Hybrid Evolutionary Algorithms

Chapter 10

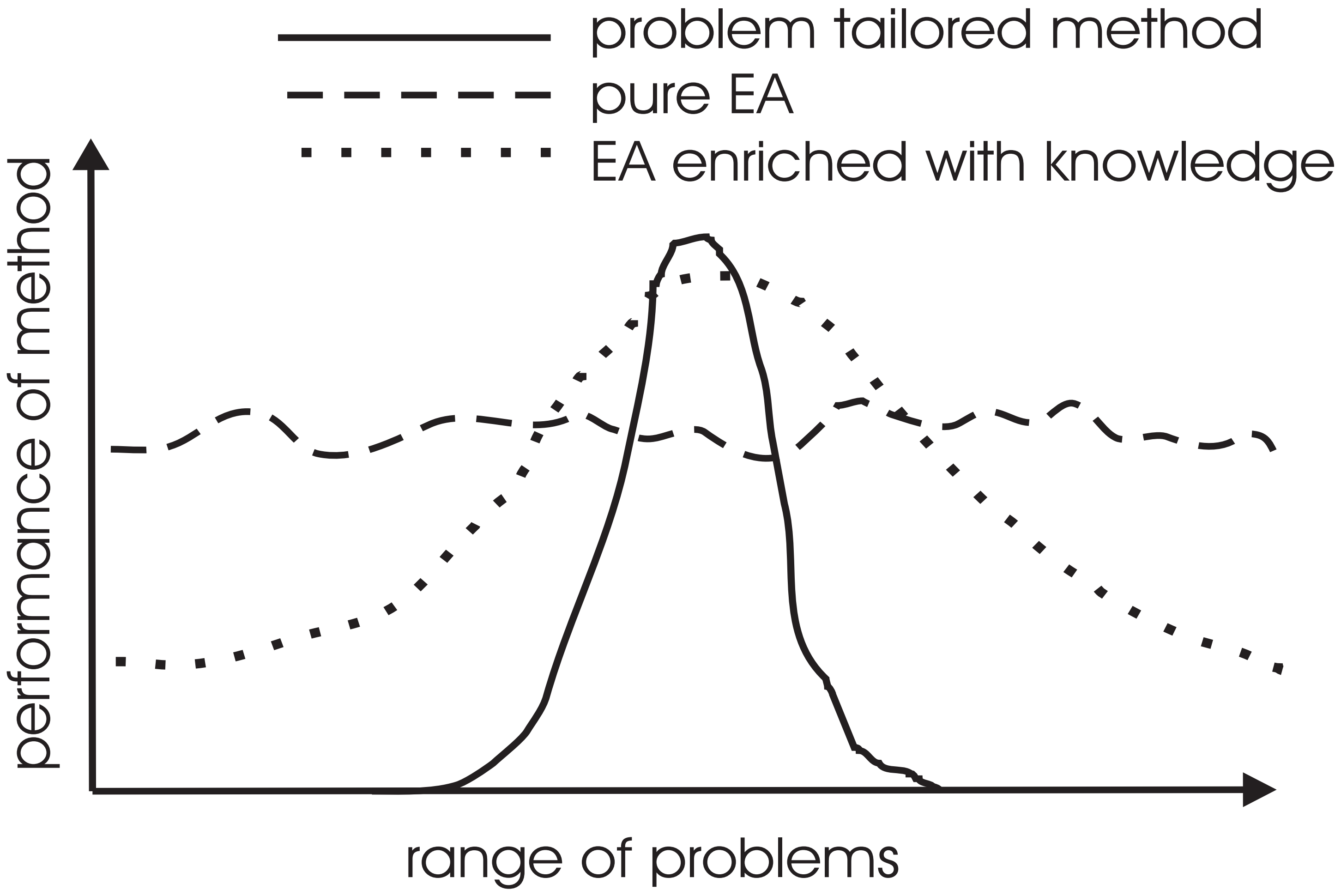
Overview

- Why to Hybridise
- Where to hybridise
- Incorporating good solutions
- Local Search and graphs
- Lamarkian vs. Baldwinian adaptation
- Diversity
- Operator choice

Why Hybridise

- Might want to put in EA as part of larger system
- Might be looking to improve on existing techniques but not re-invent wheel
- Might be looking to improve EA search for good solutions

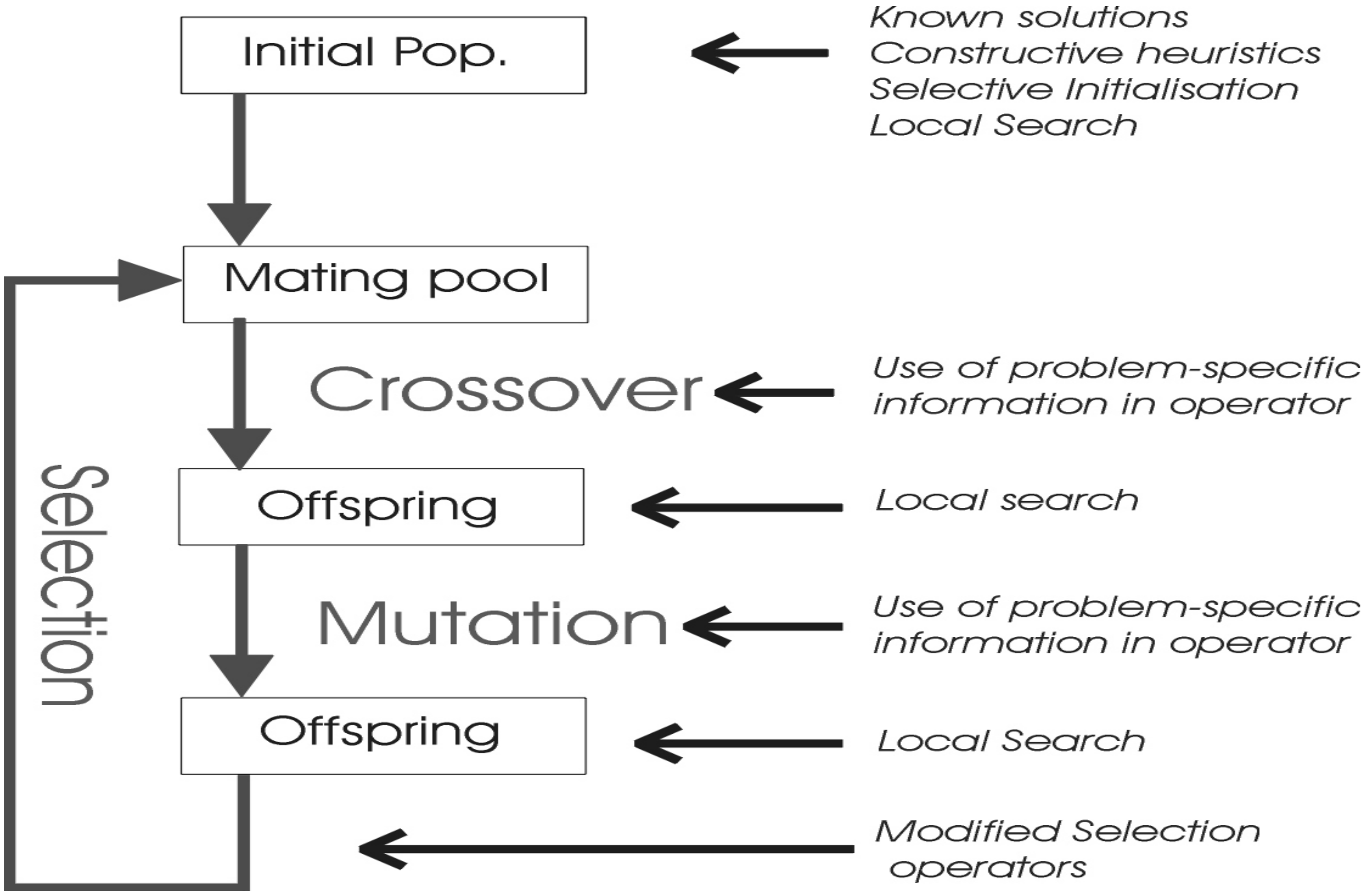
Michalewicz's view on EAs in context



Memetic Algorithms

- The combination of Evolutionary Algorithms with Local Search Operators that work within the EA loop has been termed "Memetic Algorithms"
- Term also applies to EAs that use instance specific knowledge in operators
- Memetic Algorithms have been shown to be orders of magnitude faster and more accurate than EAs on some problems, and are the "state of the art" on many problems

Where to Hybridise



Heuristics for Initialising Population

- Bramlette ran experiments with limited time scale and suggested holding a *n*-way tournament amongst randomly created solutions to pick initial population

  (n.b. NOT the same as taking the best *popsize* of *n.popsize* random points)

- Multi-Start Local Search is another option: pick *popsize* points at random to climb from
- Constructive Heuristics often exist

Initialisation Issues

- Another common approach would be to initialise population with solutions already known, or found by another technique
- Surry & Radcliffe (1994) studied ways of “inoculating” population with solutions gained from previous runs or other algorithms/heuristics
  - found ***mean*** performance increased as population was biased towards know solutions,
  - but ***best*** performance came from more random solutions

“Intelligent” Operators

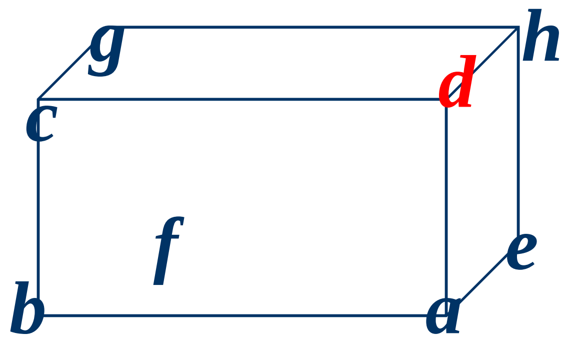
- It is sometimes possible to incorporate problem or instance specific knowledge within crossover or mutation operators
  - E.g. Merz’s DPX operator for TSP inherits common sub tours from parents then connects them using a nearesr neighbour heuristic
  - Smith (97) evolving microprocessor instruction sequences: group instructions (alleles) into classes so mutation is more likely to switch gene to value having a similar effect
  - Many other examples in literature

Local Search acting on offspring

- Can be viewed as a sort of “lifetime learning”
- Lots of early research done using EAs to evolve the structure of Artificial Neural Networks and then Back-propagation to learn connection weights
- Often used to speed-up the “endgame” of an EA by making the search in the vicinity of good solutions more systematic than mutation alone

Local Search

- Defined by combination of ***neighbourhood*** and ***pivot rule***
- Related to landscape metaphor
- *N(x)* is defined as the set of points that can be reached from *x* with one application of a move operator



$N(d) = \{a,c,h\}$

Landscapes & Graphs

- The combination of representation and operator defines a graph *G(v,E)* on the search space
- *v*, the set of vertices, is the set of all points that can be represented (the potential solutions)
- *E*, the set of edges, is the possible transitions that can arise from a single application of the operator
    - note that the edges in *E* can have weights attached to them, and that they need not be symmetrical

Example Graphs

- example : problem as above
  - *v = {a,b,c,d,e,f,g,h,}*
  - Search by flipping each bit in turn
    - E_1 = *{ ab, ad, ae, bc, bf, cd, cg, dh, fg, fe, gh, eh}*
    - symmetrical and all values equally likely
  - p^2*{ac,bd,af,be,dg, ch, fh, ge, ah, de, bg, cf}* ∪ p^3 *{ag, bh, ce, df}*

Graphs

- The ***Degree*** of a graph is the maximum number of edges coming into/out of a single point, - the size of the biggest neighbourhood
- Local Search algorithms look at points in the neighbourhood of a solution, so complexity is related to degree of graph

Pivot Rules

- Is the neighbourhood searched randomly, systematically or exhaustively ?
- does the search stop as soon as a fitter neighbour is found (***Greedy Ascent***)
- or is the whole set of neighbours examined and the best chosen (***Steepest Ascent***)
- of course there is no one best answer, but some are quicker than others to run ....

Variations of Local Search

- Does the search happen in representation space or Solution Space ?
- How many iterations of the local search are done ?
- Is local search applied to the whole population?
  - or just the best ?
  - or just the worst ?
  - see work (PhD theses) by Hart (www.cs.sandia.gov/~wehart), and Land

Information Use in Local Search

- Most Memetic Algorithms use an operator acting on a single point, and only use that information
- However this is an arbitrary restriction
  - Jones (1995), Merz & Friesleben (1996) suggest the use of a crossover hillclimber which uses information from two points in the search space
  - Krasnogor & Smith (2000) - see later - use information from whole of current population to govern acceptance of inferior moves
  - Could use Tabu search with a common list

Diversity

- Maintenance of diversity within the population can be a problem, and some successful algorithms explicitly use mechanisms to preserve diversity:
  - Merz’s DPX crossover explicitly generates individuals at same distance to each parent as they are apart
  - Krasnogor’s Adaptive Boltzmann Operator uses a Simulated-Annealing like acceptance criteria where “temperature” is inversely proportional to population diversity

Hybrid Algorithms Summary

- It is common practice to hybridise EA’s when using them in a real world context.
- this may involve the use of operators from other algorithms which have already been used on the problem (e.g. 2-opt for TSP), or the incorporation of domain-specific knowledge (e.g PSP operators)
- Memetic algorithms have been shown to be orders of magnitude faster and more accurate than GAs on some problems, and are the “state of the art” on many problems