 5 \mid 6$, and $4 5 2 1 3 \mid 6$ are identical solutions

Group-based Representation

Using the above equivalence scheme,
the chromosome ADBCEB 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

Most suitable of all three methods
Genes represent both objects and groups (bins)
Chromosomes of variable length

Clustering Problem

Example

<table>
<thead>
<tr>
<th>Features</th>
<th>Feature Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td>1 3 2 4 5</td>
</tr>
</tbody>
</table>

Object

1 1 1 1
1 1 1 1
1 1
1 1

Object Group

2 4 1 1
1 1
1 1
1 1

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

\[ \begin{array}{cccc}
2 & 7 & 4 & 3 & 1 & 6 & 2 & 5 \\
\end{array} \]

Chromosome

Mutation:
- Random permutation

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

* Reference 1
Project Scheduling 1

1.2.3.4.5.6.7

Position: Activity ID
Value: Priority of activity

Project Activity Network

1.2.3.4.5.6.7

Topologically sorted graph

Eligible activities: 2, 3, 4

Project Scheduling 2

1.2.3.4.5.6.7

Position: Activity ID
Value: Priority of activity

Project Scheduling 3

1.2.3.4.5.6.7

Topologically sorted graph

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

Project Scheduling 4

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

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

Project Scheduling 5

1.2.3.4.5.6.7

Topologically sorted graph

Eligible activities: 2, 3, 4

Project Scheduling 6

1.2.3.4.5.6.7

Eligible activities: 3, 4, 5

Topologically sorted graph
Project Scheduling 7(7)

1 2 3 4 5 6 7

1 6 4 5 2

3 7 1

Topologically sorted graph

Position-based Crossover

Parent 1

3 1 7 6 4 5 2

Child

6 1 7 2 4 5 3

Parent 2

6 5 7 1 4 2

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

Swap Mutation

Parent

3 1 7 6 4 5 2

Child

3 4 7 6 1 5 2

Local Search-based Mutation

Parent chromosome

3 1 7 6 4 5 2

Pivot gene

3 4 7 6 1 5 2

Neighborhood

3 1 4 6 7 5 2

(4, 1)

3 1 7 6 2 5 4

(4, 7)

3 1 7 4 6 5 2

(4, 2)

Fitness Function

Objective function = Min finish time of the last network activity

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

f_{max}, f_{min} = \text{max, min value of the objective function in the current generation}
f = \text{objective function value in the current generation}
\gamma = \text{real positive number in (0, 1)}

Fitness function \( g \)

\[ g = \frac{(f_{max} - f + \gamma)(f_{max} - f_{min} + \gamma)}{f_{max} - f_{min} + \gamma} \]

The selection method based on the fitness function \( g \) changes with \( \gamma \) from proportional to random
References

