Faculty of Information Technology

Advanced topics in computational science
FIT4012

Aldeida Aleti and Julian Garcia
aldeida.aleti@monash.edu, julian.garcia@monash.edu

Part 1 schedule

- Combinatorial Problems and Computational Complexity
- Systematic, Local and Stochastic Search
- Fitness Landscape Analysis
- Parameter Control for Evolutionary Algorithms
- Constrained Problems and Constraint-Handling Techniques

Optimisation

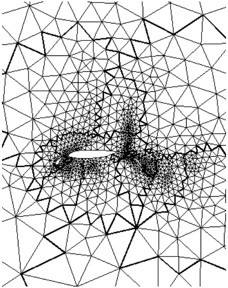
Search space S - Set of all feasible solutions

Objective function $f : S \rightarrow \Re$ - quality criterion

Goal $x^* = argmax(min)f$ - finding the **best** solution according to the criterion

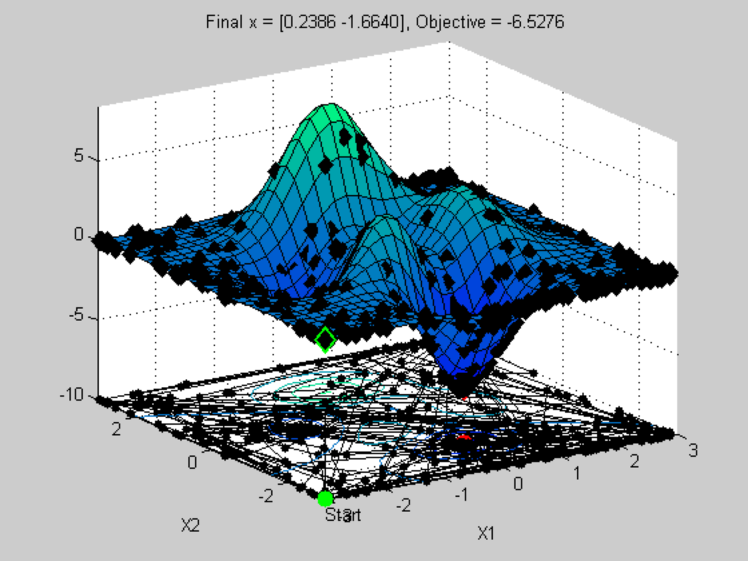
Often, optimisation problems are

- NP-hard (large search space)
- combinatorial
- with fitness function(s) which:
  - cannot be formulated as a closed-form expression,
  - irregular,
  - non differentiable,
  - non continuous.

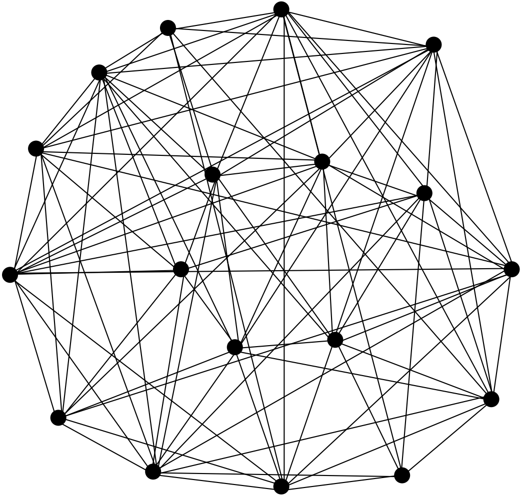


Not solvable by traditional methods!

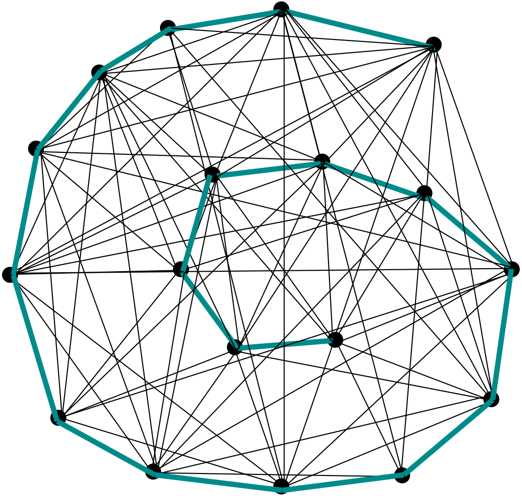
Search algorithms



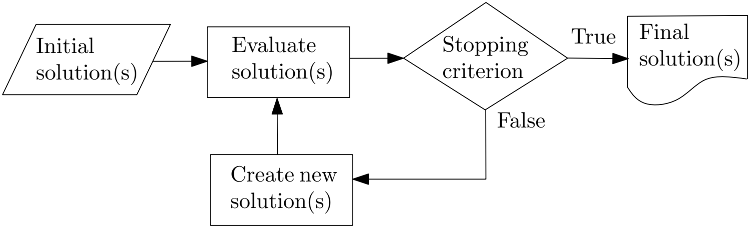
Example: Travelling Salesperson Problem



Travelling Salesperson Problem Optimised



Example: Evolutionary Algorithms



Software Testing and Code Coverage

```
if (testScore >= 70) {
    if (studentAge < 10) {
        System.out.println("You did a great job");
    } else {
        System.out.println("You did pass"); //test score >= 70
    }                                        //and age >= 10
} else { //test score < 70
    System.out.println("You did not pass");
}
```

A program with high code coverage has been more thoroughly tested and has a lower chance of containing software bugs than a program with low code coverage.

Example

```
public class Stack {
  int[] values = new int[3];
  int size = 0;
  void push(int x) {
    if(size >= values.length)        ⇐ Requires a full stack
      resize();
    if(size < values.length)         ⇐ Else branch is infeasible
      values[size++] = x;
  }
  int pop() {
    if(size > 0)⇐ May imply coverage in push and resize
      return values[size--];
    else
      throw new EmptyStackException();
  }
  private void resize(){
    int[] tmp = new int[values.length * 2];
    for(int i = 0; i < values.length; i++)
      tmp[i] = values[i];
    values = tmp;
  }
}
```

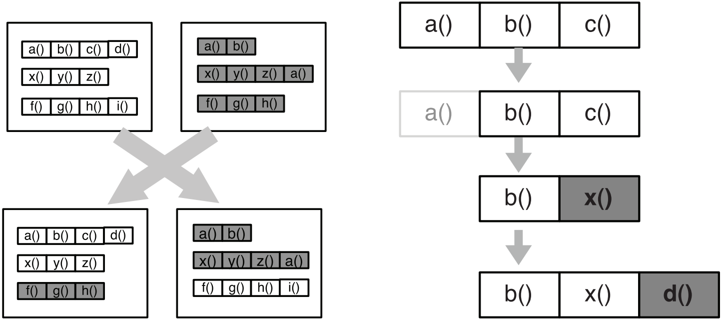
Test Data Generation

An example of a test suite

```
Stack stack0 = new Stack();
try {
  stack0.pop();
} catch(EmptyStackException e) {
}
```

```
Stack stack0 = new Stack();
int int0 = −510;
stack0.push(int0);
stack0.push(int0);
stack0.push(int0);
stack0.push(int0);
stack0.pop();
```

Genetic Algorithms for Test Data Generation



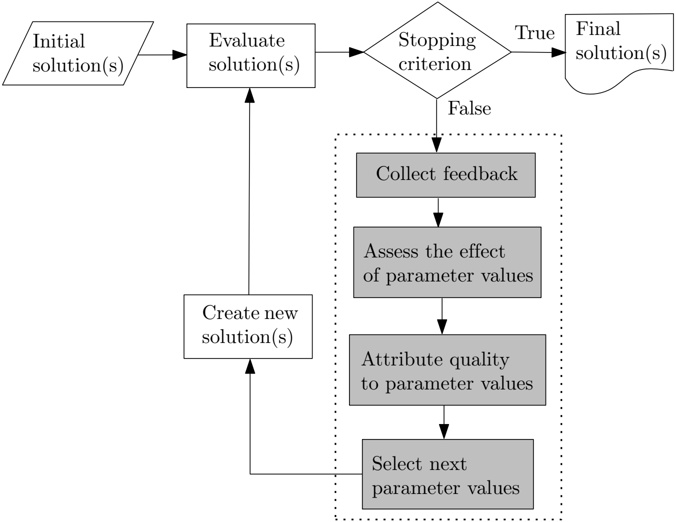
(a) Test Suite Crossover (b) Test Case Mutation

(Fraser and Arcuri, 2013)

Tuning Parameter Values

- Search algorithms (e.g. GA) have many parameters that are problem dependent.
- Different parameter values may be optimal at different stages of the optimisation process.

Adaptive Genetic Algorithm for Test Data Generation



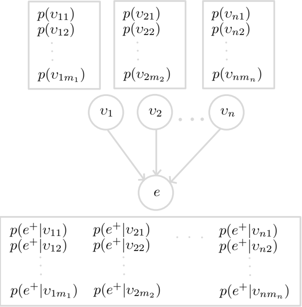
Probabilistic Parameter Control

- Estimates the effect of parameter values on the performance of the algorithm
- Problem: uncertainties about the effect of the parameter values
    - Several parameter values (υ_{ij})
    - Noise

Probabilistic effect assessment

$$e = \begin{cases} e^{+} & \text{if} \quad f(x') - f(x) > th \\ e^{-} & \textit{otherwise} \end{cases}$$

$$P(e^{+}|\upsilon_{ij}) = \frac{P(\upsilon_{ij} \wedge e^{+})}{P(\upsilon_{ij})}$$



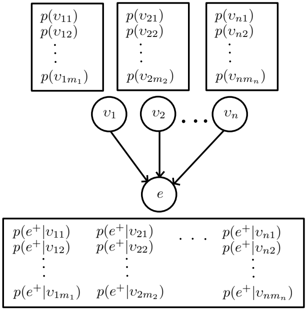
Probabilistic Parameter Control

- Estimates the effect of parameter values on the performance of the algorithm
- Problem: uncertainties about the effect of the parameter values
  - Several parameter values (υ_{ij})
  - Noise

Probabilistic effect assessment

$$e = \begin{cases} e^{+} & \text{if} \quad f(x') - f(x) > th \\ e^{-} & otherwise \end{cases}$$

$$P(e^{+}|\upsilon_{ij}) = \frac{P(\upsilon_{ij} \wedge e^{+})}{P(\upsilon_{ij})}$$

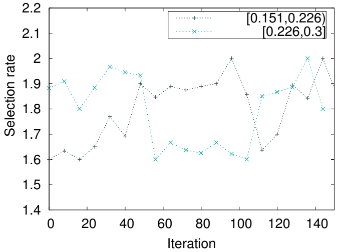


Experiments

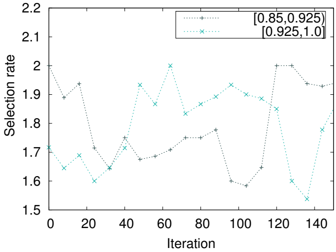
Name	# Classes	# Branches	Name	# Classes	# Branches
ifx-framework	2189	93307	mygrid	35	1266
jcvi-javacommon	565	7347	jigen	35	631
caloriecount	524	12064	shop	32	1035
openjms	486	11744	dsachat	31	951
summa	428	13711	jaw-br	29	811
lilith	311	17063	gangup	29	991
corina	310	10731	inspirento	26	571
heal	186	6070	rif	25	488
at-robots2-j	174	2201	ext4j	23	525
lhamacaw	168	4973	fixsuite	22	519
xbus	168	4422	xisemele	21	343
jiggler	140	6325	biblestudy	21	630
dom4j	136	5702	imsmart	21	183
jnfe	128	2428	jgaap	19	222
hft-bomberman	125	1956	templateit	19	692

Adaptation of crossover and mutation rates

Mutation rate

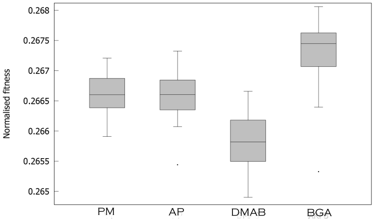


Crossover rate

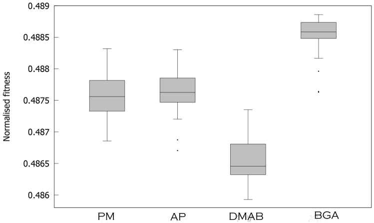


Results

135 classes

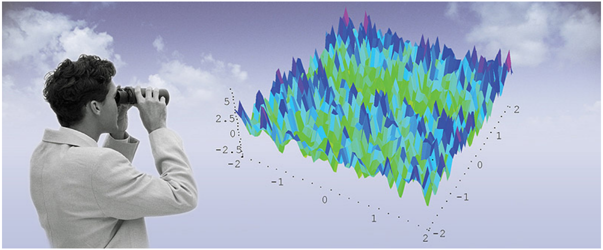


246 classes



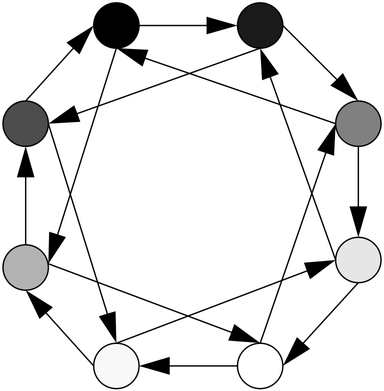
So what?

- Algorithm parameters are the key factor determining the performance of the optimisation method,
- Feedback from the search can help select the right search strategy



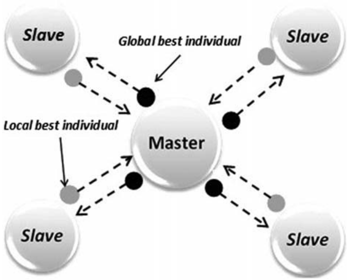
Parallel Genetic Algorithms

Island model



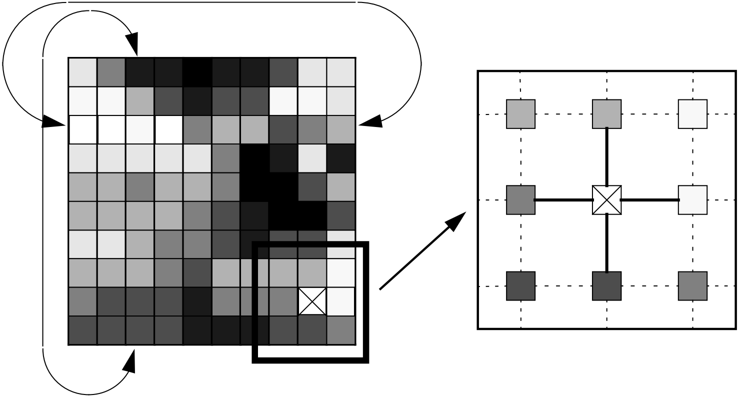
Parallel Genetic Algorithms (continued)

Master-slave model



Parallel Genetic Algorithms (continued)

Cellular model



Assignments

to be continued ...