



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU22012: Data Structures and Algorithms II

Lecture 1: Intro and Logistics

Dr Anthony Ventresque

Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's [policies and procedures](#).

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

© Trinity College Dublin 2020



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Who am I?

- Anthony Ventresque
 - ORIG10 (SCSS Building)
- BSc and MSc in Philosophy, BSc, MSc and PhD in Computer Science (France), Research positions in NTU (Singapore), UCD and IBM Ireland
- Research Areas:
 - Software Quality (Testing)
 - Accessibility (CV/ML)
 - Large Software Systems (Big Data, the Cloud)



Mutation Analysis

Code:

```
if ((a <= 0) || (b <= 0) || (c <= 0)) {  
    return TriangleType.INVALID;  
}  
if ((a == b) && (b == c)) {  
    return TriangleType.EQUILATERAL;  
}
```

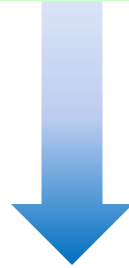
Tests:

```
@Test  
public void testInvalidNeg() {  
    assertEquals(INVALID, Triangle.classify(-1, 1, 1));  
    assertEquals(INVALID, Triangle.classify(1, -1, 1));  
    assertEquals(INVALID, Triangle.classify(1, 1, -1));  
}
```

Mutation Analysis

Code:

```
if ((a <= 0) || (b <= 0) || (c <= 0)) {  
    return TriangleType.INVALID; //This line has been executed  
}
```

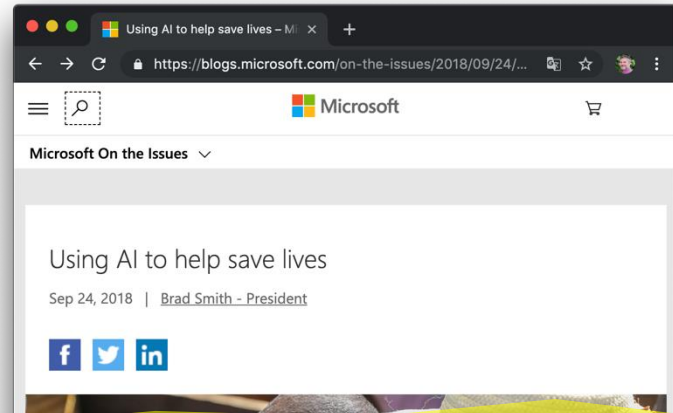


Mutant:

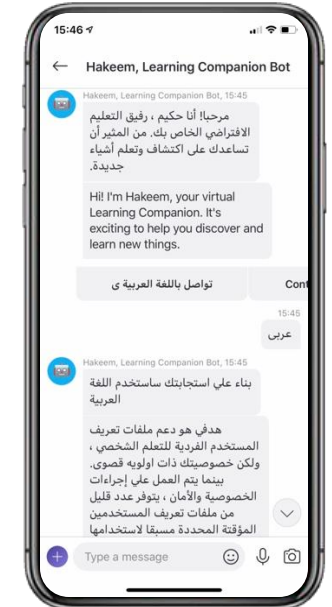
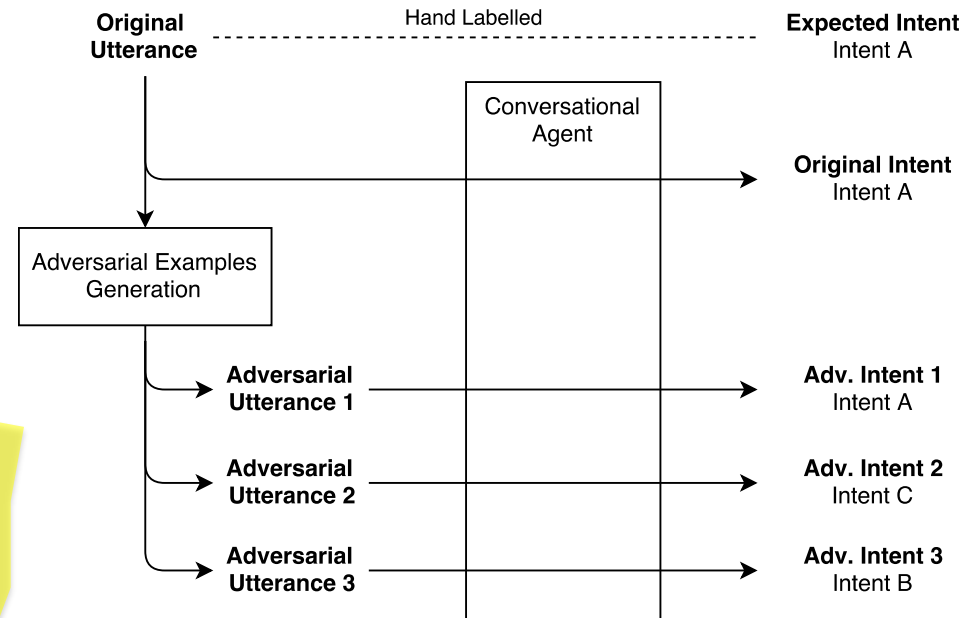
```
if ((a < 0) || (b <= 0) || (c <= 0)) {  
    return TriangleType.INVALID;  
}
```

There is no test for $a == 0$
=> the mutant is ***not detected***

Testing Conversational AI (Chatbots)



[...] Microsoft is working already with the Norwegian Refugee Council, NetHope and University College Dublin to develop a chatbot using AI technologies, such as language understanding, machine translation and speech recognition [...]



Sign Language Recognition

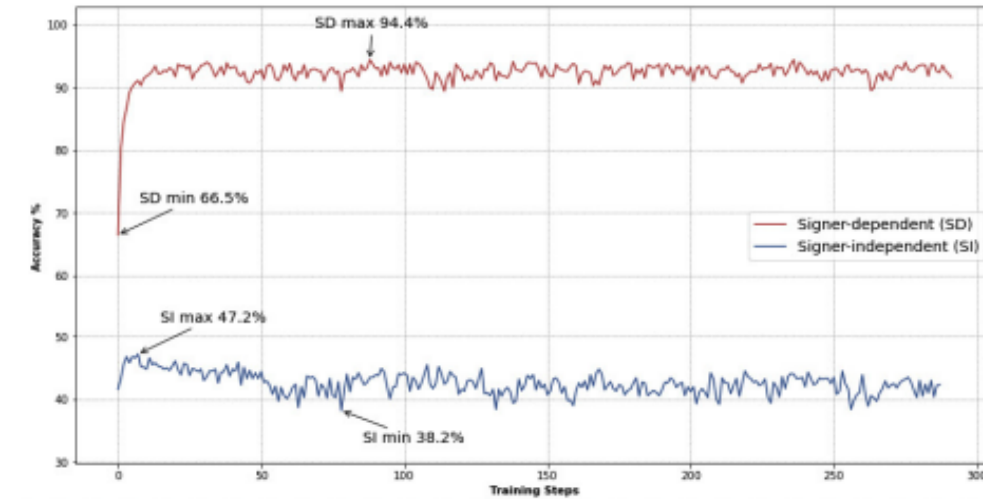
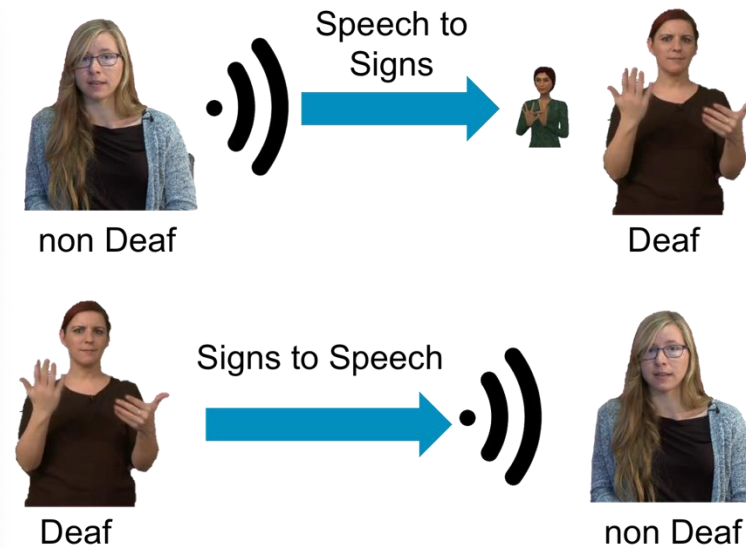
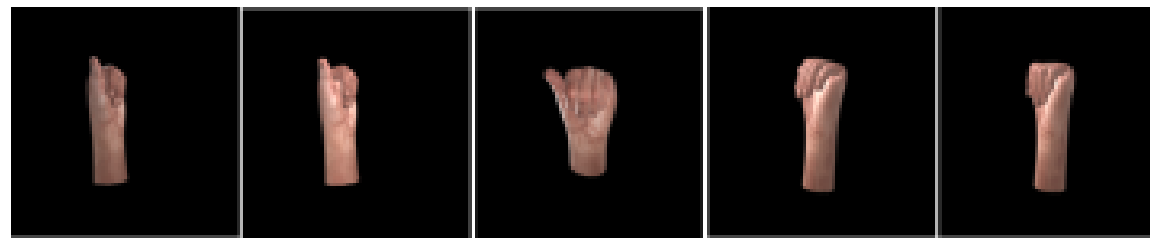
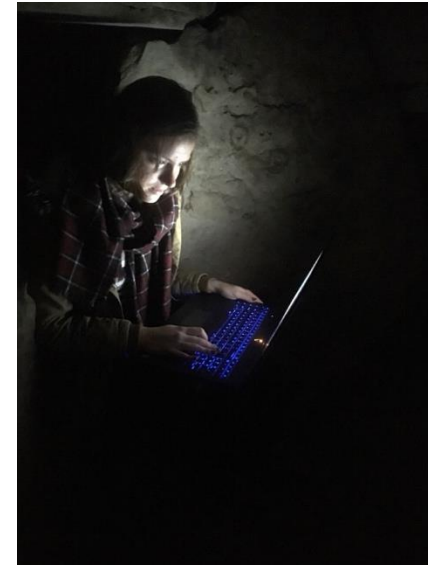
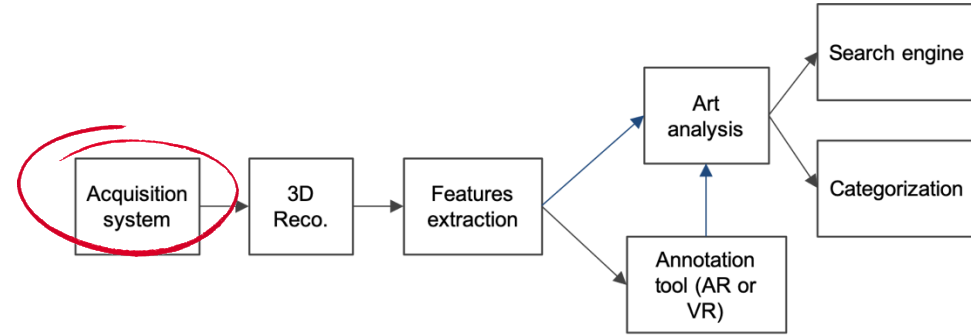


Figure 2: Validation accuracy for the signer-dependent and signer-independent models trained on greyscale images with only layers added to the pre-trained VGG network fine-tuned. Minimum and maximum values are labeled.



Computational Archaeology



Outline

- Structure of the module
- Examples of what DS&A can do for you
- Course Content

Timetable

Lectures

- Wednesdays 9-11 JOLY4

Labs

- Mondays 10-11
- Tuesdays 10-11
- Tuesdays 2-3
- Attend only your assigned time slot
- L2.01 - SCSS Lab 1, Biosciences (spillover in L2.02, but most likely we'll only need L2.01 so go there first!)

Lectures – Hilary Term

- Reading week – March 3rd – 10th no lectures
- Bank holidays – use other lab slots
- Lectures streamed (could change in the future)
 - In class test and exam mostly on lecture content

Labs – Hilary Term

- Labs are self contained
 - Introduce new tech
 - Practice of some of the lectures' concepts
 - help you with assignments
- Assignment issues/questions
 - POST ON BLACKBOARD – no assignment questions will be answered over email – everyone usually has same questions/issues, so let's make the questions/answers of benefit to everyone
 - Before posting, check if question came up before
- I am also available to discuss
 - Office hour Thursday 2-3pm (could change in future); email me first Anthony.ventresque@tcd.ie

Assessment

- 1 quiz each week: ***not graded*** (5-10 minutes)
- 1 or 2 (TBC) programming assessment: **40%**
- In-class test (week ??, 1h): **20%**
- Exam (2h): **40%**

What are we going to learn?

What's going on?

```
import sys
```

```
def show_list_growth(n):  
    my_list = []  
    for i in range(n):  
        a = len(my_list)  
        b = sys.getsizeof(my_list)  
        print("Length: ", a, " Size in bytes: ", b)  
        my_list.append(i)
```

```
show_list_growth(50)
```


What's going on?

Length: 0	Size in bytes: 56
Length: 1	Size in bytes: 88
Length: 2	Size in bytes: 88
Length: 3	Size in bytes: 88
Length: 4	Size in bytes: 88
Length: 5	Size in bytes: 120
Length: 6	Size in bytes: 120
Length: 7	Size in bytes: 120
Length: 8	Size in bytes: 120
Length: 9	Size in bytes: 184

mprof (Python)

memory_profiler has a built-in ability to plot the output
Philippe Gervais (INRIA, Google) implemented a nice feature

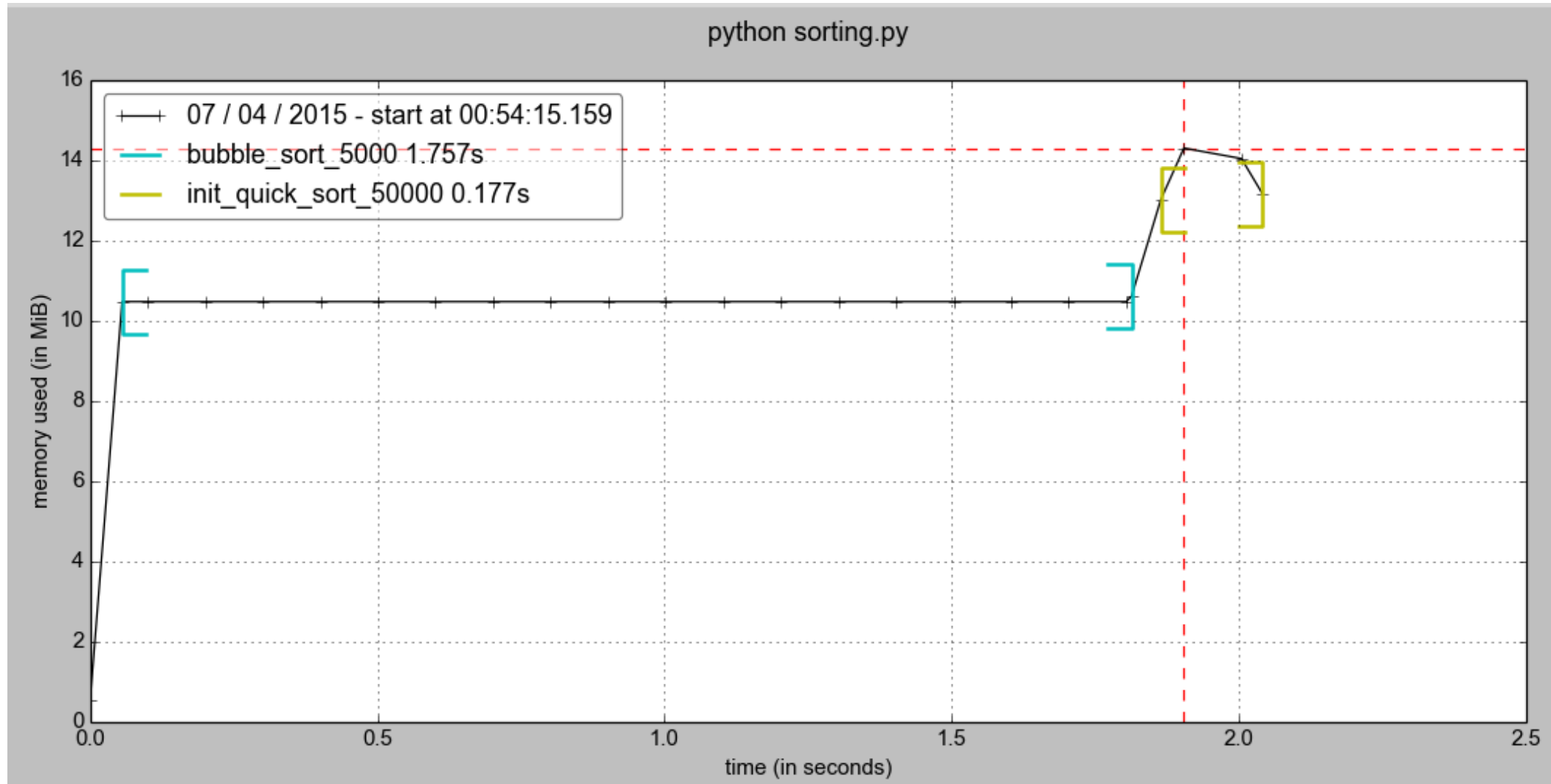
Run:

```
anthony@hibernia$ mprof run sorting.py  
anthony@hibernia$ mprof plot
```

Example for mprof (sorting.py)

```
import random
import time
@profile
def bubble_sort_5000(items):
    [...]
@profile
def init_quick_sort_50000(items):
    [...]
def main():
    items = [random.randint(0, 10000) for c in range(5000)]
    bubble_sort_5000(items)
    items = [random.randint(0, 10000) for c in range(50000)]
    init_quick_sort_50000(items)
```

Output of mprof



Complex Problems

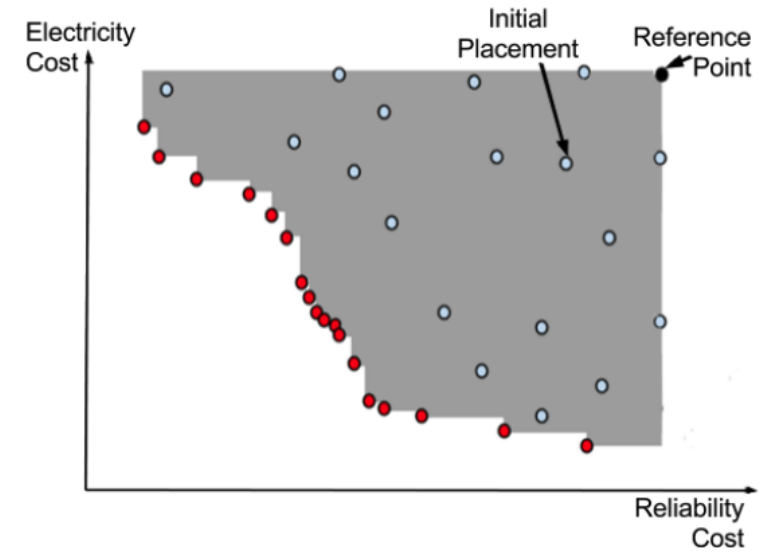
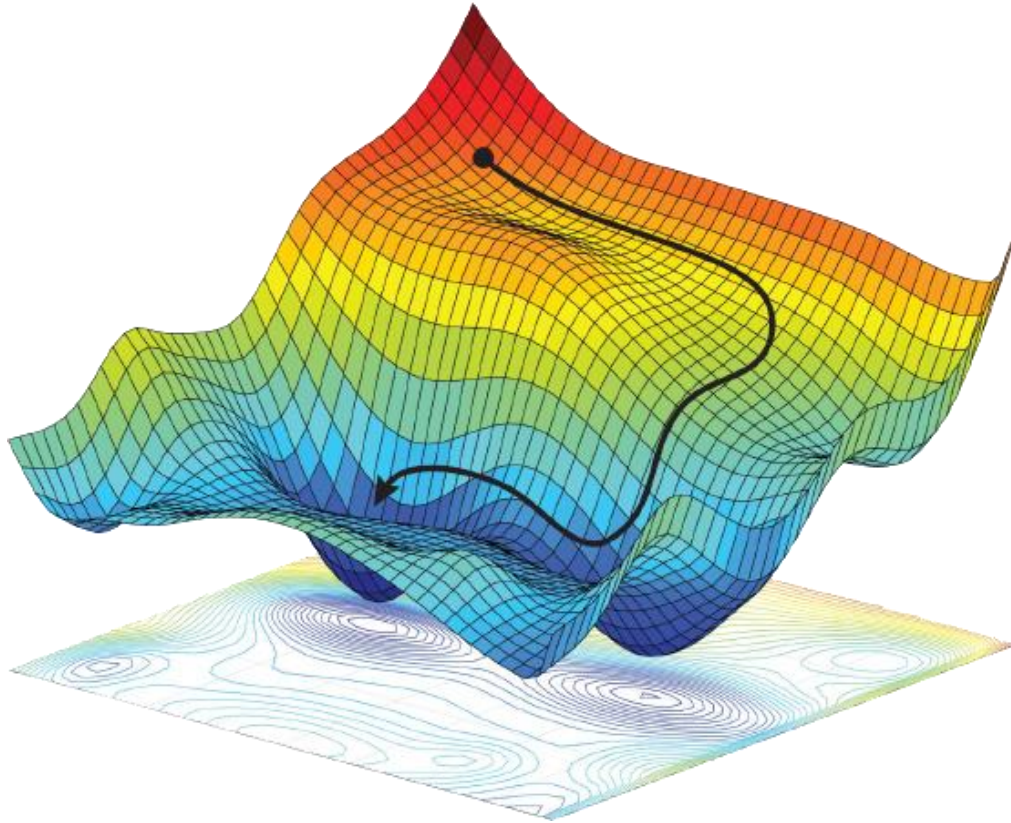


Figure 4: Metrics: number of Pareto solutions (red dots) and hypervolume (grey area).

Course content - Review and expand

- Sorting algorithms
 - Insertion sort, heapsort
 - Selection sort, shellsort, mergesort, quicksort
 - Space and time trade offs
 - Select and compare based on input type and size
- Algorithmic approaches
 - Brute force, exhaustive search, decrease and conquer, divide and conquer, greedy, dynamic programming ...

Course content – New Topics

- Graphs – shortest path
 - Dijkstra
 - Depth-first, breadth-first search, Prim, Kruskal, Topological sort
 - Shortest paths - Bellman-Ford, Floyd-Warshall
 - What to use based on graph – directed, undirected, acyclic, negative edge weights etc
- Network flow algorithms
 - Maxflow, Ford-Fulkerson
- Strings
 - String sorts
 - Substring search
- Optimisation
 - Local Search
 - Random Search
 - Genetic Algorithms

Summary

- Lectures streamed
- Labs self contained and/or support for assignments
- Weekly Quizzes
- In-class mid-term & final “exam”
- Any non-private questions – Blackboard forum only!