Myth 1: SBSE relies on experimentation. There's little theoretical & comparative work.

FACT: ① Vigorous time complexity analysis of search algorithms for testing has appeared in several papers.

② There are established methodology and benchmarking studies in the search community.

③ However, there are still huge gaps between theory & practice.

Myth 2: Search algorithms are complex/complicated

FACT: • Evolutionary algorithms are not significantly more complex than simulated annealing.

• It's true that it's inapproriate to use complex algorithms for simple problems.

• There has been significant progress made recently in characterising difficult problems: what makes a problem hard for a search algorithm?

EA-Hardness — linking problem characteristics to algorithmic features

⋮
↓

Guide problem classification

XIN YAO 15/05/2009

The fitness issue:

- hard to define precisely & accurately
- hard to quantify

No easy solutions, but ...

- Co-evolution
- Interactive evolutionary computation (IEC)
- Evolutionary computation
  as a discovery engine
  not just a problem solving tool

Supporting what Enrique said:

There are many search algorithms that may be of interest to SErs:

- optimisation in an uncertain environment
- Dynamic optimisation
  Robust optimisation

XIN YAO 15/05/2009

Interactive Evolution

Very little work on Interactive Evolution in SBSE.

why?

+ perhaps because of the difficulties

but this approach could be so useful

+ incorporate domain knowledge
+ capture real world assumptions
+ take account of "fuzzy, messy, ill defined human stuff"

—— o ——

in requirement problems:- Juan Jo Durillo's talk:-

- explore/expose unstated ? {dependencies, requirements, assumptions} etc

in architectural problems Dongsun Kim's talk:-

- capture qualatative assessments
- explore developer cog. models

... Maxim Shevertalov's talk ...

in authorship (not direct but related...)

- tailored pretty printing
- compare styles & performance
- ...

ICPC 07 Keynote paper

Mark Harman

Kpodjedo et al.

Q - 1 :-

① Why "linear" regression ?

Q - 2 :-

② Don't you think that C & K metrics are software dependent ?

Q - 3 :-

③ wouldn't it be useful to test for statistical significance for R^2 for RQ-1 ?

Wasif Afzal

13/05/2009

SBST: Search Based Software Testing

... a plea for Multi Objective SBST

Testing is not just about coverage

In fact it is seldom about a single objective!

multiple objectives:

1. coverage
2. time to execute
3. length of trace
4. oracle cost
5. exercise known bugs

... others?

Traditional techniques often have to be reformulated to handle more than one objective

SBSE can be extended to multiple objectives with comparative ease.

some possibilities...

... Covering arrays with fewest constraint violation
best ~~co~~cover ?
lowest oracle cost
...

Interaction ⇝

?

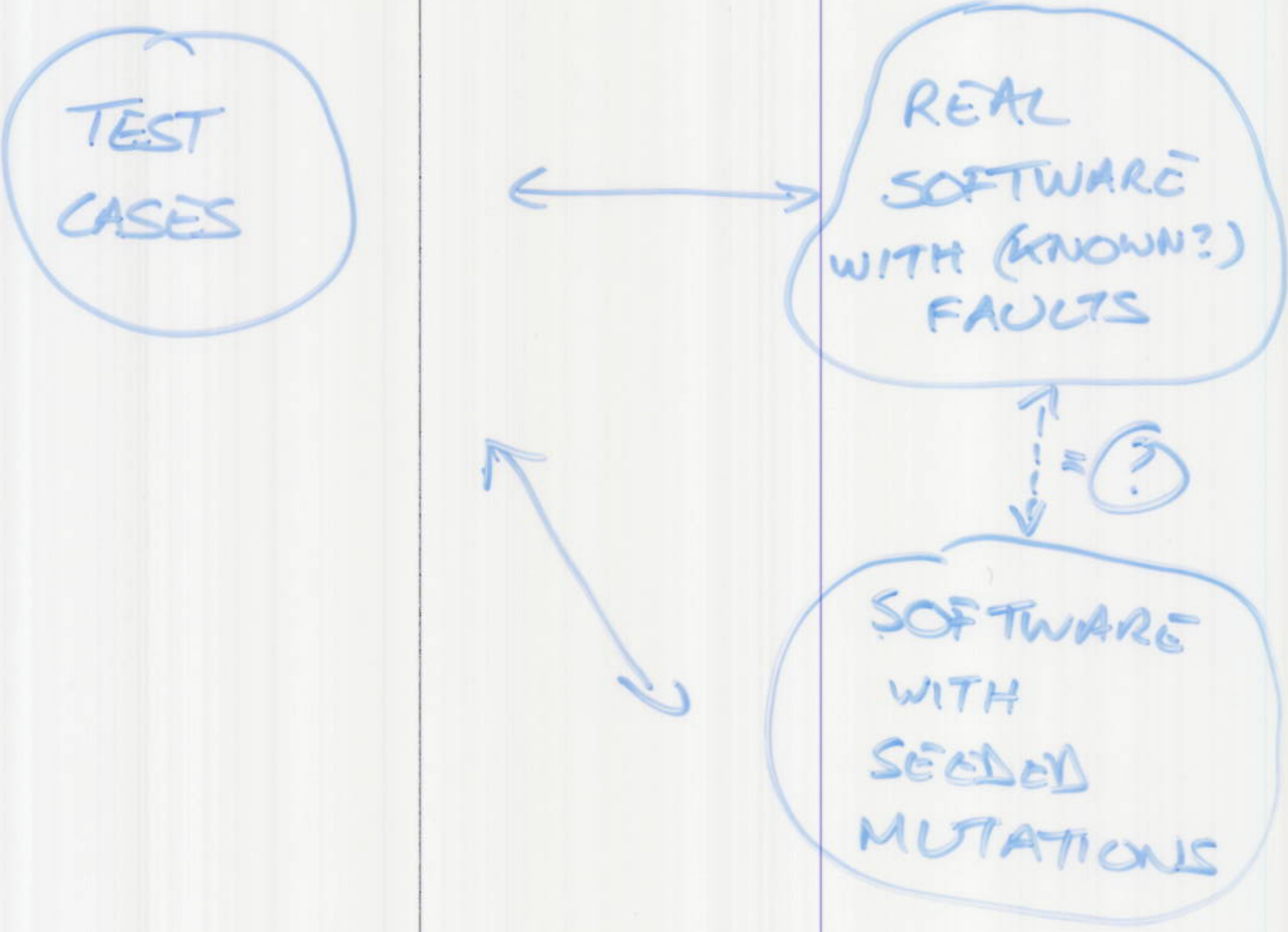
interaction sequences that

- maximize fault cover (previous faults)
- minimize oracle cost
- longest/shortest sequence
- include certain paths etc.

Ajax ⇝

Mark Harman

13/05/2009


TEST CASES
REAL SOFTWARE WITH (KNOWN?) FAULTS
= ?
SOFTWARE WITH SEEDED MUTATIONS

SIMON POULDING

13/05/2009

example of the 'psp' plastic slide principal...

Q Why not use static analysis?

A Sure. SBSE is complementary not conflicting

Q When can I use SBSE?

A when the space is too large for static analysis

Q when should I use static analysis?

A: If a precise* solution can be found in reasonable time

*or goodenough

Model checking to prove absence of deadlock is "exhaustive search on an optimized search space that is sufficiently small"

- when it's toolarge we can't use exhaustive search.

... but the space is still interesting & important

13/05/2009

Mark Harman