Multi-objective Evolutionary Algorithms: Applications and Technology

Lyndon While
lyndon@csse.uwa.edu.au

Walking Fish Group
The University of Western Australia
wfg.csse.uwa.edu.au
Overview

- Introduction to evolutionary algorithms
- Introduction to multi-objective optimisation
- WFG MOEA applications
- WFG MOEA technology
Optimisation and evolutionary algorithms

- An *optimisation problem* is one where the performance of a solution is measured on a continuous scale
  - usually don’t expect to find an optimal solution
  - “good enough, cheap enough, soon enough”

- A common technique for solving optimisation problems is *evolutionary algorithms*
  - population-based search technique where solutions accumulate good features through inheritance and mutation over a number of generations
  - “evaluation is easier than discovery”
An evolutionary algorithm in action

A maximisation problem

1

Evaluate

2

Select

3

Reproduce & mutate

4

5

6

7

8

MOEAs: Applications and Technology
Multi-objective optimisation

- A *multi-objective* optimisation problem is one where a solution is measured against more than one criterion
  - e.g. for vehicles: safety, performance, cost, maintainability, …
  - usually can’t optimise all criteria simultaneously

- An algorithm for solving a MOOP returns a set of solutions offering a range of trade-offs between the various criteria
  - e.g. a Hummer vs. a Volvo vs. a Porsche vs. a Daewoo

- Because of their population-based operation, EAs lend themselves very naturally to MOOPs
Two objectives, both being maximised

Each solution is plotted by its values in the objectives

\( a \) dominates \( b \) because it is better in every objective

\( b \) and \( c \) are mutually non-dominating

The \textit{rank} of a soln is the number of others that dominate it

Selection is based primarily on ranks
Having multiple objectives means that the fitness of each solution is a vector. For example, on the previous slide, a pair of numbers.

This complicates selection in MOEAs. There is no total ordering on fitnesses, hence the concepts of domination and ranks.

It also introduces diversity issues in objective space. We want to offer the client genuine choices.

It also makes comparing algorithms’ results difficult. Metrics have to compare sets, not just scalars.
Building a MOEA requires

- A *genetic representation* of a solution
  - captures what varies between solutions
  - omits features which are common to all solutions
- Several *objectives*
  - each quantified into a fitness function
- A selection process
- A reproduction process
- An *initialisation* procedure
  - randomisation vs. seeding
- Termination criteria
Mining is a huge business worldwide, and especially in Western Australia.

**Comminution** is a collection of physical processes used to reduce the sizes of particles in raw ore.

- comminution is performed by networks of crushing equipment

But designing an effective comminution network is hard:

- vast search space of potential solutions
- competing criteria to optimise
- inaccurate and slow models
- many infeasible solutions
- conservative engineers

Enter MOEAs:

- in conjunction with Rio Tinto
Focus here is solely on the design and operation of the cone crusher.

Variables:
- shapes of the crushing surfaces
- operational settings

Objectives:
- maximise capacity
- maximise product quality
Genetic representation of a cone crusher

- bowl liner
- mantle
- closed-side setting
- eccentric angle
- rpm
Two conflicting objectives

- Maximise the *capacity* of the circuit
  - not the same as maximising the capacity of the crusher:
    - also need to re-process recirculating material
  - easily achieved: crush very hard

- Maximise *product quality*
  - all output is < 32mm, but otherwise bigger is better
  - easily achieved: crush very gently

- Fitness calculations use a detailed physical model of the crusher operation, plus mass-balancing of the circuit
- Initialisation is from an existing good design
Can this MOEA generate good designs?

- Gen 200
- Gen 20
- Gen 4
- Gen 0

Normalized capacity vs. Normalised P80
Two example designs

A design with good quality

- CSS: 15.0
- Angle: 1.31
- RPM: 217

Normalised Capacity: 1.09
Normalised P80: 1.12

A design with good capacity

- CSS: 16.9
- Angle: 1.24
- RPM: 105

Normalised Capacity: 3.24
Normalised P80: 0.99
Other WFG/Rio mining projects

- Optimising unit selections and network layouts
  - comminution can be done by one big unit,
  - or by several units in parallel,
  - or by several units in sequence,
  - or by any combination of these

- Optimising for robustness
  - all industrial processes are “noisy”
  - coping with unexpected inputs is crucial
  - coping with wear is crucial
2D cutting involves placing a set of shapes onto a surface such that the shapes do not overlap, so that they can later be cut out. The usual objective is to minimise waste of the underlying material. A second important objective is to minimise the time required to cut out the shapes. Other objectives are also sometimes relevant. The problem extends straightforwardly to packing objects in a 3D volume and arranging (e.g.) events in an nD space. C&P has many applications in a range of industries.
A typical 2D problem instance

\[ N_{10} : 200 \text{ shapes, width } = 70, \text{ optimal height } = 150 \]
Two independent objectives

- Minimise waste
  - we minimise the height of the semi-infinite strip used by the packing

- Minimise processing time
  - we minimise the number of independent cuts required by the packing
What is a good genetic representation?

- A representation that simply lists the positions of the shapes is unlikely to work well
  - too many interdependencies
  - mutating a good solution will probably produce rubbish

- **Heuristic approaches** have worked well previously

- A heuristic takes a partial packing and a set of shapes yet to be placed, and it determines
  - which shape to place next, and
  - where to place it
Four alternative heuristics

- $n$ heuristic applications generates the packing
  - but no single heuristic performs well alone.

MOEA chooses B (rotated 90°)
(Nearest Height, Largest Width, Least Cuts Large, Least Cuts Small)
For a problem instance with \( n \) shapes, each solution’s genome is a sequence of \( n \) heuristics:
- e.g. for a problem with eight shapes

\[
<LCL_R, NH_R, LCL_L, NH_R, LW_L, NH_L, LCS_R, LCS_R>
\]

Applying the \( n \) heuristics in order gives the complete packing for that solution.

The population is initialised with a mixture of:
- randomly-generated sequences
- sequences that use the same heuristic \( n \) times
- sequences derived using a fast deterministic algorithm
Results on benchmark problems

- Two problem types
  - $RF$ allows 90° rotation of shapes, $OF$ doesn’t
- Three problem sets
- Each entry gives the average percentage waste
  - lower is better

<table>
<thead>
<tr>
<th></th>
<th>SPGAL</th>
<th>MOSP-D</th>
<th>MOSP-P</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1.93</td>
<td>6.07</td>
<td>0.91</td>
</tr>
<tr>
<td>N</td>
<td>—</td>
<td>4.45</td>
<td>0.69</td>
</tr>
<tr>
<td>B</td>
<td>2.71</td>
<td>2.31</td>
<td>1.30</td>
</tr>
<tr>
<td><strong>RF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.52</td>
<td>3.11</td>
<td>0.66</td>
</tr>
<tr>
<td>N</td>
<td>—</td>
<td>3.53</td>
<td>0.50</td>
</tr>
<tr>
<td>B</td>
<td>1.93</td>
<td>1.48</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Both objectives are being minimised.
Two example packings

N₅: 50 shapes, width = 100, optimal height = 100

1% waste, 55 cuts
3% waste, 44 cuts
Other WFG applications

- Sports scheduling
  - Australian Football League
  - Super 14 rugby
  - National Hockey League

- Spam-filtering

- Games
  - Spoof
  - Pac-Man
  - Robocup
  - Guess-It
  - Othello
  - Hnefatafl
  - Prisoner’s Dilemma
  - Knight’s Tours
Hypervolume is the most widely-used metric for comparing the results of MOO algorithms. The hypervolume of a set of solutions is the size of the part of objective space that they dominate collectively relative to a reference point. A larger hypervolume is taken to mean a better set.

Hypervolume captures in one number both the convergence and the spread of a set. Hypervolume has nicer mathematical properties than most other metrics. But hypervolume is expensive to calculate.
The hypervolume of the set is the size of the union of the four rectangles.

The reference point in all our examples is the origin.

Each solution dominates the rectangle between itself and the reference point.

The hypervolume of the set is the size of the union of the four rectangles.
The hypervolume of the set is the size of the union of the four cuboids
Algorithms for calculating hypervolume

- Old (slow) algorithms
  - Set union
  - LebMeasure

- Modern (faster) algorithms
  - HSO
    - FPL
    - IIHSO
  - HOY

- The latest (super-fast) algorithm
  - WFG
ihv(p) is the volume that is dominated by p

ehv(p, S) is the volume that is dominated by p but not by S

ehv(X, \{A, B, C, D\}) = ihv(X) - hv(\{A', B', C\}D')

Each Z dominates the intersection of Z and X

A' is dominated by B' and can be discarded
Hypervolume as a sum of exclusive volumes

$$hv(\{A,B,C,D\}) = ehv(D, \{\})$$
$$+ ehv(C, \{D\})$$
$$+ ehv(B, \{C,D\})$$
$$+ ehv(A, \{B,C,D\})$$
The WFG hypervolume algorithm

\[ hv([s_1, \ldots, s_m]) = \sum_{k=1}^{m} ehv(s_k, \{s_{k+1}, \ldots, s_m\}) \]

\[ ehv(p, \{s_1, \ldots, s_m\}) = ihv(p) - hv(nds(\{limit(p, s_1), \ldots, limit(p, s_m)\})) \]

\[ ihv([p_1, \ldots, p_n]) = p_1 \times \cdots \times p_n \]

\[ limit([p_1, \ldots, p_n], [s_1, \ldots, s_n]) = [\text{worse}(p_1, s_1), \ldots, \text{worse}(p_n, s_n)] \]

\[ nds(S) \quad \text{returns the non-dominated subset of} \quad S \]
WFG vs. other algorithms: random 7D data

Number of solutions

Time (seconds)

- HOY
- FPL
- IHSO
- WFG
WFG vs. other algorithms: random 7D data
Other WFG MOEA technology

- Use of hypervolume within MOEAs
- Multi-objective toolkit
- Constraints as objectives
- Noise
- Visualisation
**WFG personnel**

- Lyndon While
- Luigi Barone (now at SolveIT Software)
- Phil Hingston (Edith Cowan University)
- Lucas Bradstreet
- Simon Huband
- Anthony Di Pietro
- Numerous other students

- wfg.csse.uwa.edu.au
Any questions?