SBSE

Search-Based Software Engineering

ZHENG LI (李征)
lizheng@mail.buct.edu.cn

北京化工大学

Thanks Mark Harman for the partial of slides

软件重要性

Our civilization runs on software

软件是现代文明的重要基础

C++的设计者 **Bjarne Stroustrup**

软件的内涵

☺ 软件 = 程序 + 文档(百科全书)

⇒软件 = 算法 + 数据结构 + 文档

软件的内涵

☺ 软件 = 程序 + 文档(百科全书)

⇒软件 = 算法+数据结构 + 文档

☺从认知的角度来理解软件

软件 = 知识 + 使用

软件的内涵

☺ 软件 = 程序 + 文档(百科全书)

⇒软件 = 算法+数据结构 + 文档

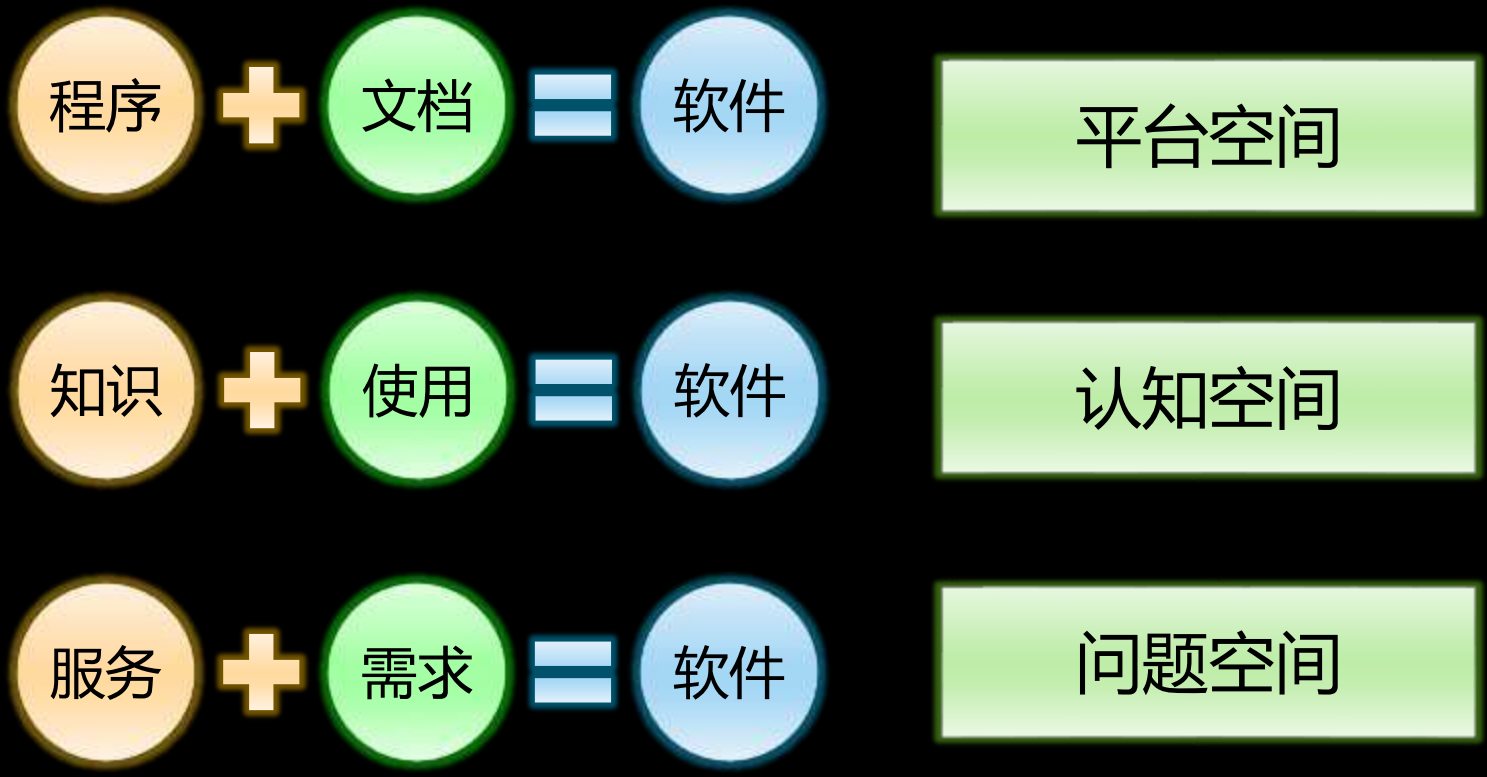
☺从认知的角度来理解软件

软件 = 知识 + 使用

☺从使用的角度来理解软件

软件 = 服务 + 需求

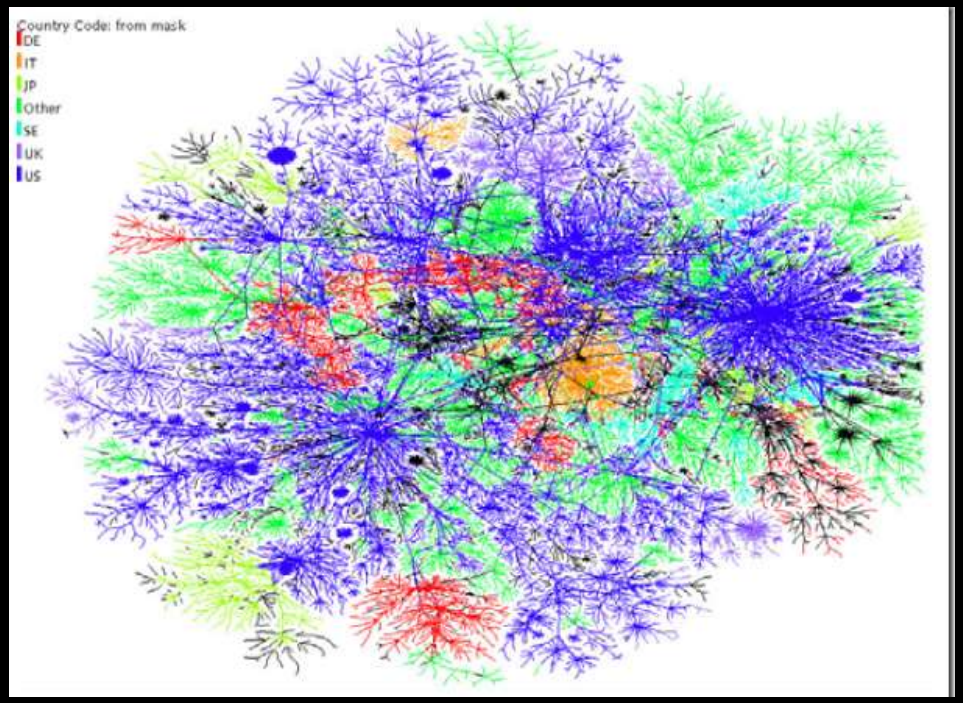
软件的内涵（吕建）



平台空间：硬件平台+系统软件

- ☺从1946年诞生的第一台电子计算机
- ☺到现在是GUC（**Global Ubiquitous Computer**）

- ☺网络就是计算机



认知空间

知识编程的探索:

☺ **Prolog**程序 = 固化推理 + 知识编程

⇒ 知识编程机制 + 推理机制

☺ 信息**Web** = 海量的网页 + 搜索引擎

⇒ 开放容歧结构 + 搜索机制

☺ 语义**Web** = 带结构的网页 + **Agent**

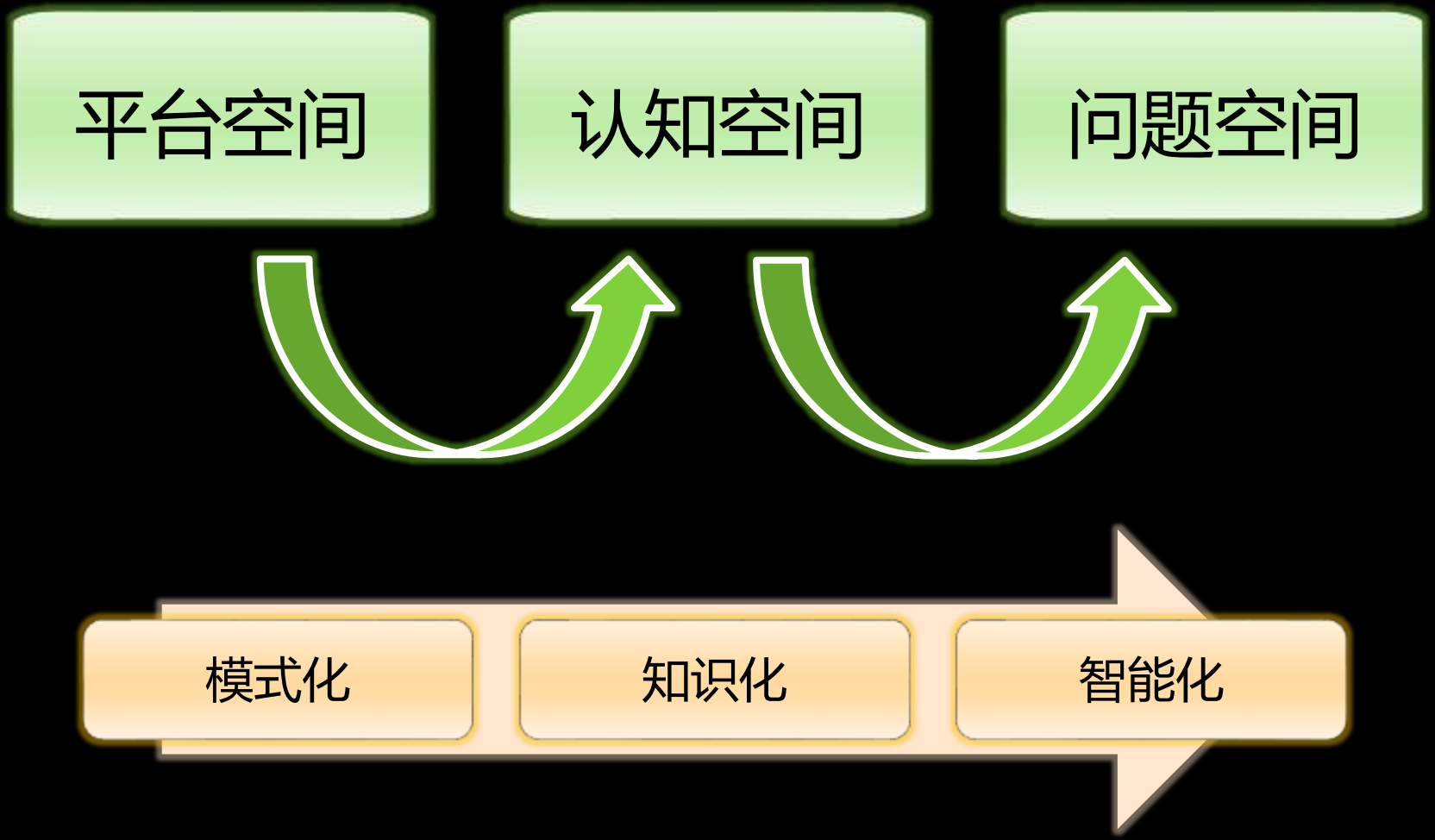
⇒ 可用知识表示 + 协同合作

问题空间

实现需求与服务的匹配

基于人工智能的问题求解

趋势与挑战：软件工程



问题求解框架下AI与SE融合

- 问题空间
  - 在问题空间中构造所有可能的解决问题的方法

状态空间的一个解是一个有限的操作算子序列，它使初始状态转化为目标状态：

问题求解框架下AI与SE融合

- 问题空间
- 解空间
  - 在解空间搜索最优解或近似最优解

What is SBSE

In SBSE we apply search techniques to search large search spaces, guided by a fitness function that captures properties of the acceptable software artefacts we seek.

like google search?
like code search?
like breadth first search?

What is SBSE

In SBSE we apply search techniques to search large search spaces, guided by a fitness function that captures properties of the acceptable software artefacts we seek.

Tabu Search

Ant Colonies

Particle Swarm Optimization

Genetic Algorithms

Hill Climbing

Genetic Programming

Simulated Annealing

Greedy

LP

Random

Estimation of Distribution Algorithms

What is SBSE

In SBSE we apply search techniques to search large search spaces, guided by a fitness function that captures properties of the acceptable software artefacts we seek.

Tabu Search
Ant Colonies
Particle Swarm Optimization
Genetic Algorithms
Hill Climbing
Genetic Programming
Simulated Annealing
Random
Greedy
LP
Estimation of Distribution Algorithms

Wes Weimer, ThanVu Nguyen, Claire Le Goues, Stephanie Forrest.
Automatically Finding Patches Using Genetic Programming. ICSE 2009 best paper.

What is SBSE

let's listen to software engineers ...

... what sort of things do they say?

Software Engineers Say

We need to satisfy business and technical concerns

We need to reduce risk while maintaining completion time

We need increased cohesion and decreased coupling

We need fewer tests that find more nasty bugs

We need to optimise for all metrics M1,…, Mn

Software Engineers Say

We need to satisfy business and technical concerns

We need to reduce risk while maintaining completion time

We need increased cohesion and decreased coupling

We need fewer tests that find more nasty bugs

We need to optimise for all metrics M1,..., Mn

Software Engineers Say

We need to satisfy business and technical concerns

We need to reduce risk while maintaining completion time

We need increased cohesion and decreased coupling

We need fewer tests that find more nasty bugs

We need to optimise for all metrics M1,..., Mn

Software Engineers Say

We need to satisfy business and technical concerns

We need to reduce risk while maintaining completion time

We need increased cohesion and decreased coupling

We need fewer tests that find more nasty bugs

We need to optimise for all metrics M1,..., Mn

Software Engineers Say

We need to satisfy business and technical concerns

We need to reduce risk while maintaining completion time

We need increased cohesion and decreased coupling

We need fewer tests that find more nasty bugs

We need to optimise for all metrics M1,..., Mn

Software Engineers Say

We need to satisfy business and technical concerns

We need to reduce risk while maintaining completion time

We need increased cohesion and decreased coupling

We need fewer tests that find more nasty bugs

We need to optimise for all metrics M1,…, Mn

Software Engineers Say

Requirements: We need to satisfy business and technical concerns

Management: We need to reduce risk while maintaining completion time

Design: We need increased cohesion and decreased coupling

Testing: We need fewer tests that find more nasty bugs

Refactoring: We need to optimise for all metrics M1,..., Mn

All have been addressed in the SBSE literature

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

optimize

fit for purpose

within constraints

Engineering words

with acceptable bounds

tolerance

improve performance

optimise

reduce cost

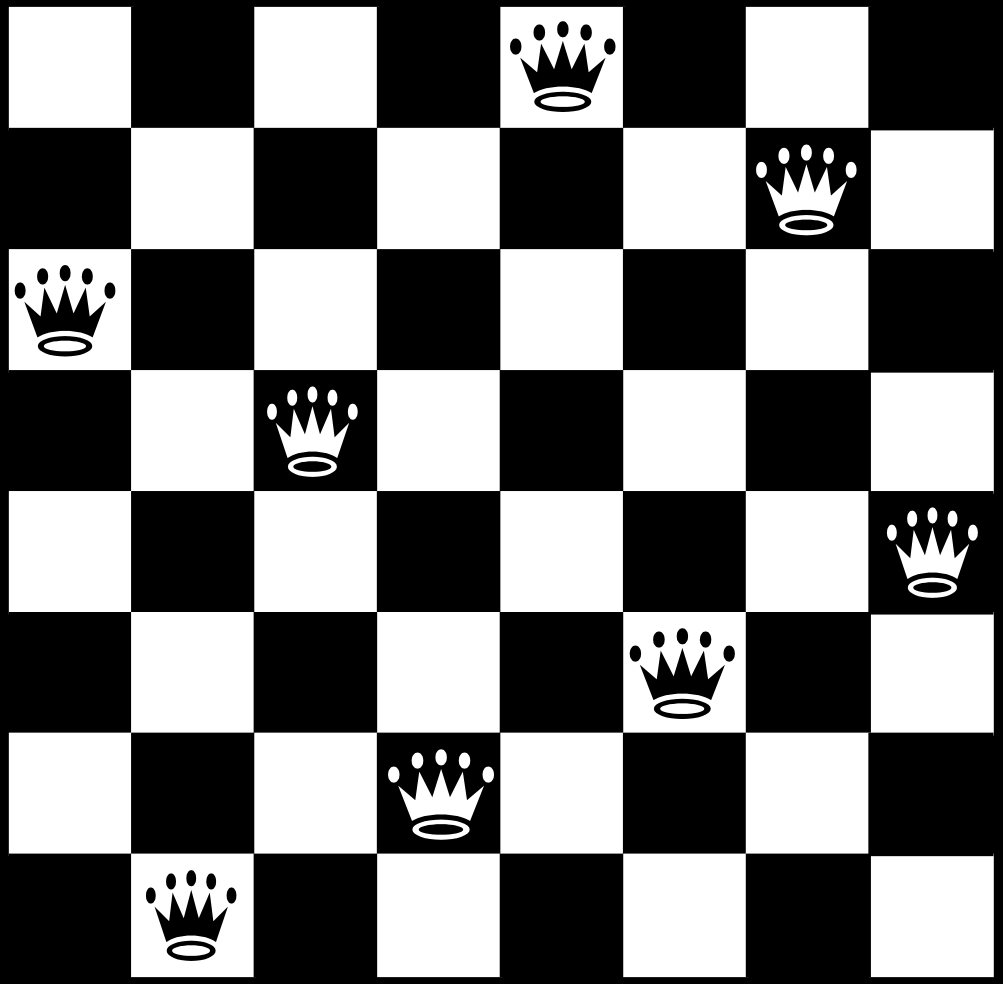
optimize

fit for purpose

within constraints

Why?

Eight Queens Problem



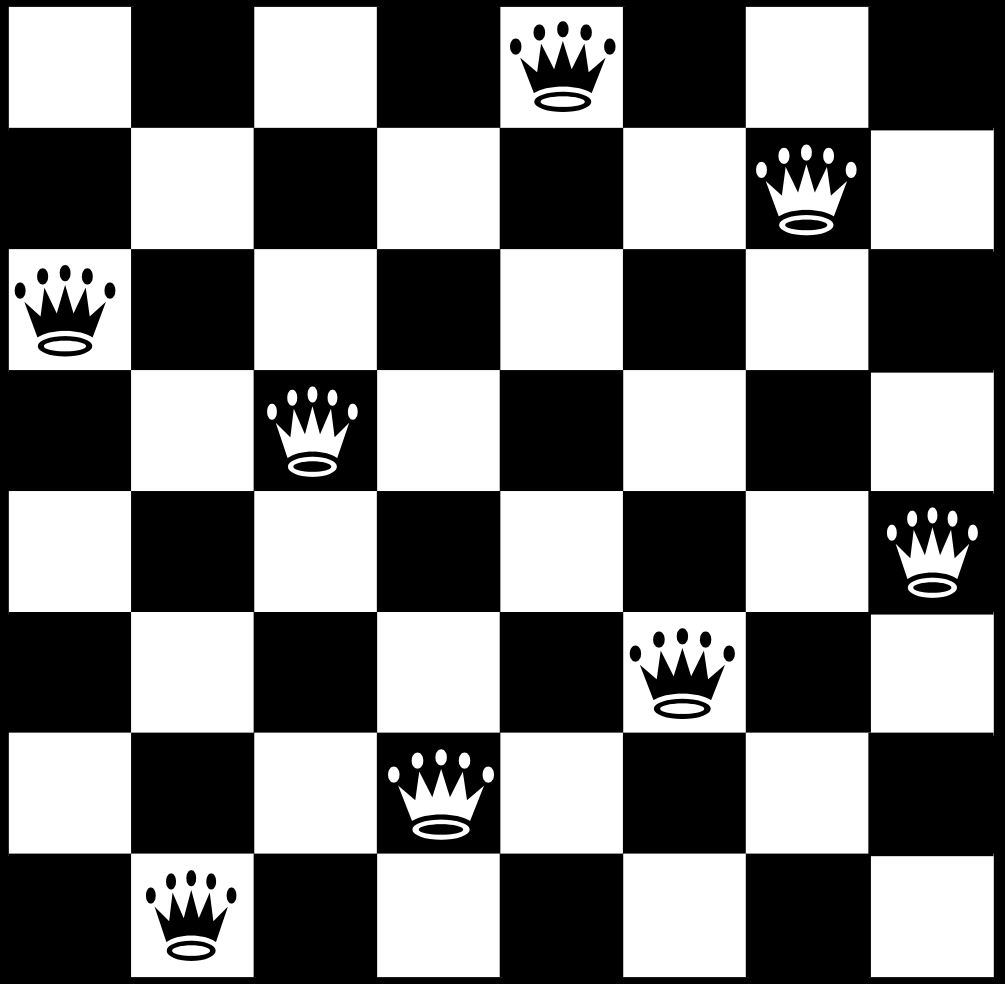
Traditional AI Search

- 依次摆放皇后
- 摆放时，定义f(x)=剩下未放行中能够用来放皇后的空格数，可以看出f(x)愈大愈好
- 搜素策略

第i个皇后放到第i行中与前面i-1个皇后不在同一列或对角线上，且f(x)值最大的空格中。

Search from the solution space

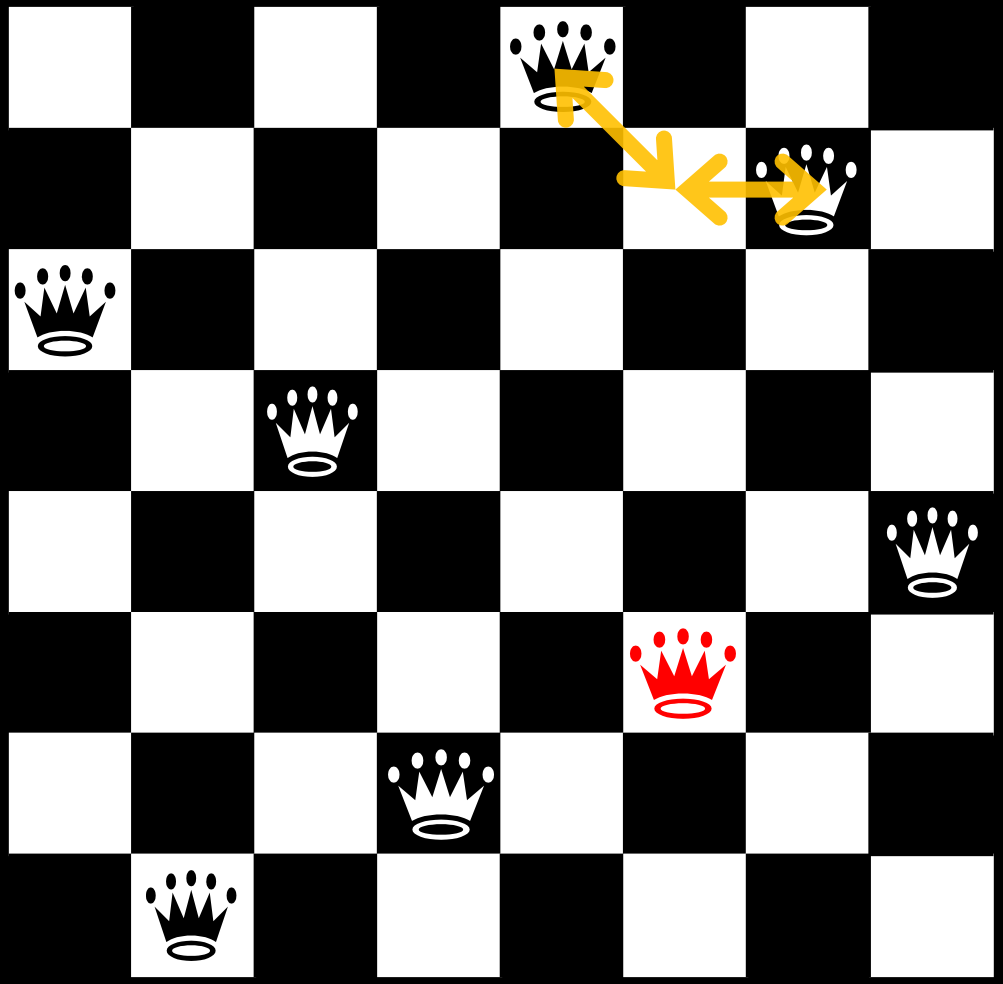
Perfect



Score 0

Eight Queens Problem

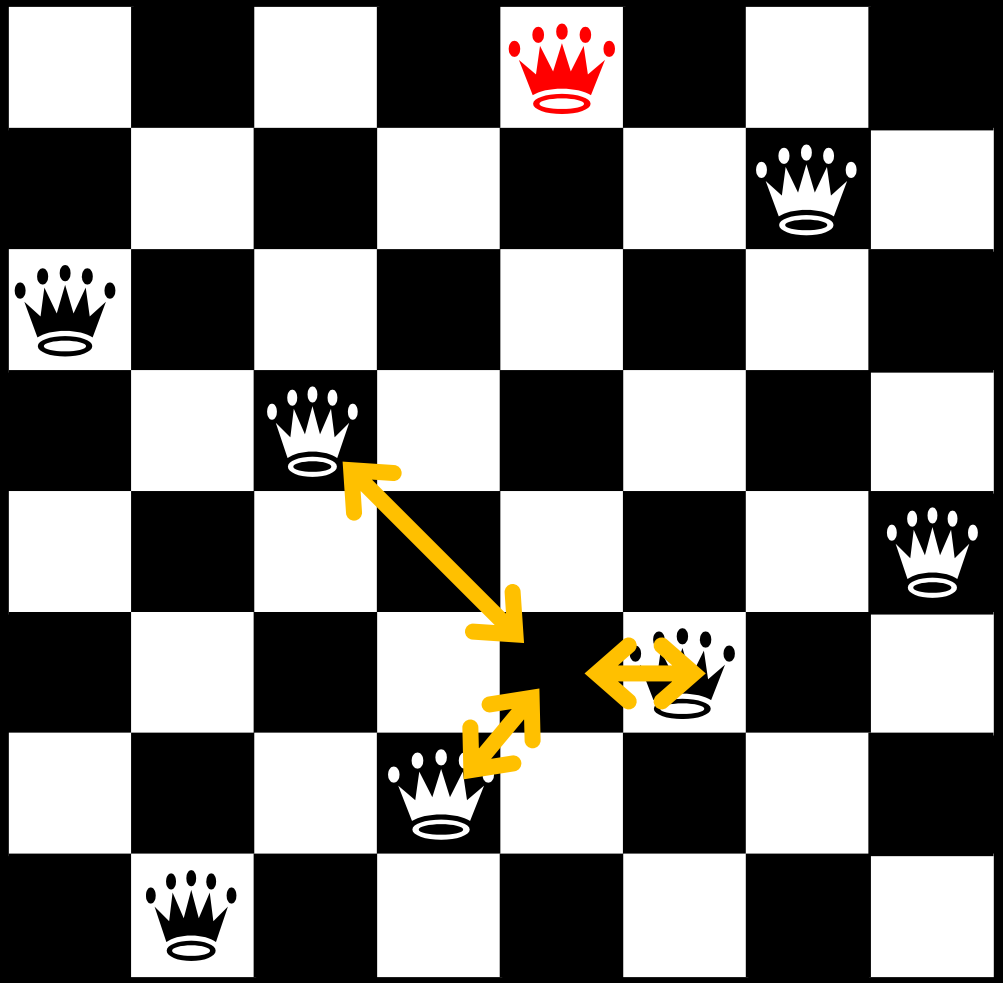
Two Attacks



Score -2

Eight Queens Problem

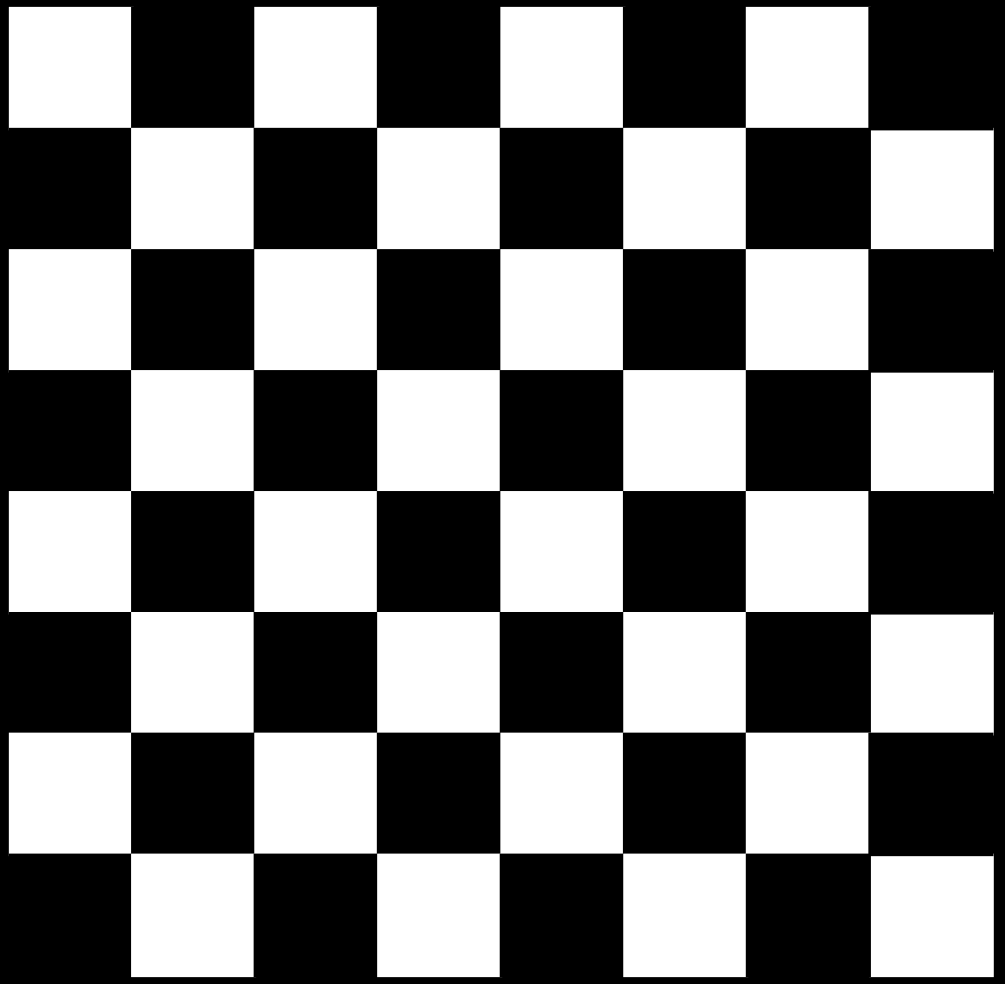
Three Attacks



Score -3

Eight Queens Problem

Place 8 queens on the board



so that there are no attacks

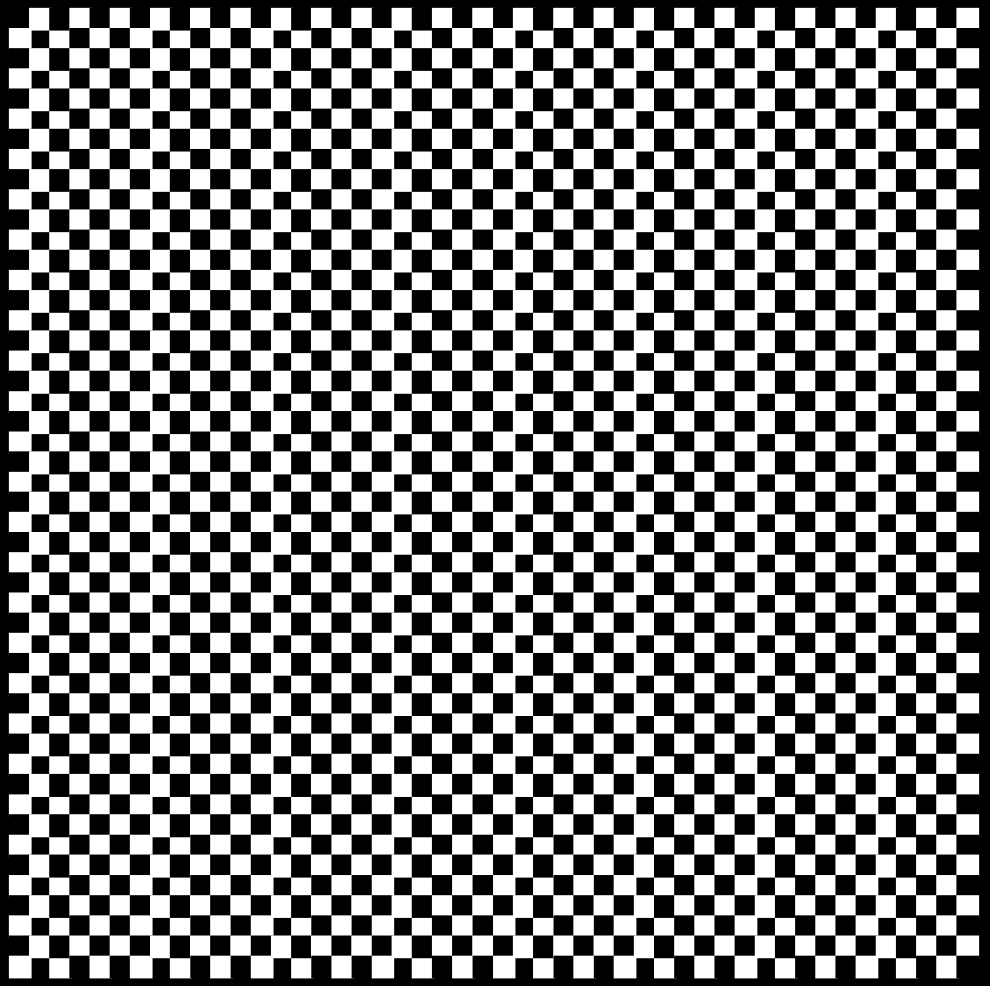
Search from the solution space

Representation

Fitness

Eight Queens Problem

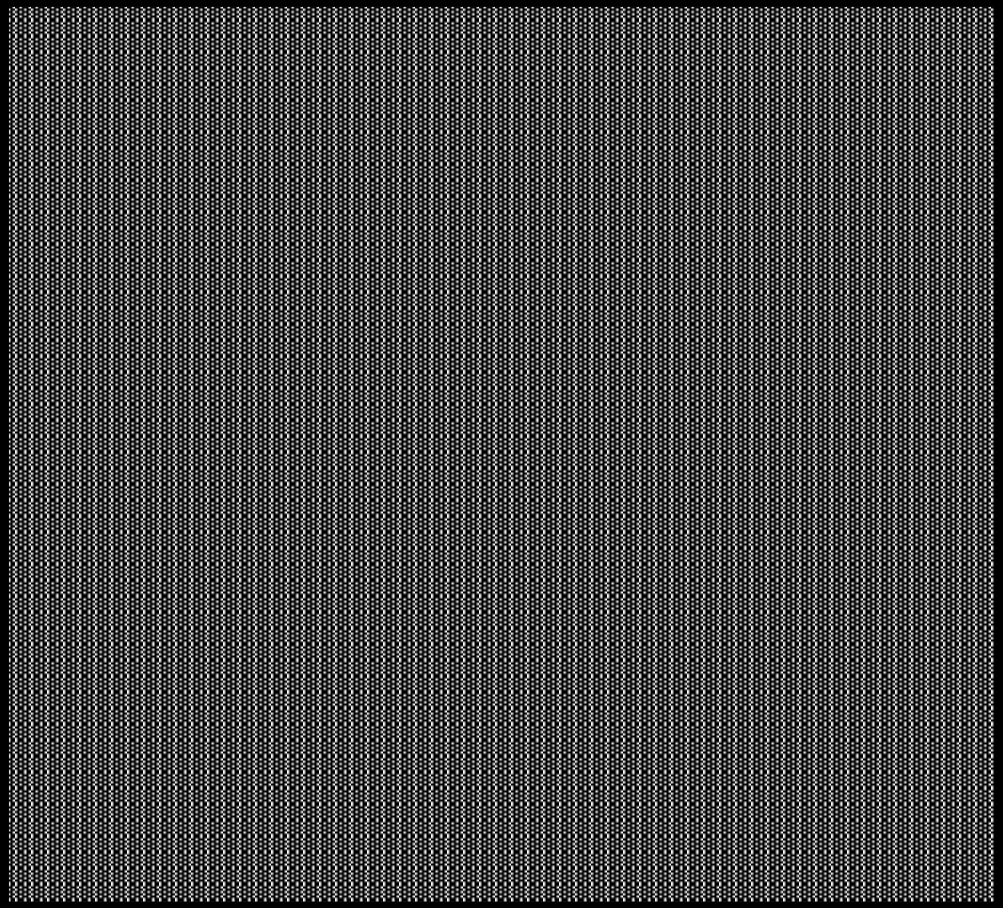
Place 44 queens on the board



so that there are no attacks

Eight Queens Problem

Place 400 queens on the board



so that there are no attacks

Checking vs Generating

Task One:

Write a method to determine which is the better of two placements of N queens

Task Two:

Write a method to construct a board placement with N non attacking queens

Checking vs Generating

Task One:

Write a method to determine which is the better of two placements of N queens

Task Two:

Write a method to construct a board placement with N non attacking queens

Checking vs Generating

Search Based Software Engineering

Write a method to determine which is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering

Write a method to determine which is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering

Write a method to determine which is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering
Write a fitness function to determine which is the better of two solutions

Conventional Software Engineering
Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering
Write a fitness function to guide a search

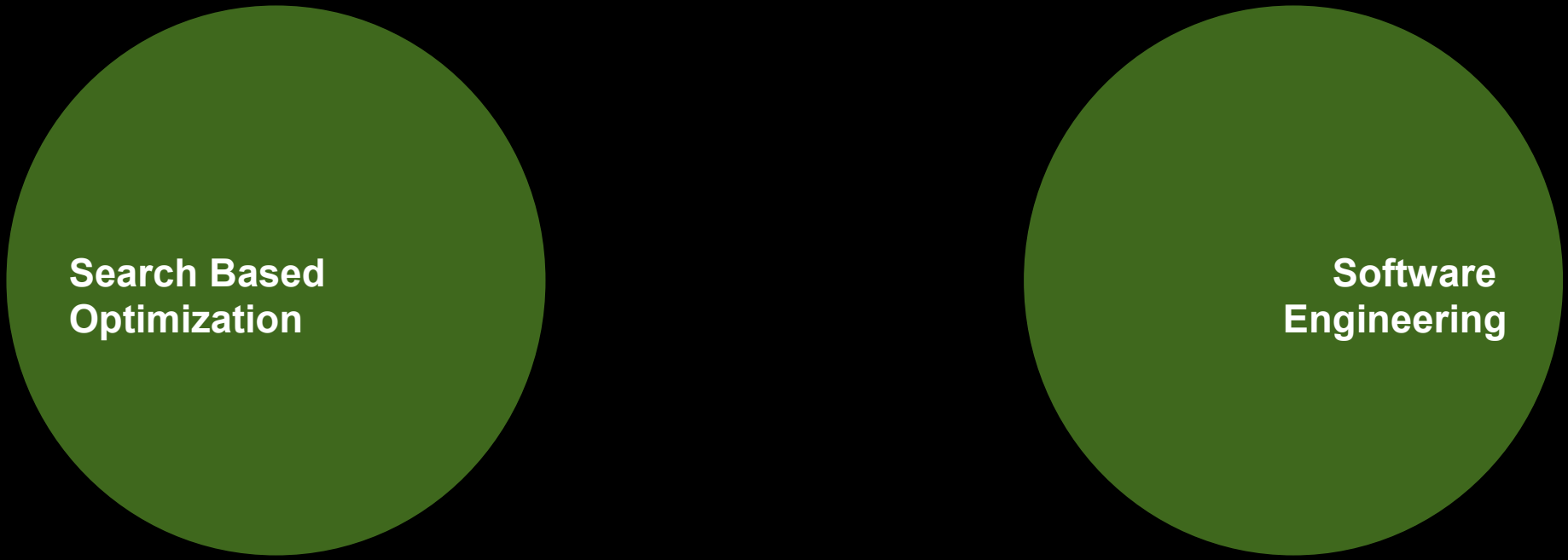
Conventional Software Engineering
Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering
Write a fitness function to guide automated search

Conventional Software Engineering
Write a method to construct a perfect solution

What is SBSE



The advantages of SBSE

n Queens Problem

如何在1分钟内摆放300万个皇后

http://blog.sciencenet.cn/blog-267533-535185.html

The advantages of SBSE

Insight-rich

Scalable

Robust

Generic

Realistic

The advantages of SBSE

Insight-rich

Scalable

Robust

Generic

Realistic

The advantages of SBSE

Insight-rich

Scalable

Robust

Generic

Realistic

The advantages of SBSE

Insight-rich

Scalable

Robust

Generic

Realistic

The advantages of SBSE

Insight-rich

Scalable

Robust

Generic

Realistic

The advantages of SBSE

Insight-rich

Scalable

Robust

Generic

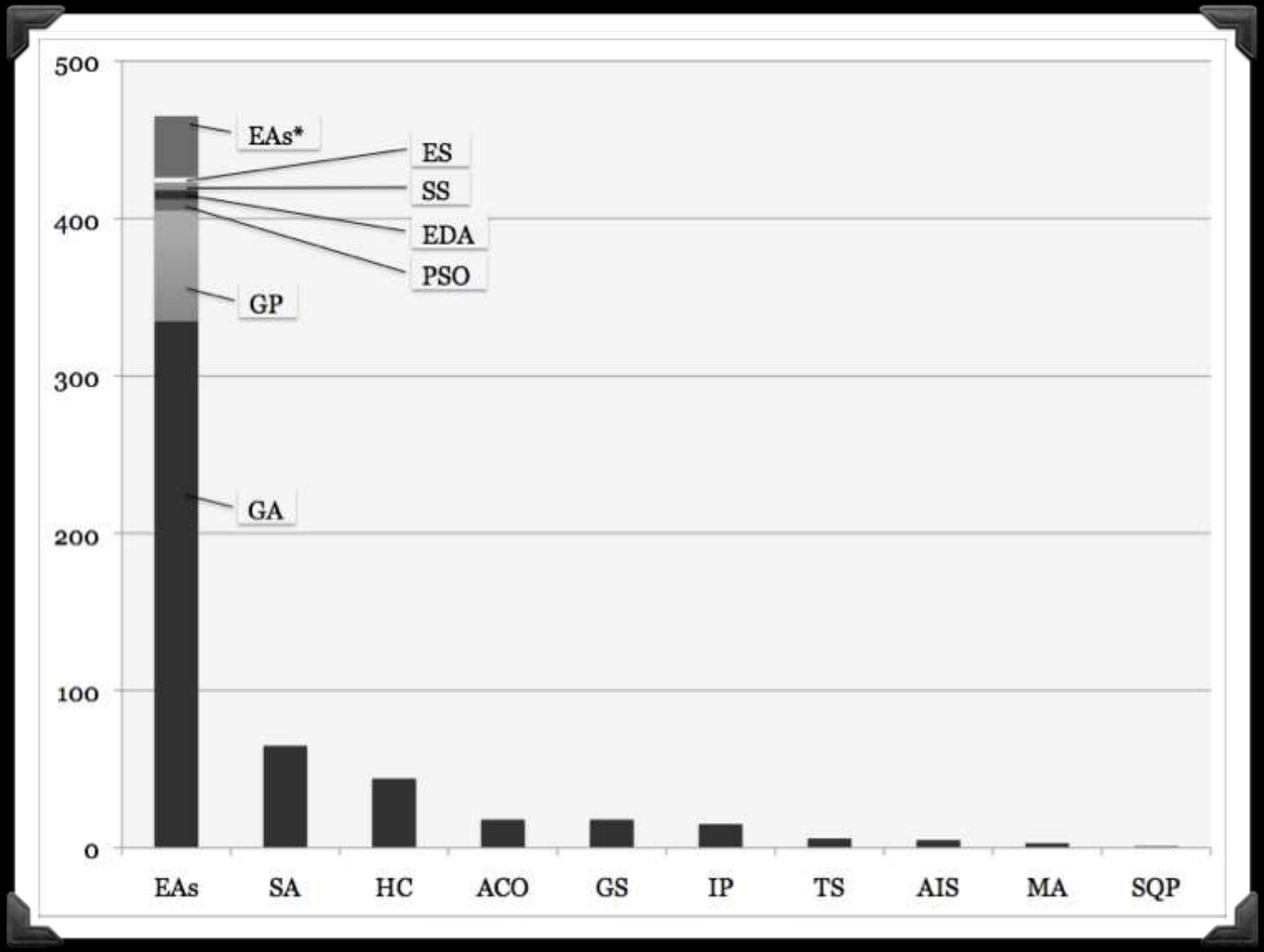
Realistic

Evolutionary Algorithms

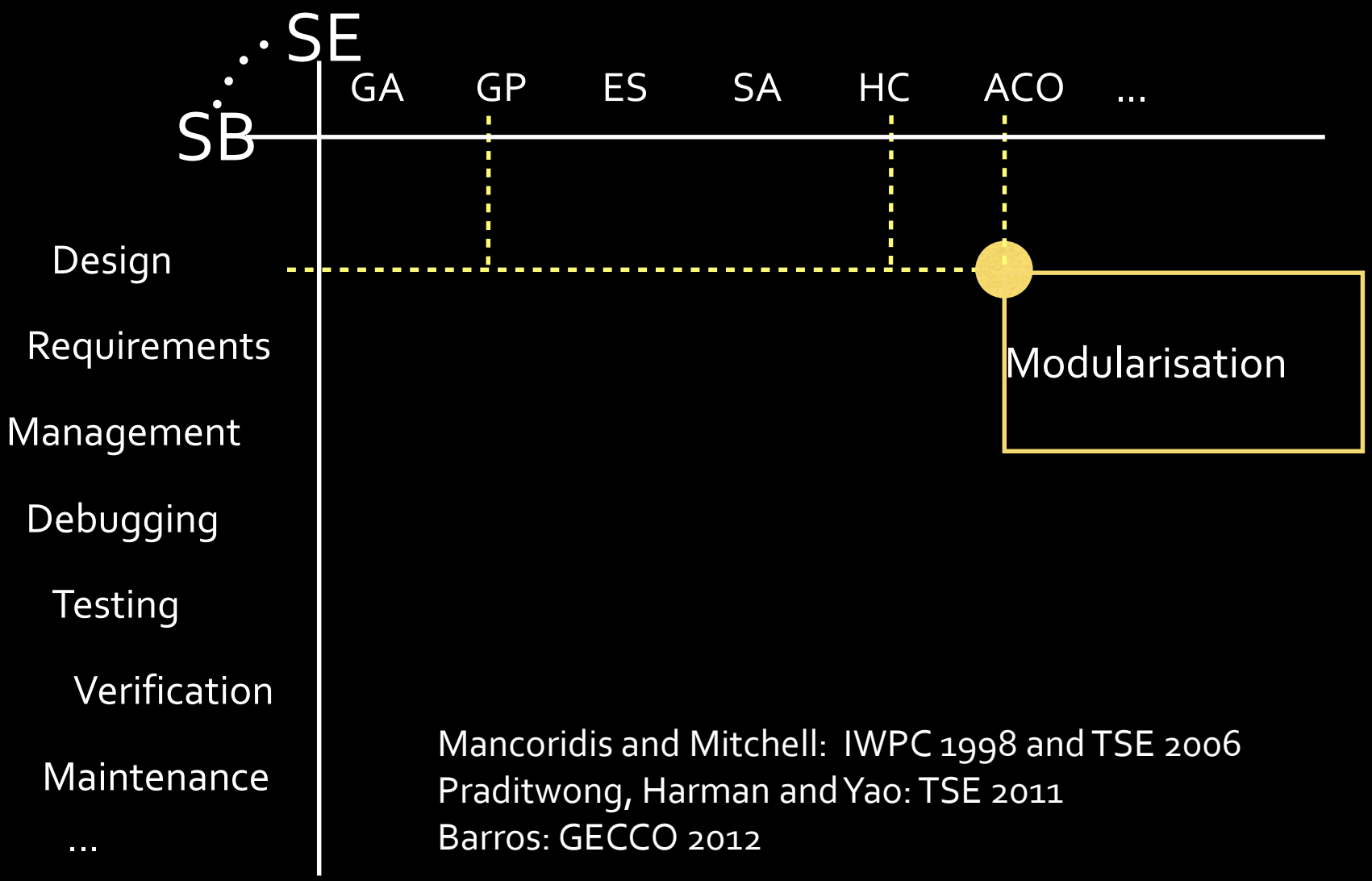
about 70% of all SBSE

uses evolutionary computation

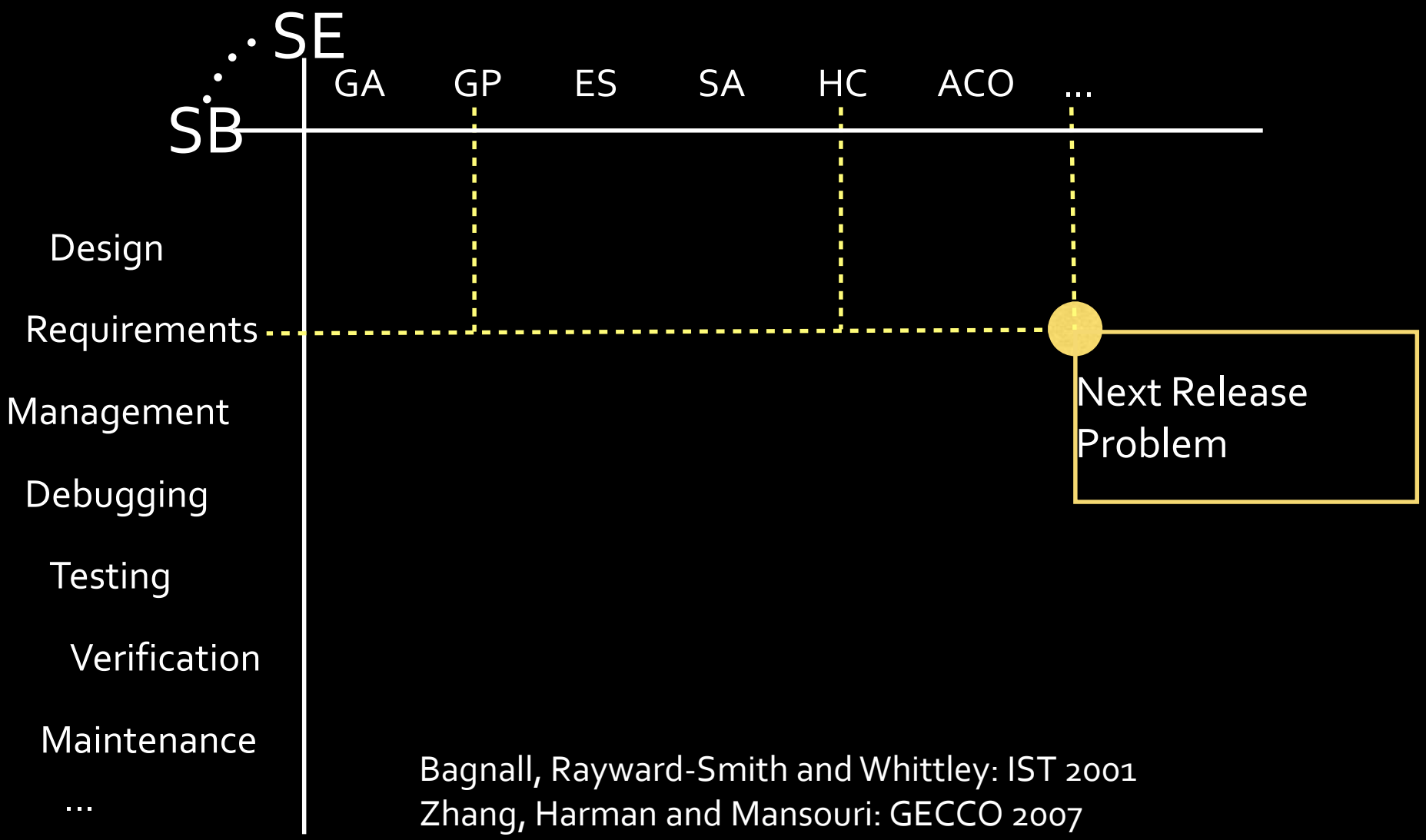
Search Based Algorithms Used



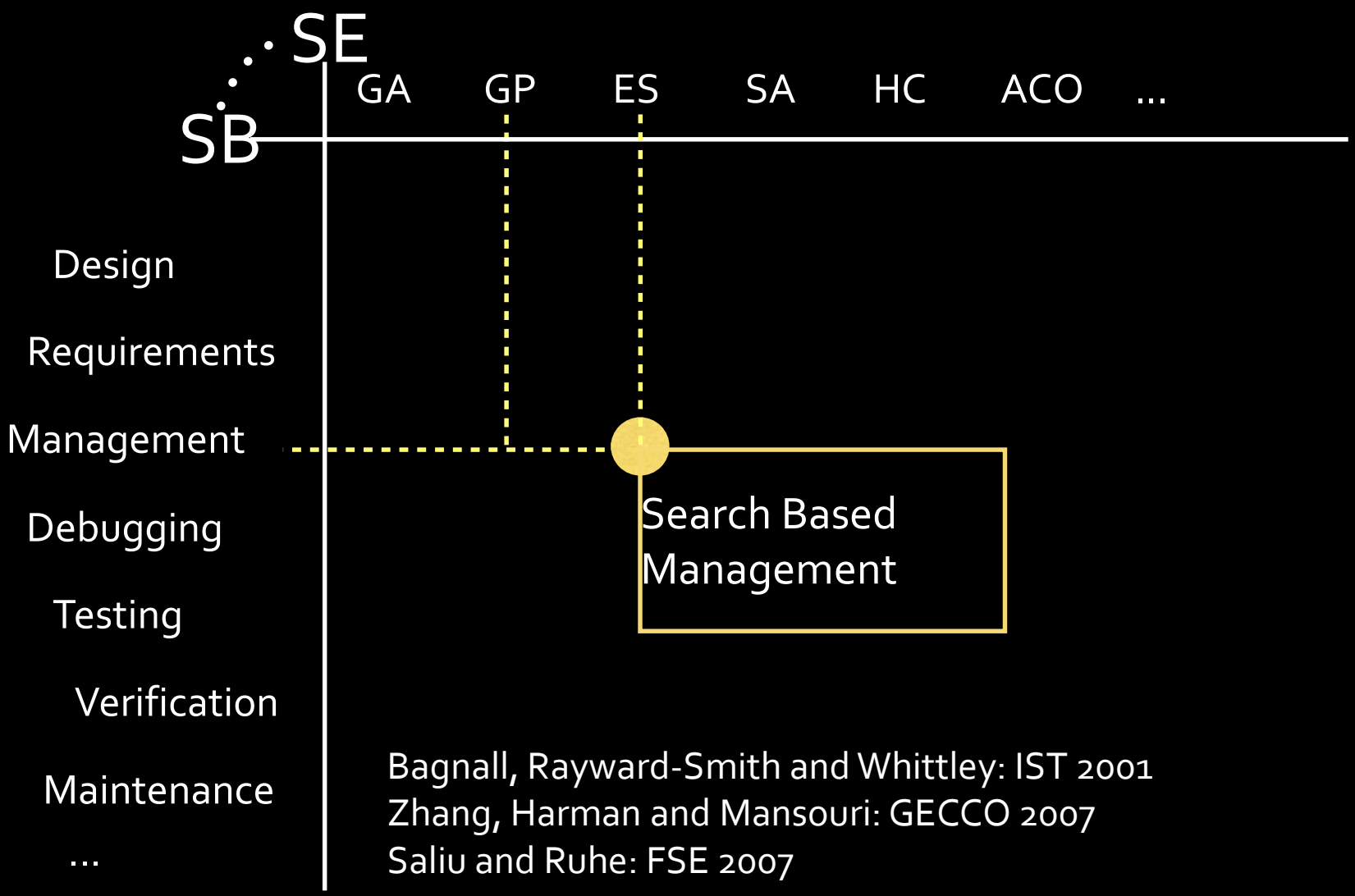

SE
SB
GA
GP
ES
SA
HC
ACO
...
Design
Requirements
Management
Debugging
Testing
Verification
Maintenance
...



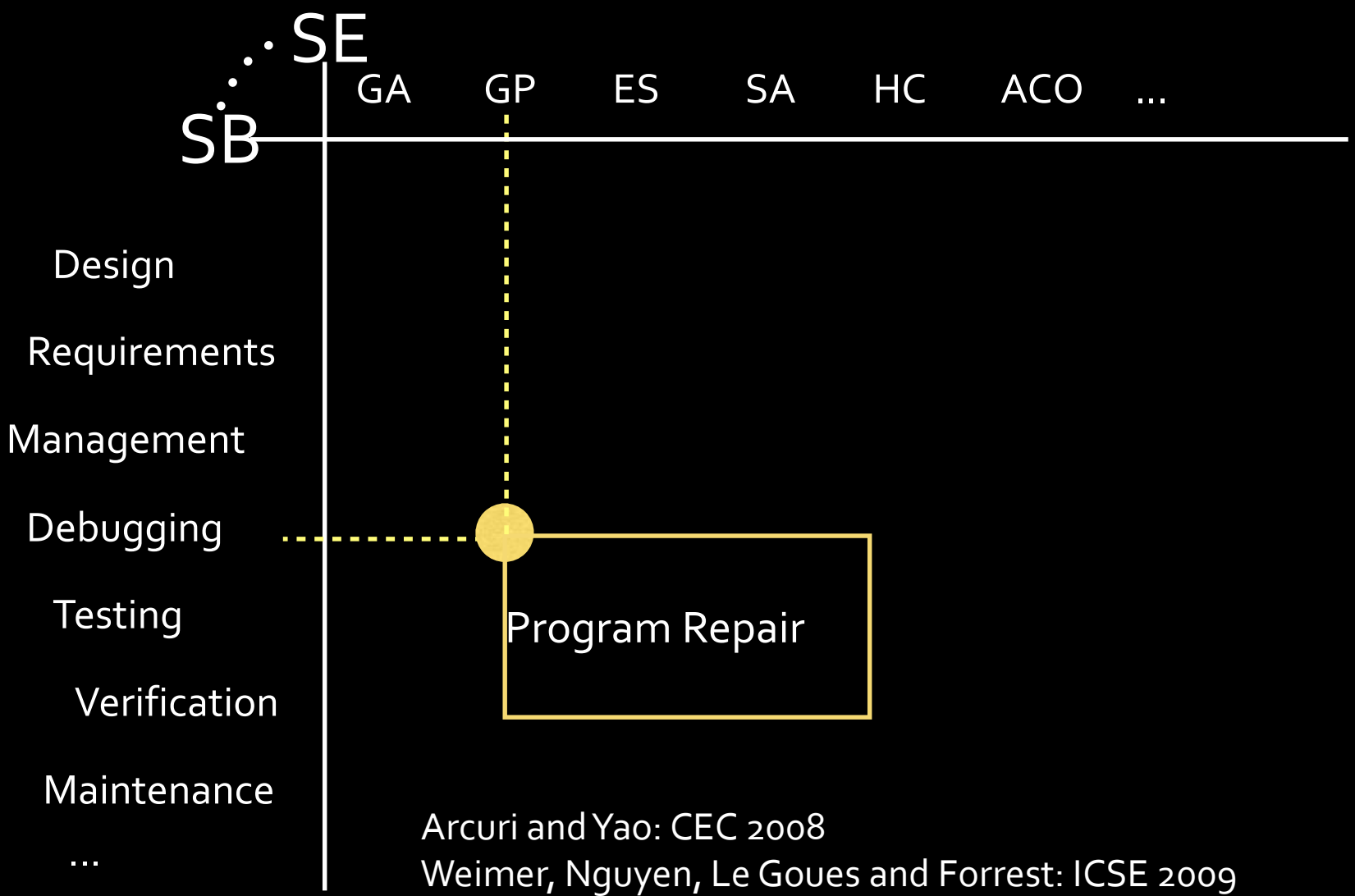
Mancoridis and Mitchell: IWPC 1998 and TSE 2006
Praditwong, Harman and Yao: TSE 2011
Barros: GECCO 2012



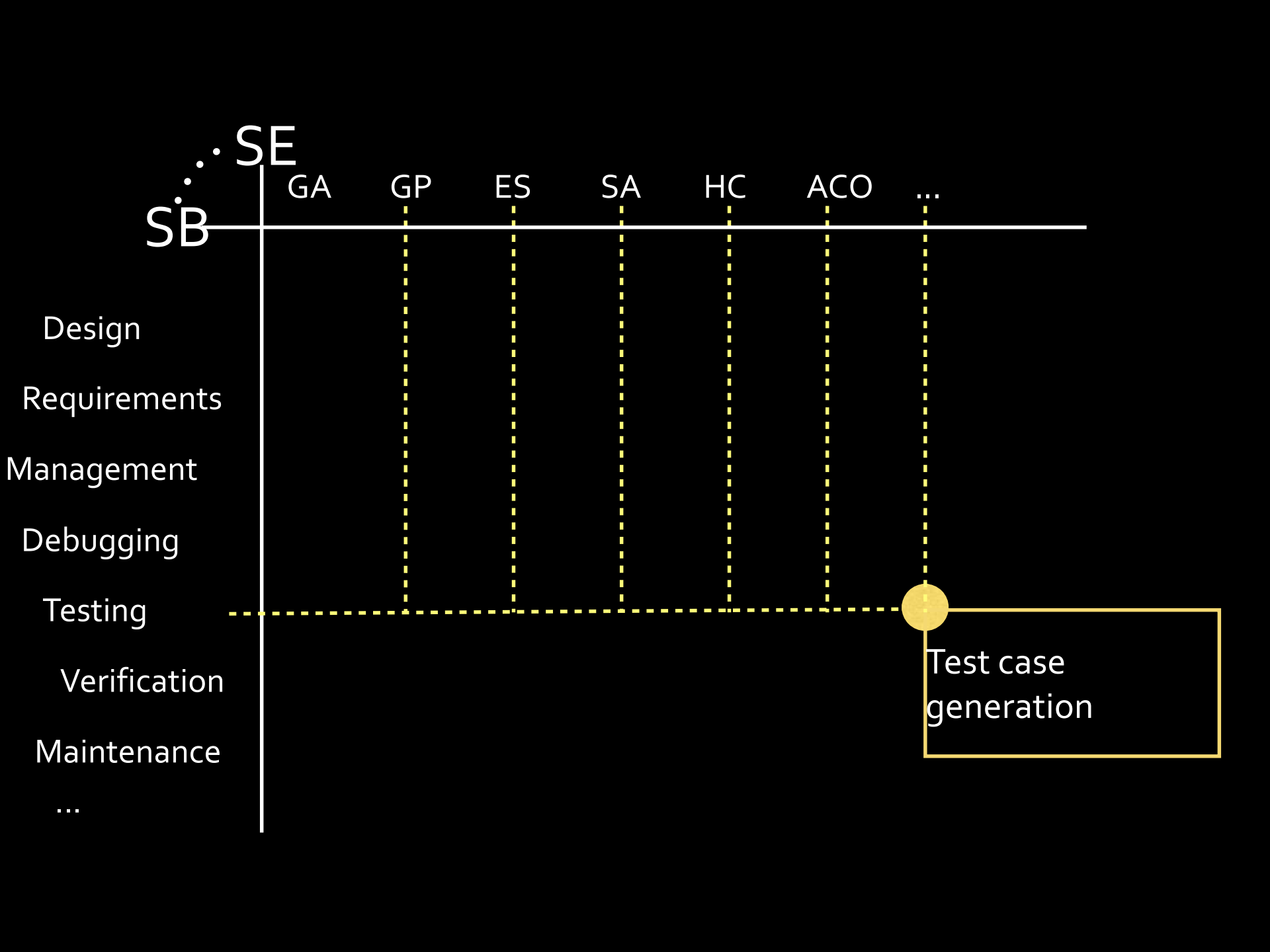
Bagnall, Rayward-Smith and Whittley: IST 2001
Zhang, Harman and Mansouri: GECCO 2007
Saliu and Ruhe: FSE 2007

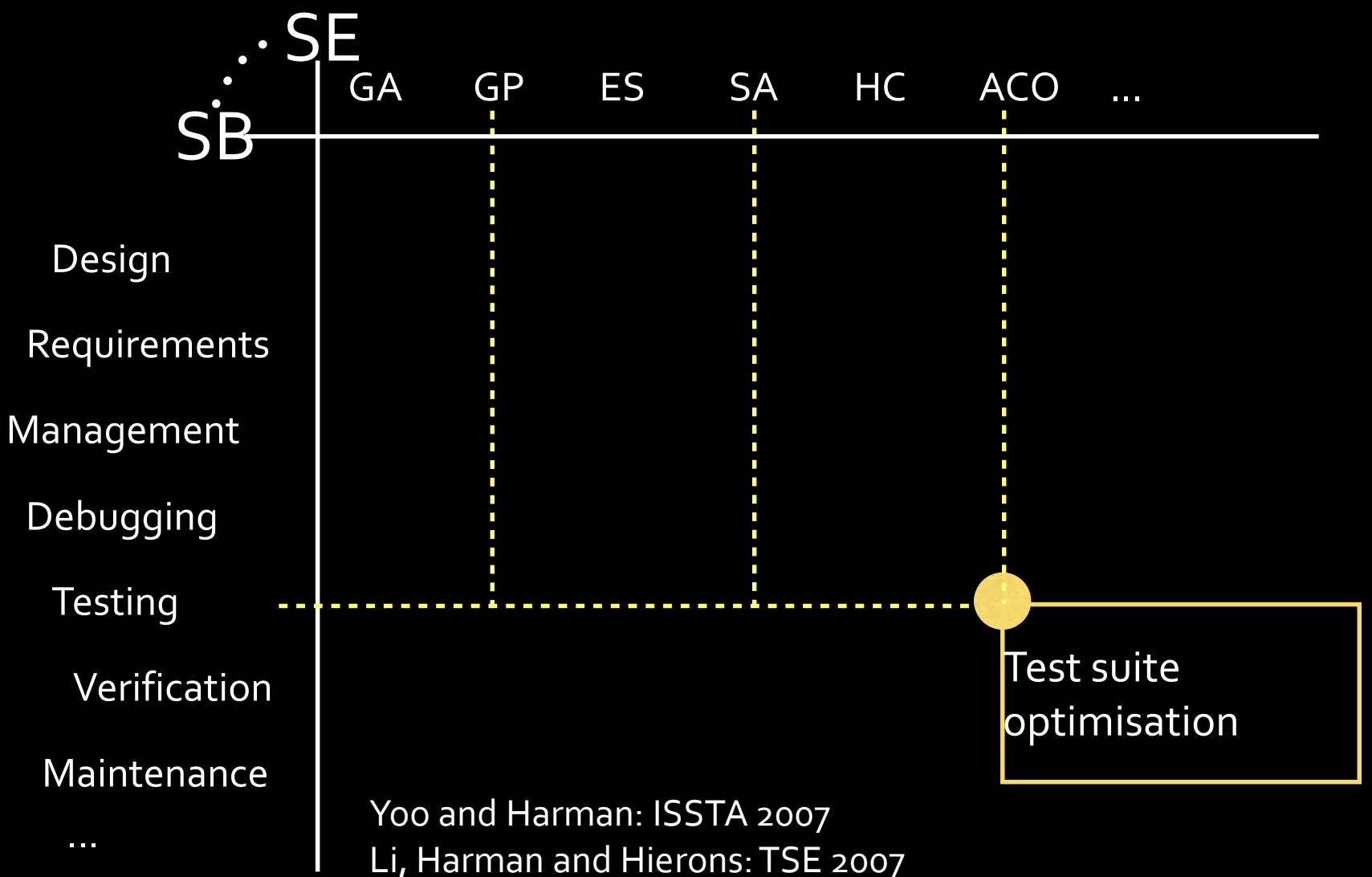


Bagnall, Rayward-Smith and Whittley: IST 2001
Zhang, Harman and Mansouri: GECCO 2007
Saliu and Ruhe: FSE 2007

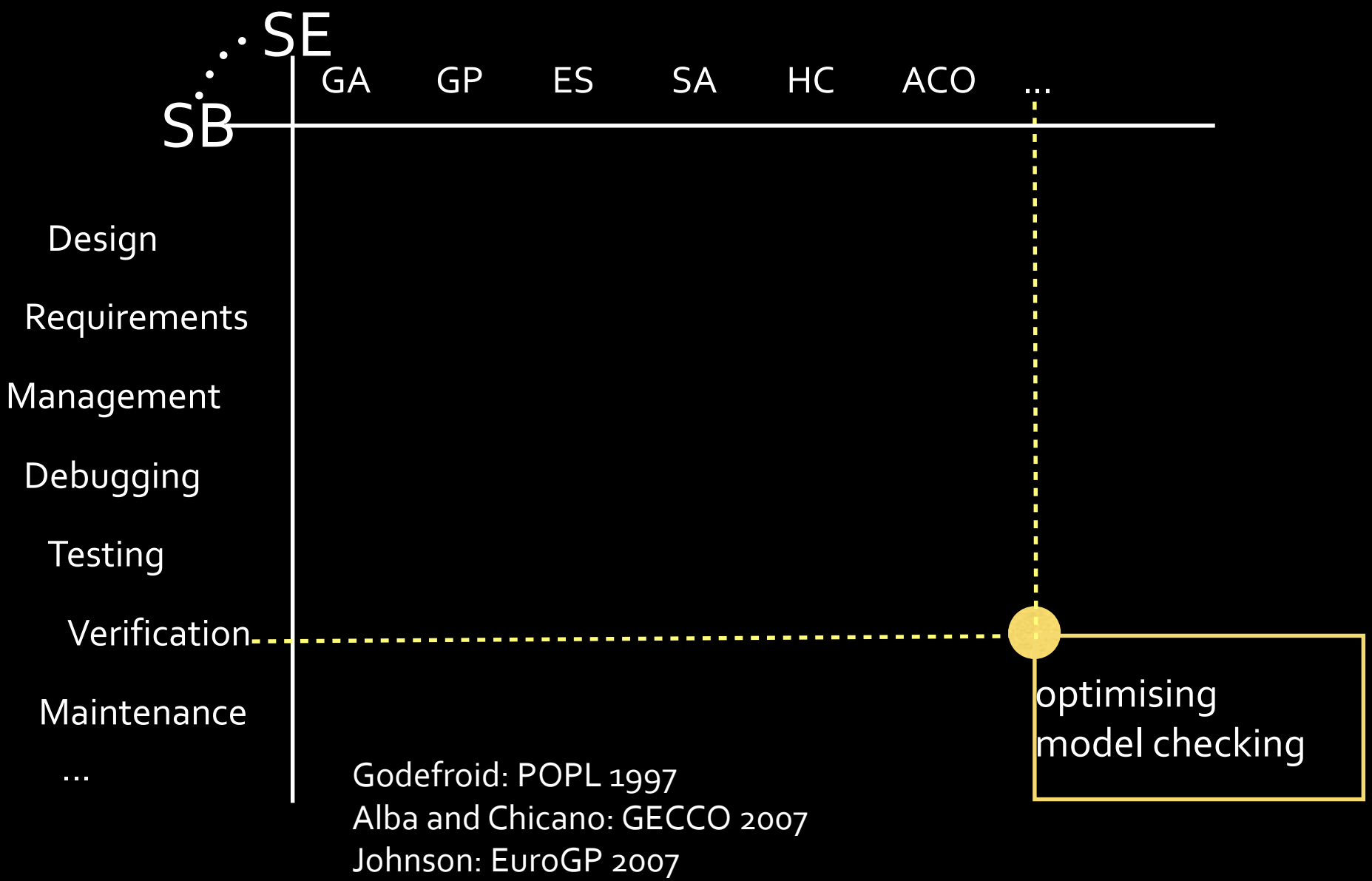


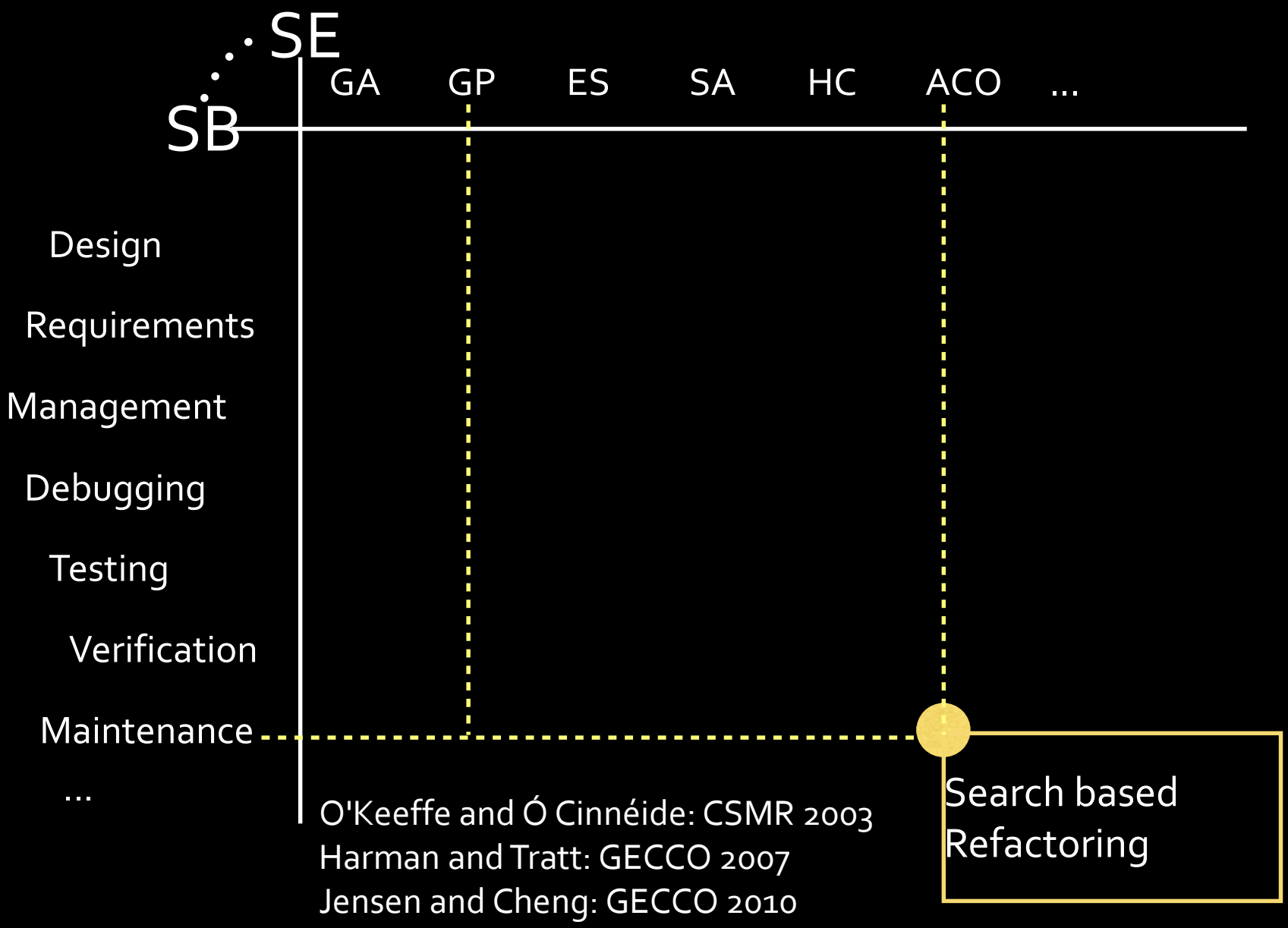
Arcuri and Yao: CEC 2008
Weimer, Nguyen, Le Goues and Forrest: ICSE 2009
Kim, Nam, Song and Kim: ICSE 2013


SE
SB
GA
GP
ES
SA
HC
ACO
...
Design
Requirements
Management
Debugging
Testing
Verification
Maintenance
...
Test case generation



Yoo and Harman: ISSTA 2007
Li, Harman and Hierons: TSE 2007
Mirarab, Akhlaghi and Tahvildari: TSE 2012


SE
SB
GA
GP
ES
SA
HC
ACO
...
Design
Requirements
Management
Debugging
Testing
Verification
Maintenance
...
optimising
model checking
Godefroid: POPL 1997
Alba and Chicano: GECCO 2007
Johnson: EuroGP 2007


SE
SB
GA
GP
ES
SA
HC
ACO
...
Design
Requirements
Management
Debugging
Testing
Verification
Maintenance
...
O'Keeffe and Ó Cinnéide: CSMR 2003
Harman and Tratt: GECCO 2007
Jensen and Cheng: GECCO 2010
Search based
Refactoring

Examples

Single Simple Objective

☺Optimising Temporal Testing

☺Structural Testing

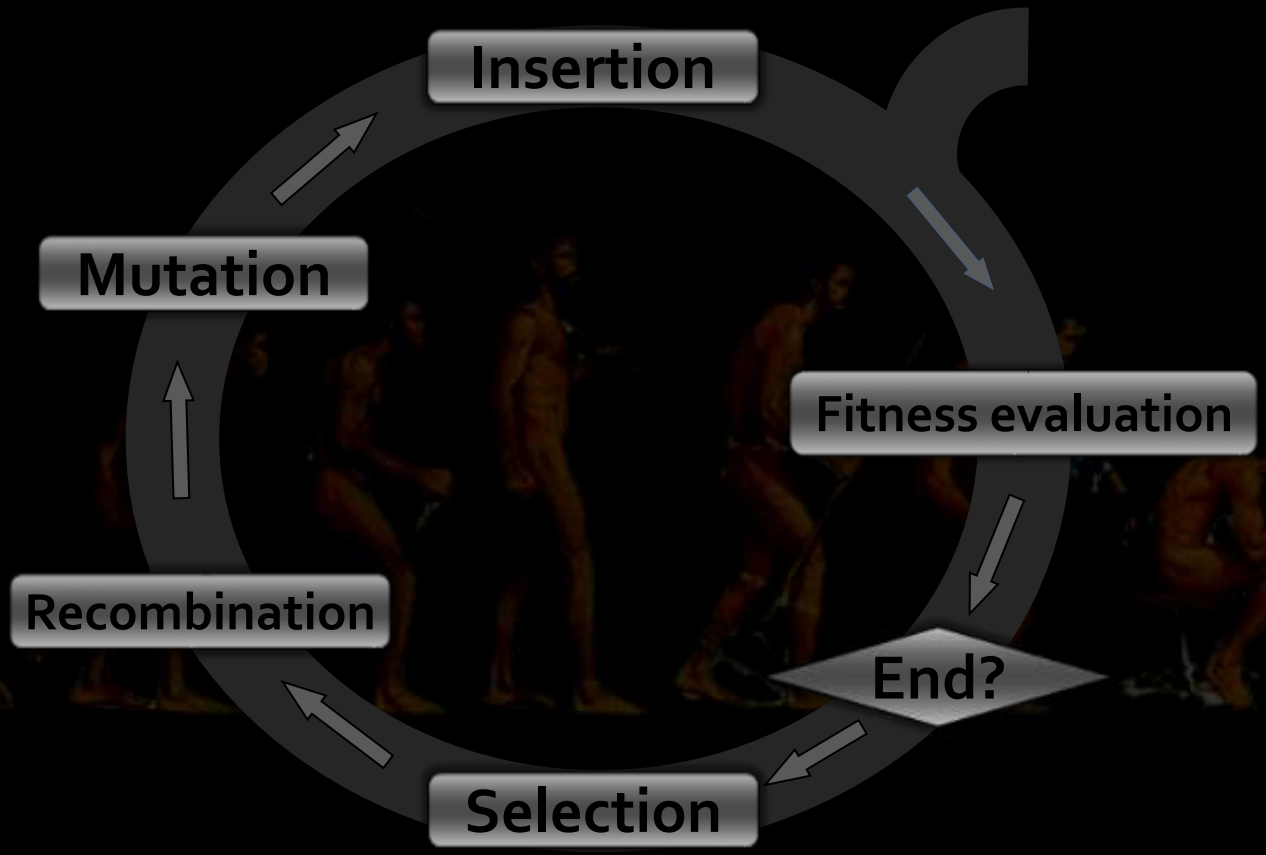
Temporal testing

Verifying timing constraints of real-time systems by means of evolutionary testing

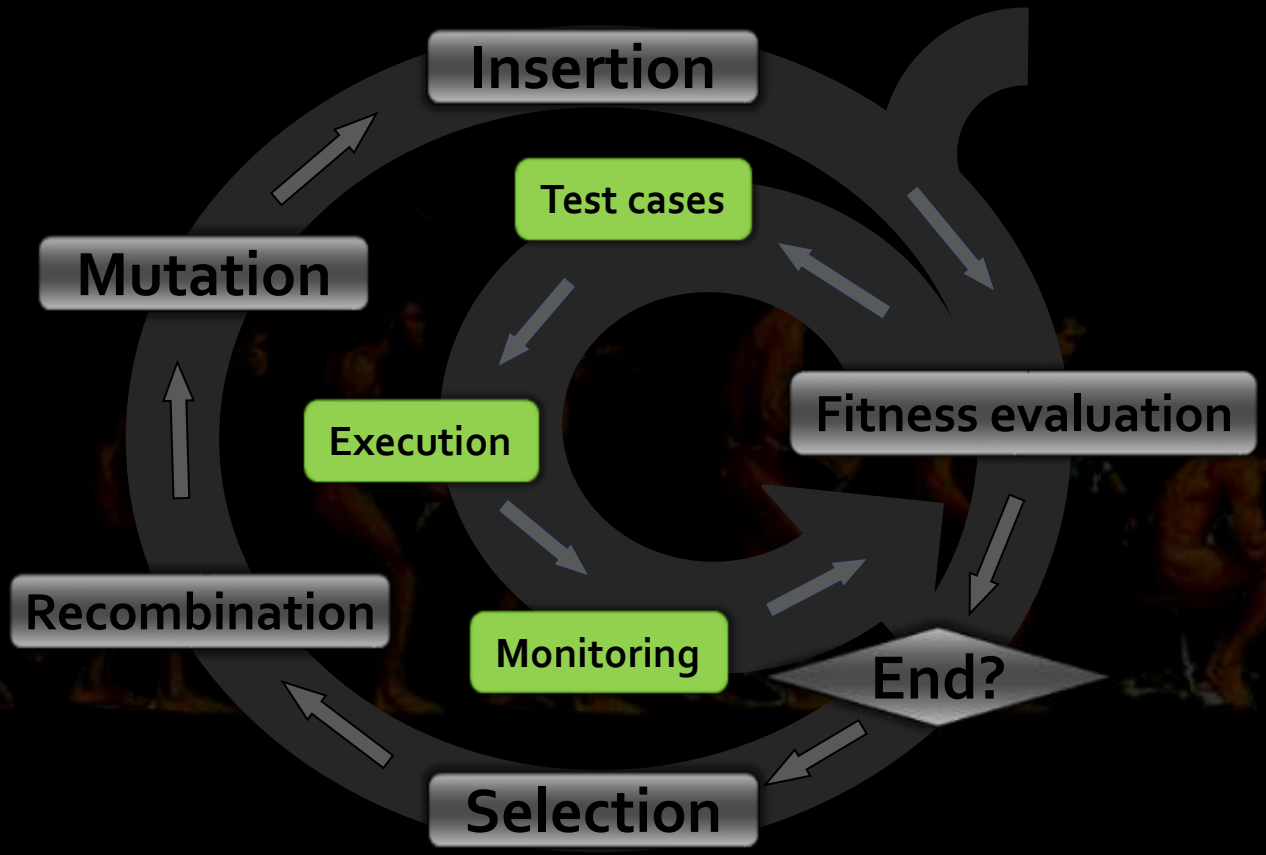
Joachim Wegener and Matthias Grochtmann. Real Time Systems 1998.

Evolutionary Testing

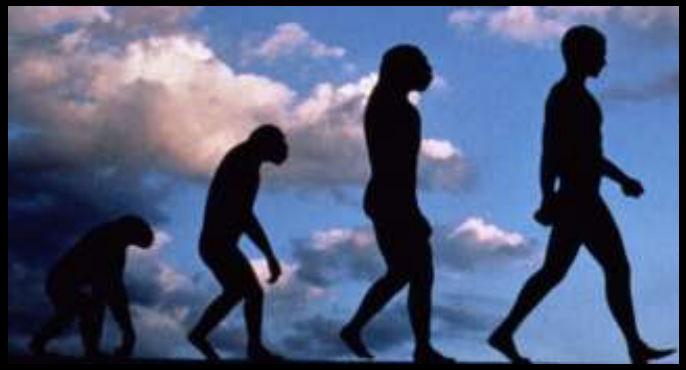
Evolutionary Testing

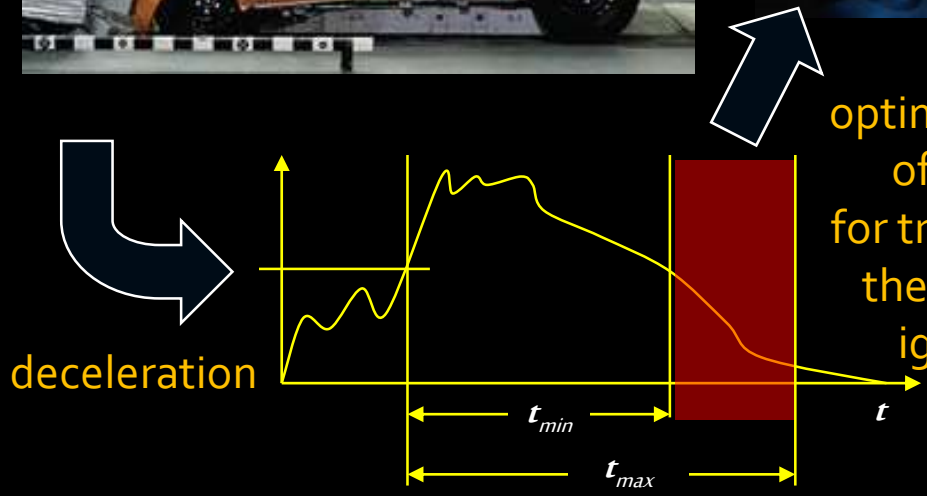


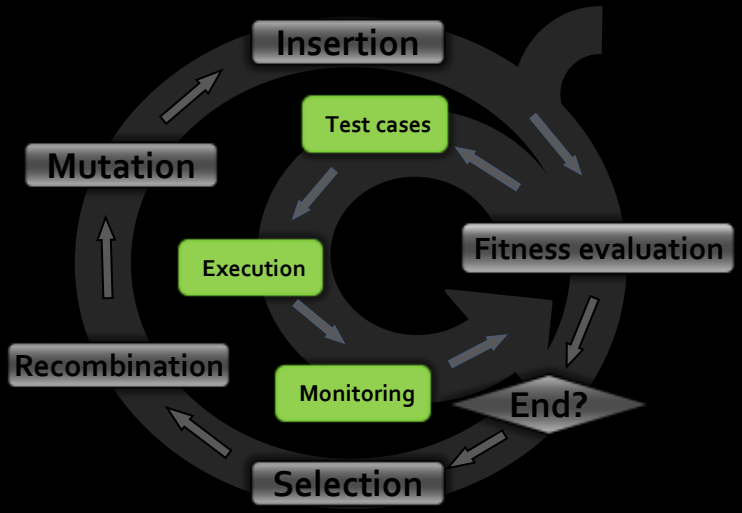
Evolutionary Testing



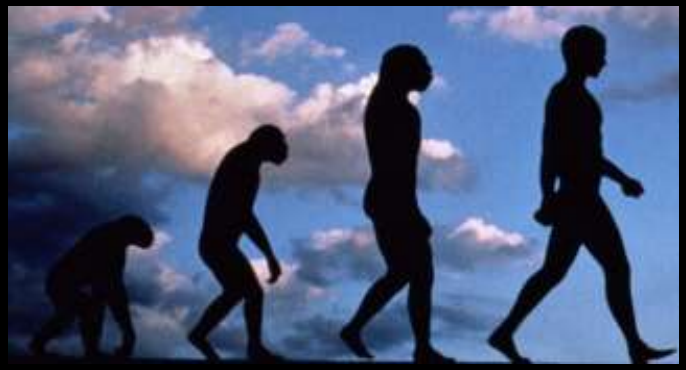
Daimler Temporal Testing

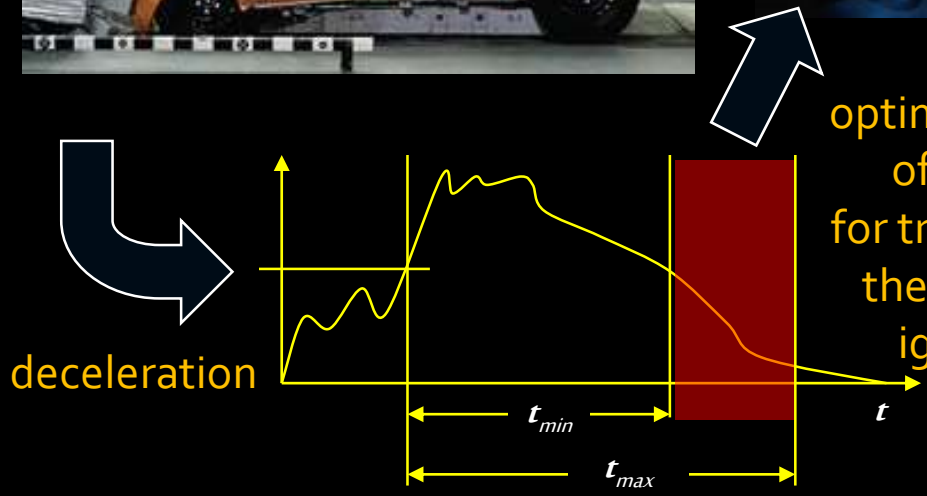


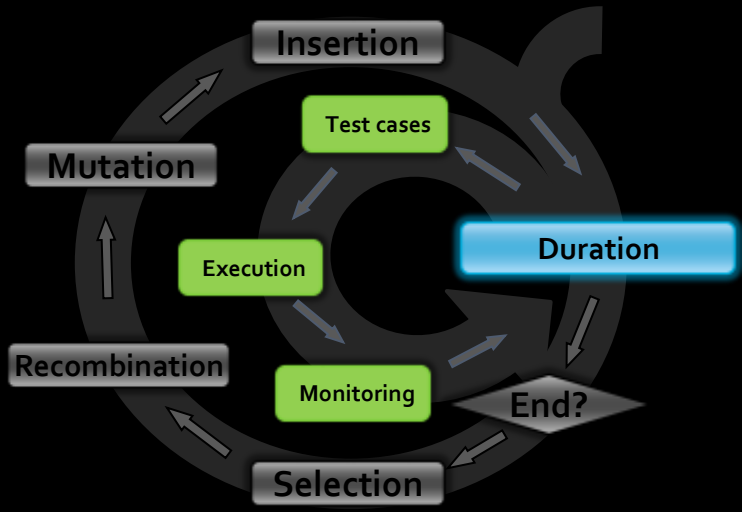




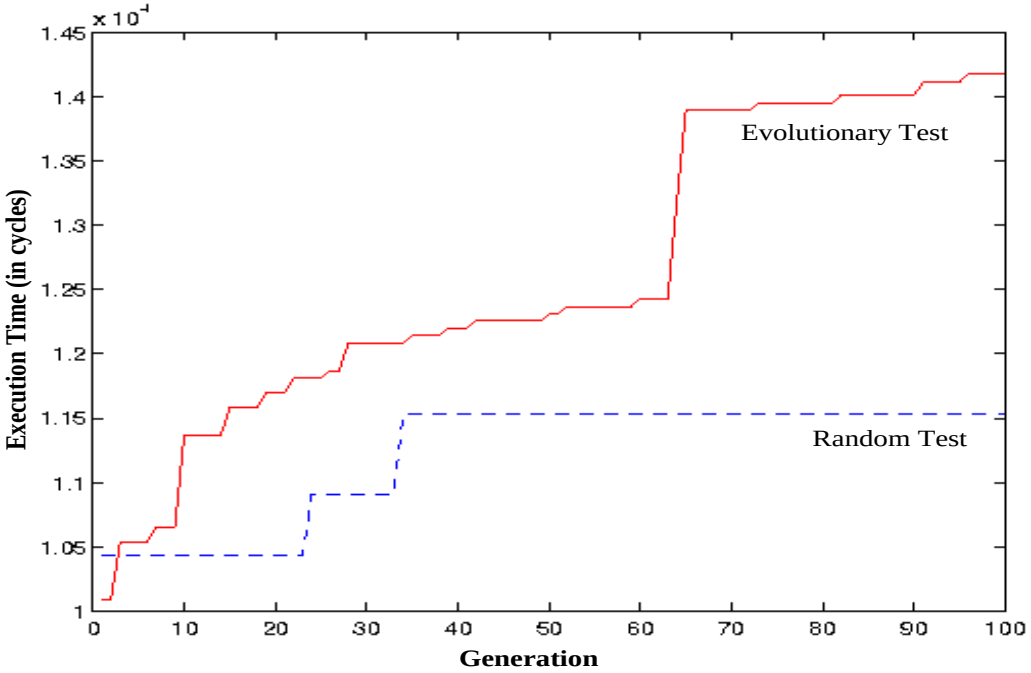
Daimler Temporal Testing

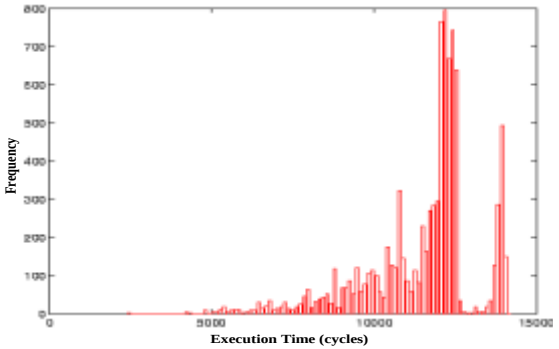


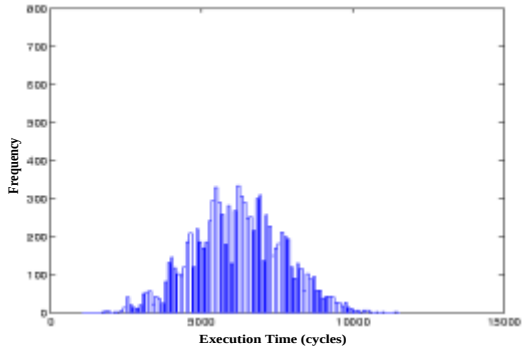




Evolution vs Random for Temporal Testing

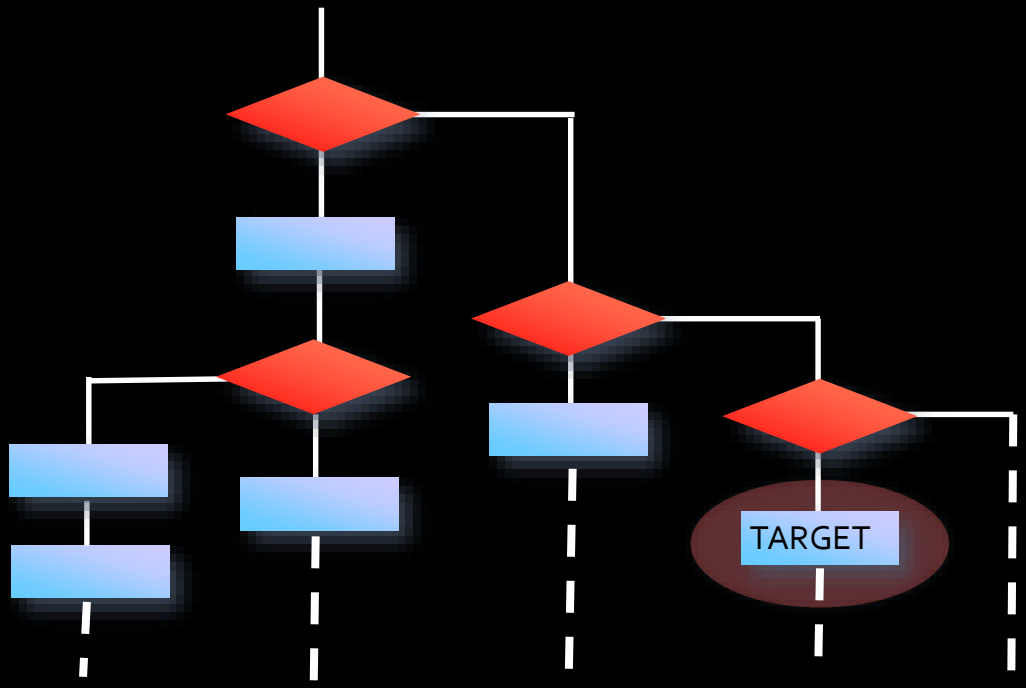






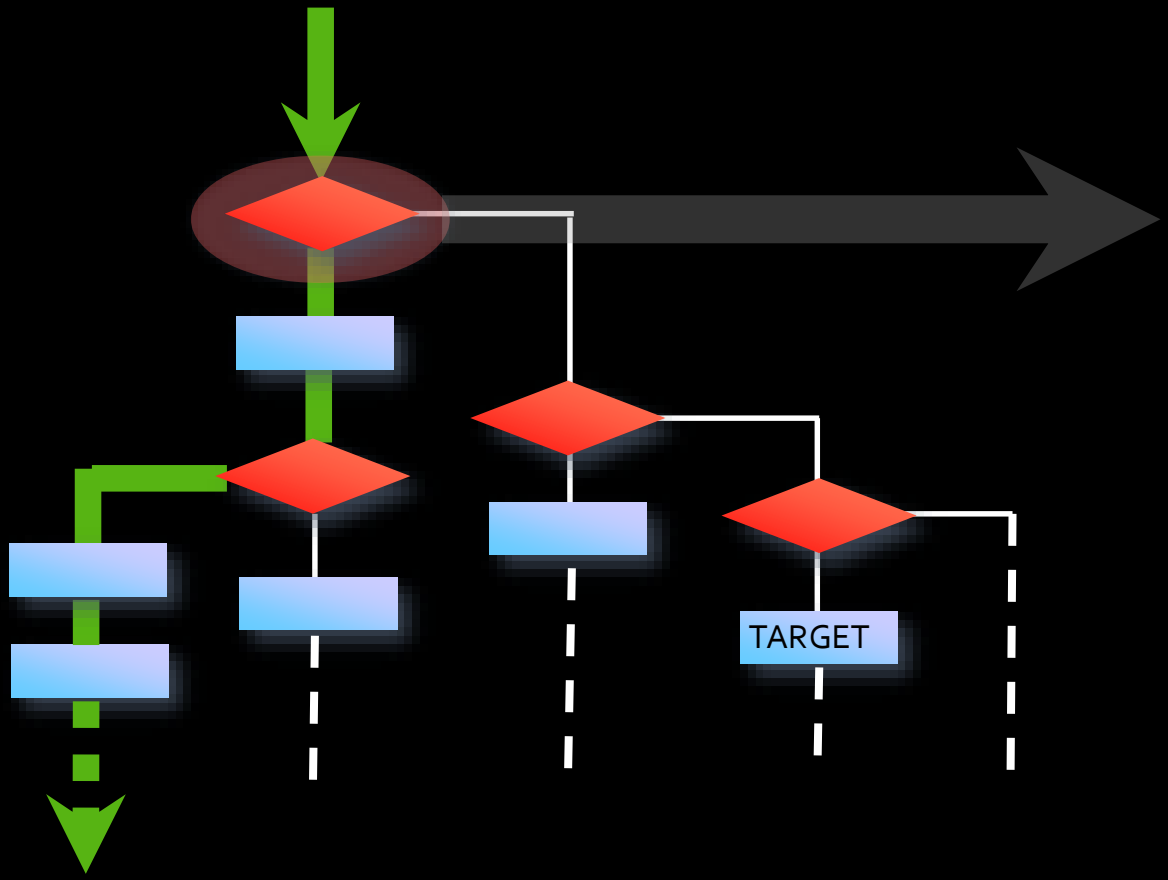
Optimising Structural Testing

Program Coverage



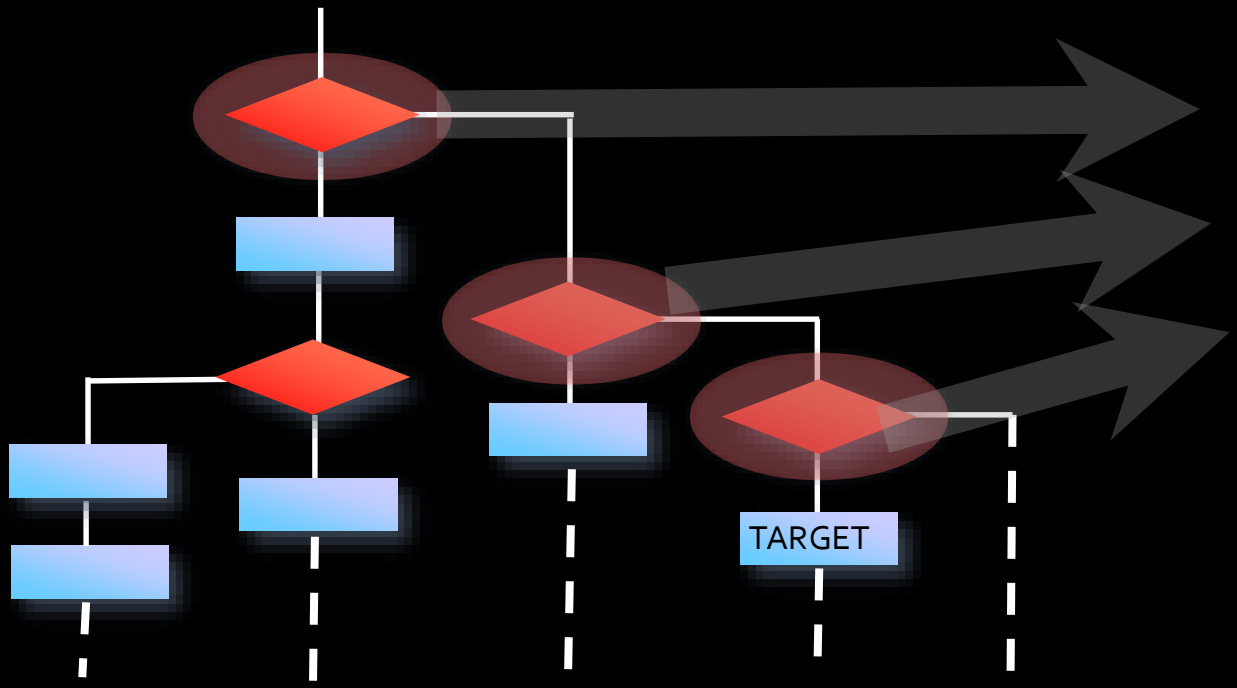
control flow graph of a program containing a structure we want to cover

Fitness evaluation



The test data executes the 'wrong' path

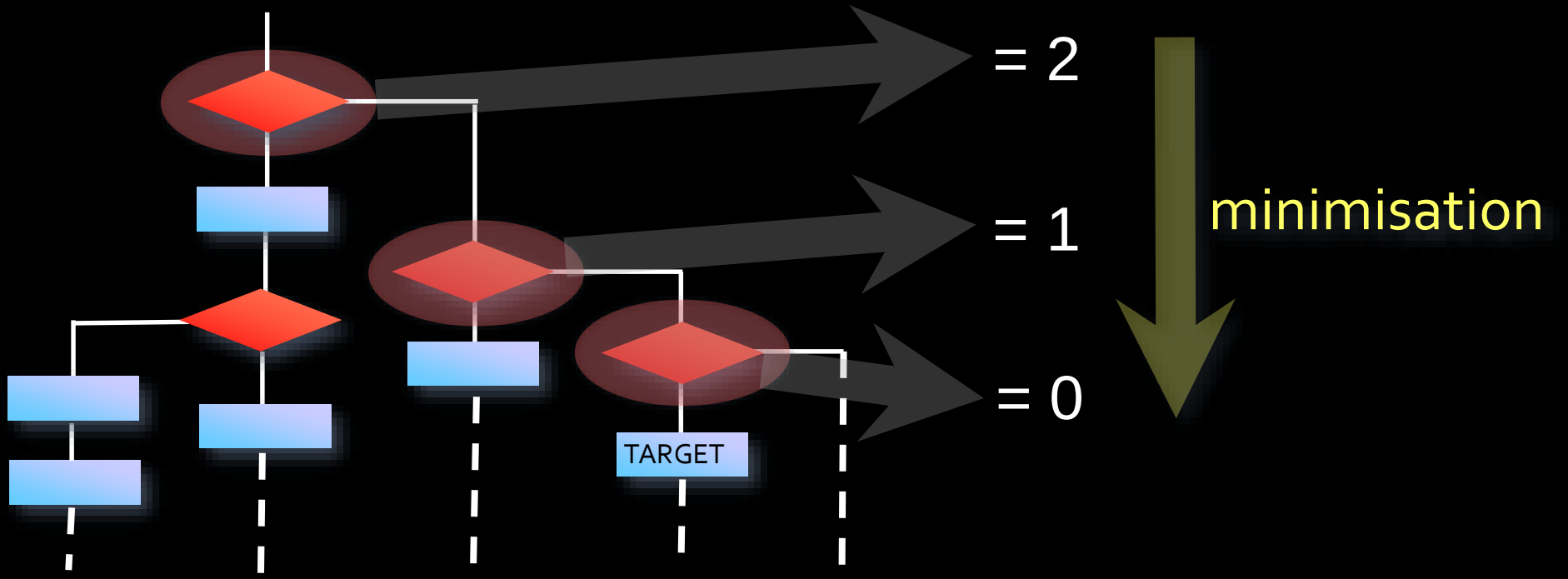
Analysing control flow



The outcomes at key decision statements matter.

These are the decisions on which the target is control dependent

Approach Level



Analysing predicates

Approach level alone gives us coarse values

```
if (a == b) {
    // ....
}
```

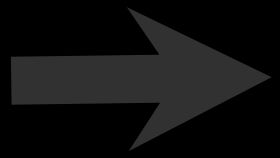
a = 50, b = 0
a = 45, b = 5
a = 40, b = 10
a = 35, b = 15
a = 30, b = 20
a = 25, b = 25

getting 'closer' to being true

Branch distance

Associate a distance formula with different relational predicates

```
if (a == b) {
    // ....
}
```



```
abs(a-b)
```

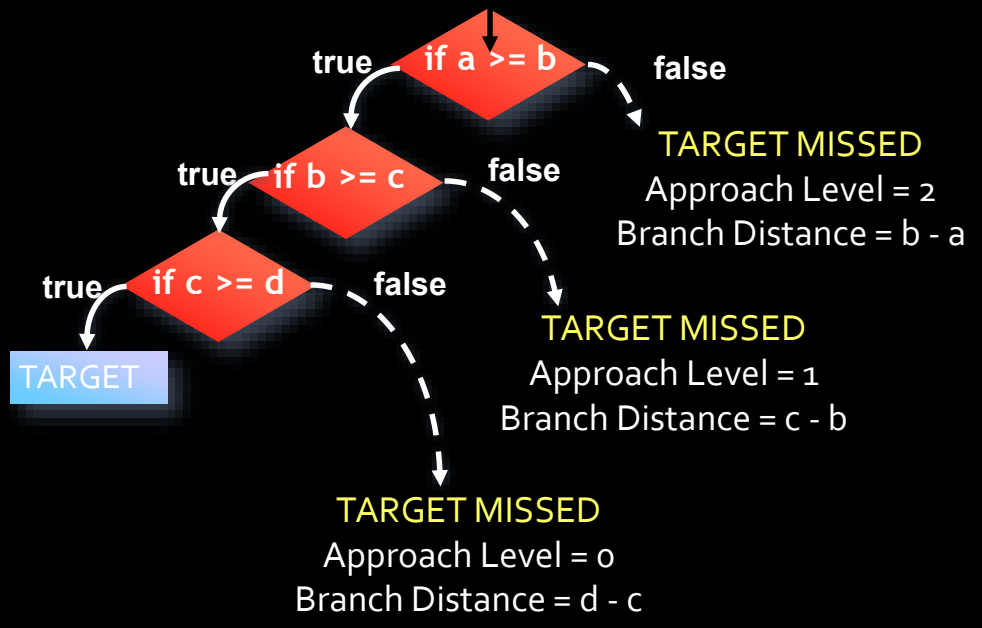
a = 50, b = 0	branch distance = 50
a = 45, b = 5	branch distance = 40
a = 40, b = 10	branch distance = 30
a = 35, b = 15	branch distance = 20
a = 30, b = 20	branch distance = 10
a = 25, b = 25	branch distance = 0

getting 'closer' to being true

Putting it all together

Fitness = approach Level + *normalised* branch distance

```
void f1(int a, int b, int c, int d)
{
    if (a > b)
    {
        if (b > c)
        {
            if (c > d)
            {
                // target
. . .
```

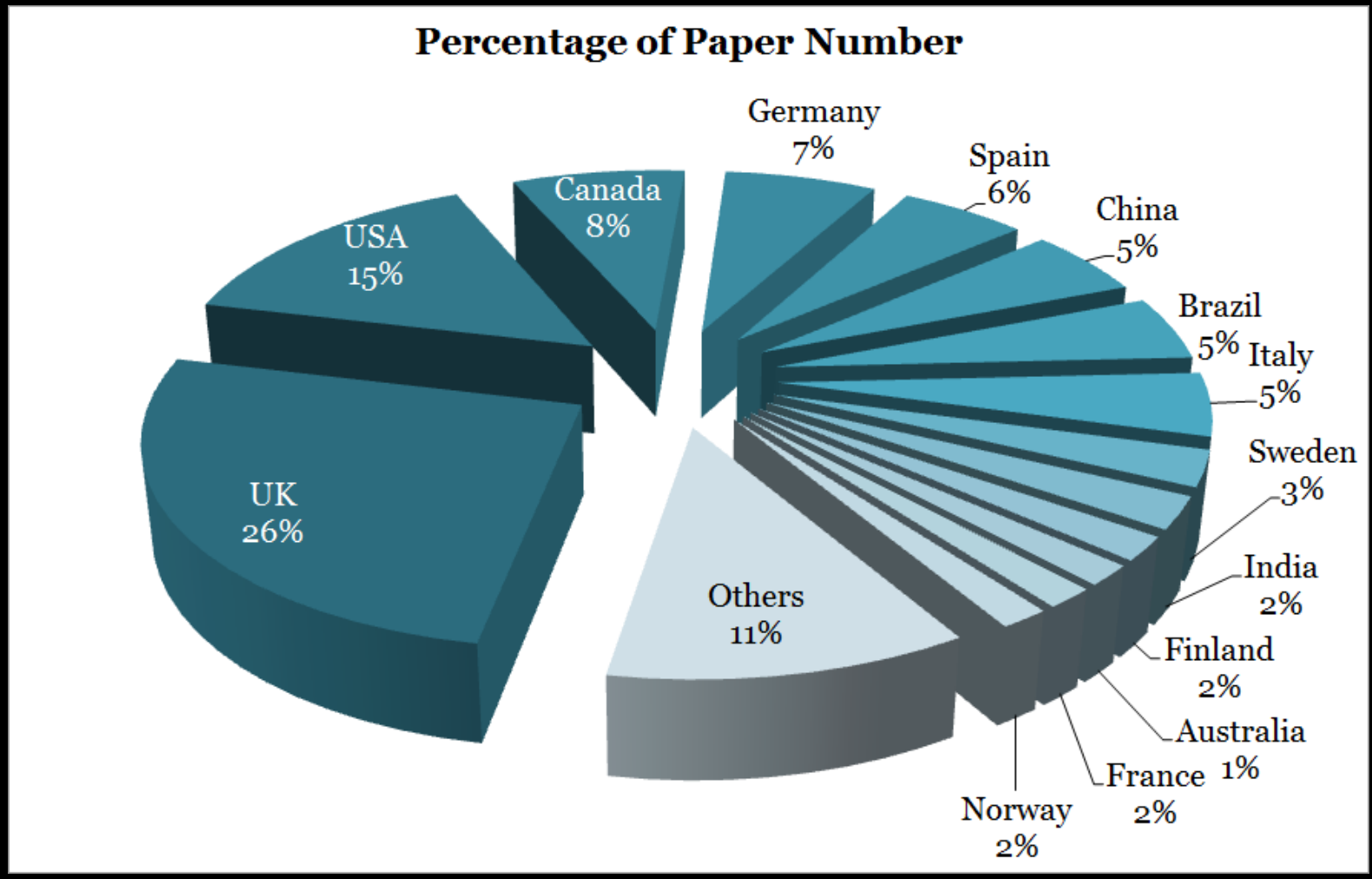


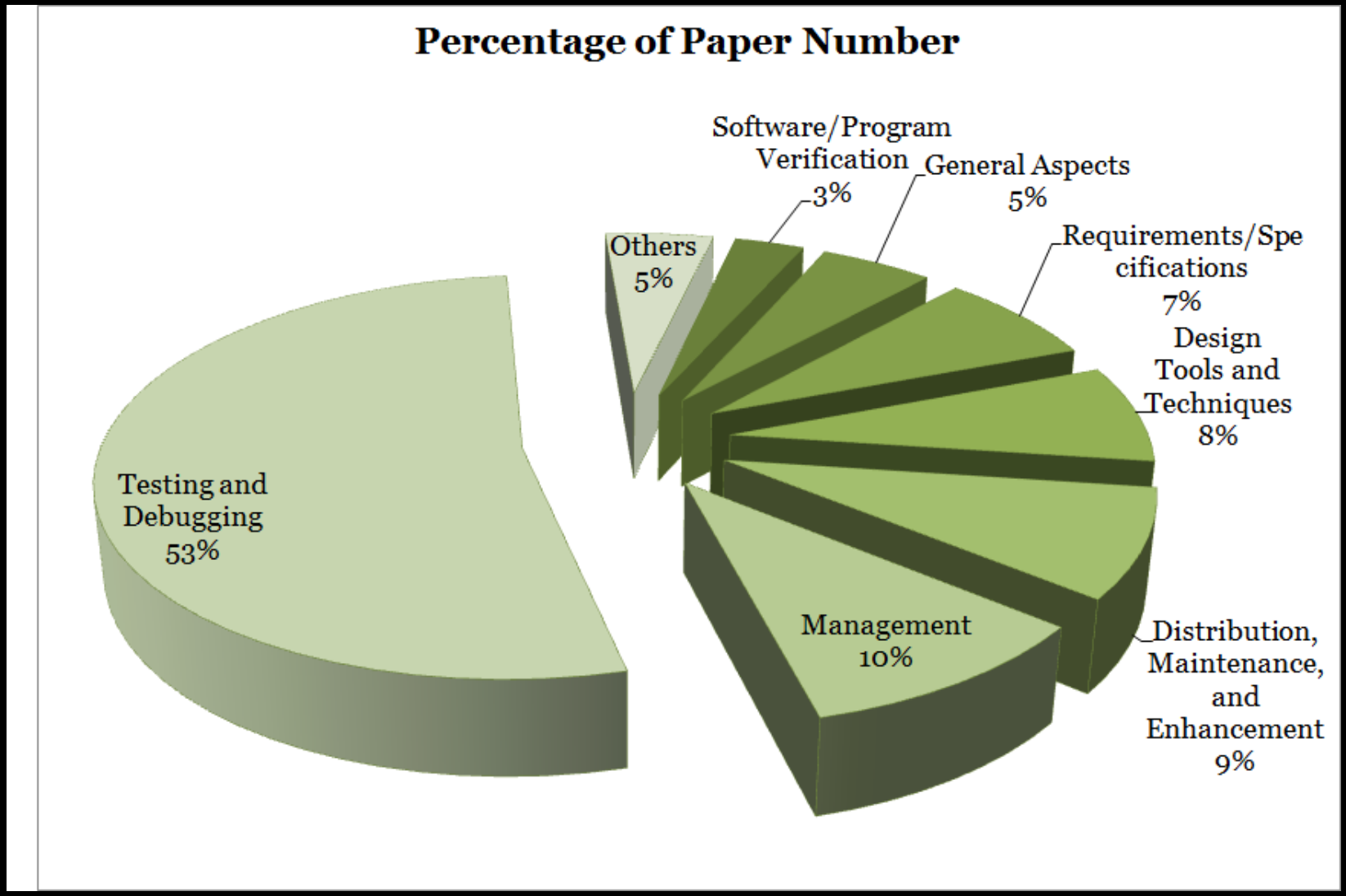
normalised branch distance between 0 and 1
indicates how close approach level is to being penetrated

Structural Testing

Evolutionary test environment for automatic structural testing

Wegener J, Baresel A, Sthamer H. *Information and Software Technology* 2001


Percentage of Paper Number
Germany 7%
Spain 6%
China 5%
Brazil 5%
Italy 5%
Sweden 3%
India 2%
Finland 2%
Australia 1%
France 2%
Norway 2%
Others 11%
UK 26%
USA 15%
Canada 8%


Percentage of Paper Number
Software/Program Verification 3%
General Aspects 5%
Requirements/Specifications 7%
Design Tools and Techniques 8%
Distribution, Maintenance, and Enhancement 9%
Management 10%
Testing and Debugging 53%
Others 5%

Optimising Multiple Objectives

Multi Objective Search

Many problems have multiple objectives

Often we have many metrics

Recent SBSE work has been multi objective

Pareto optimality can yield insight

Multi Objective Search

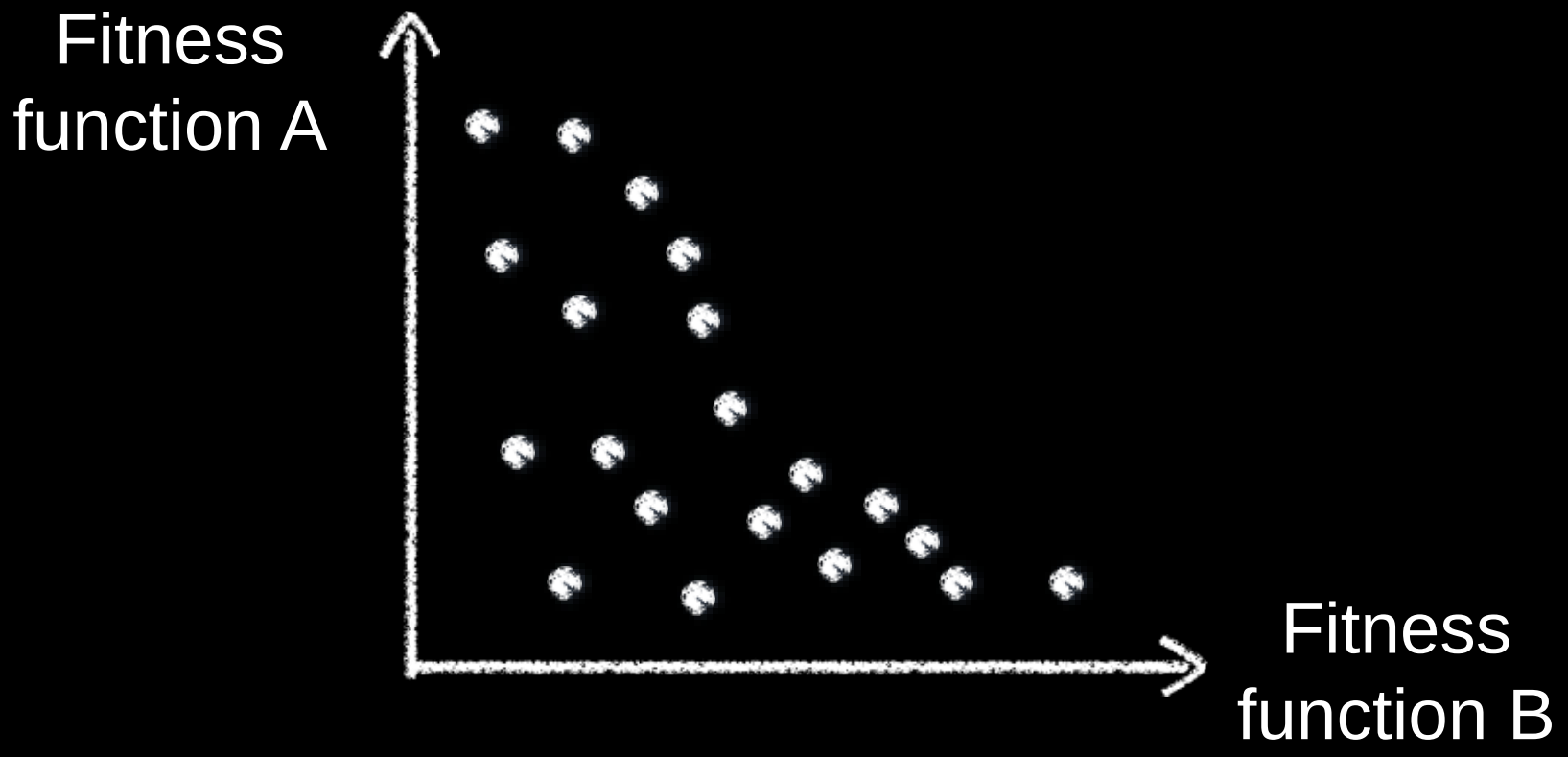
Many problems have multiple objectives

Often we have many metrics

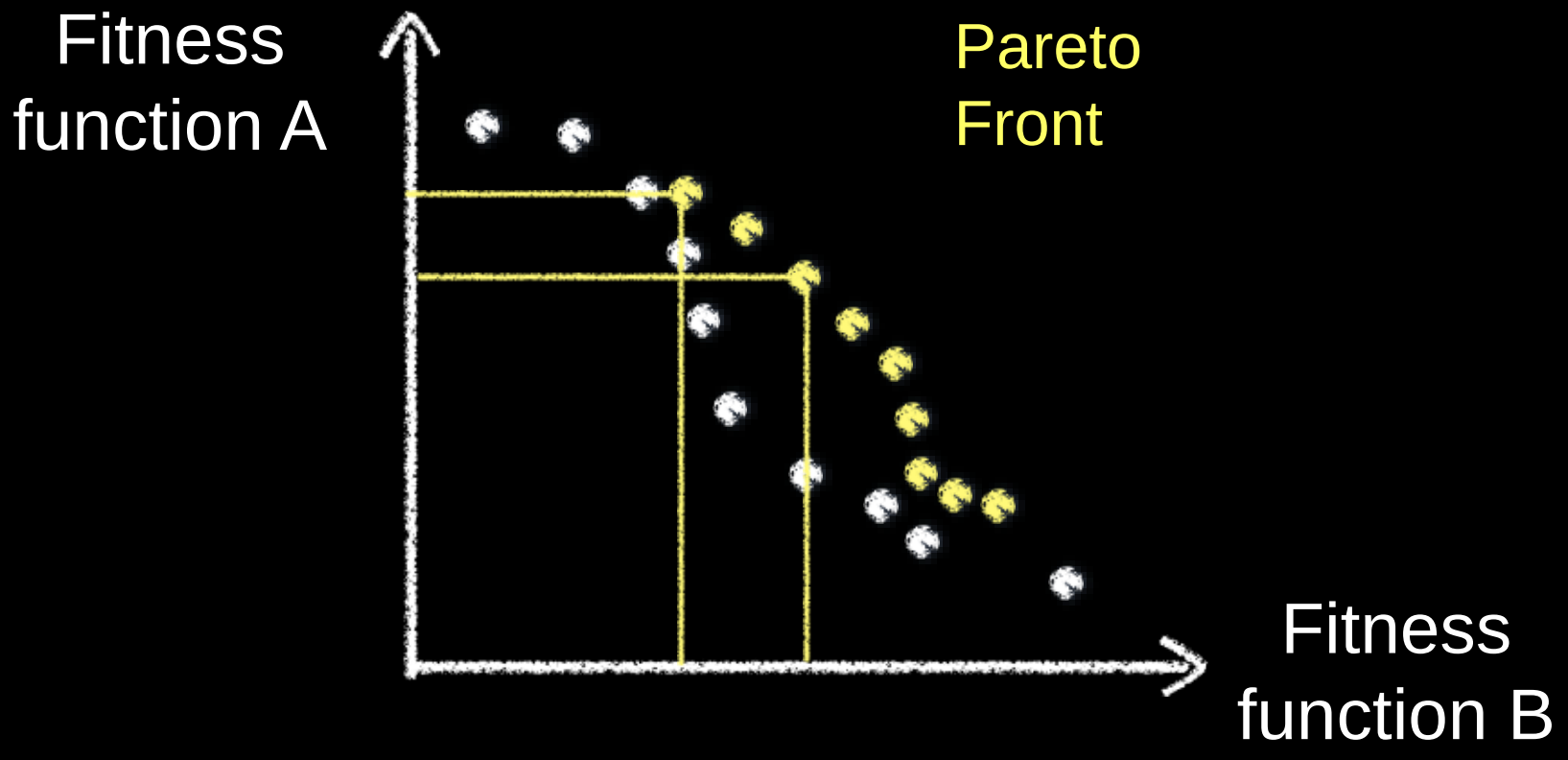
Recent SBSE work has been multi objective

Pareto optimality can yield insight

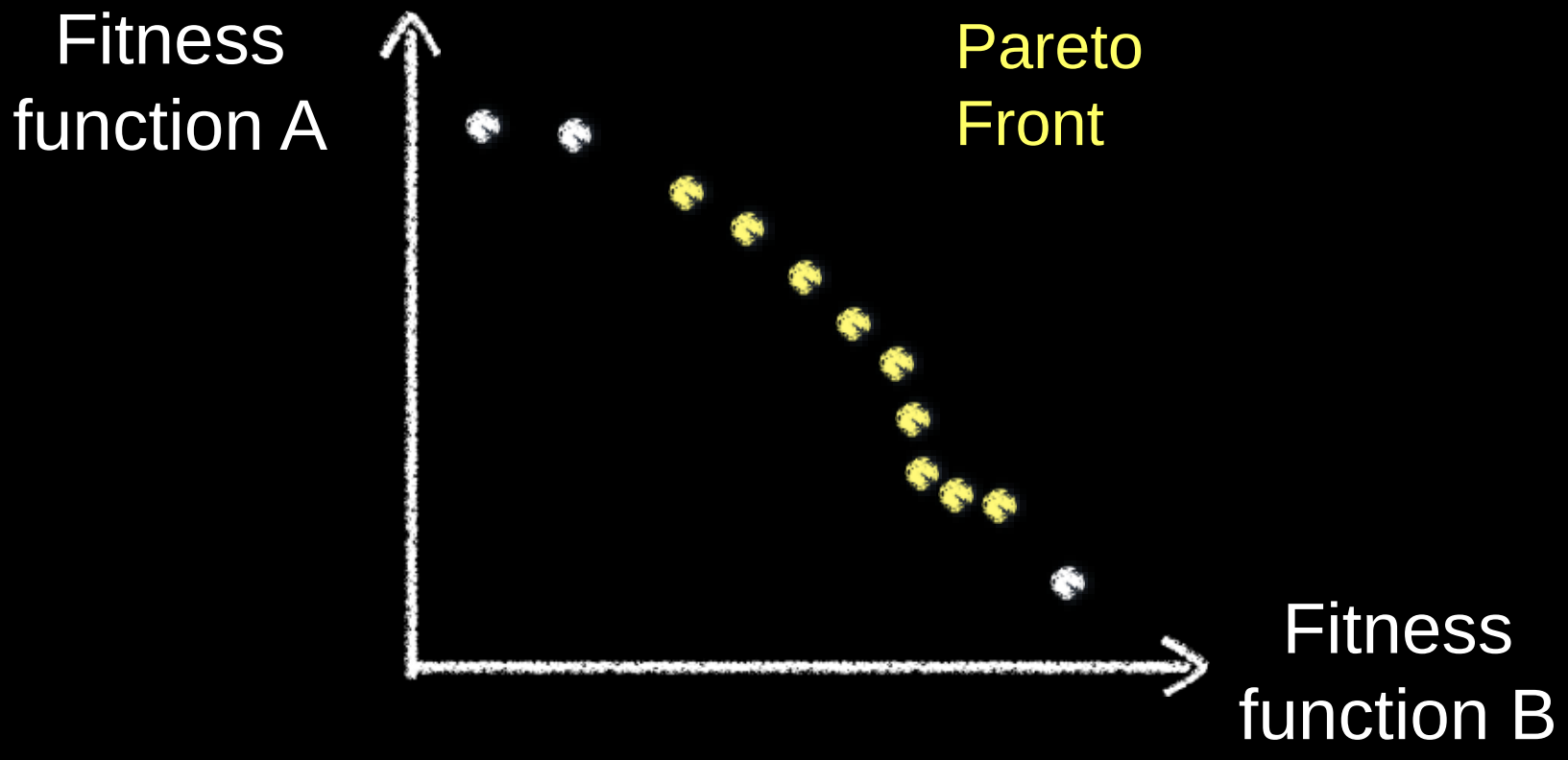
Multi Objective Search



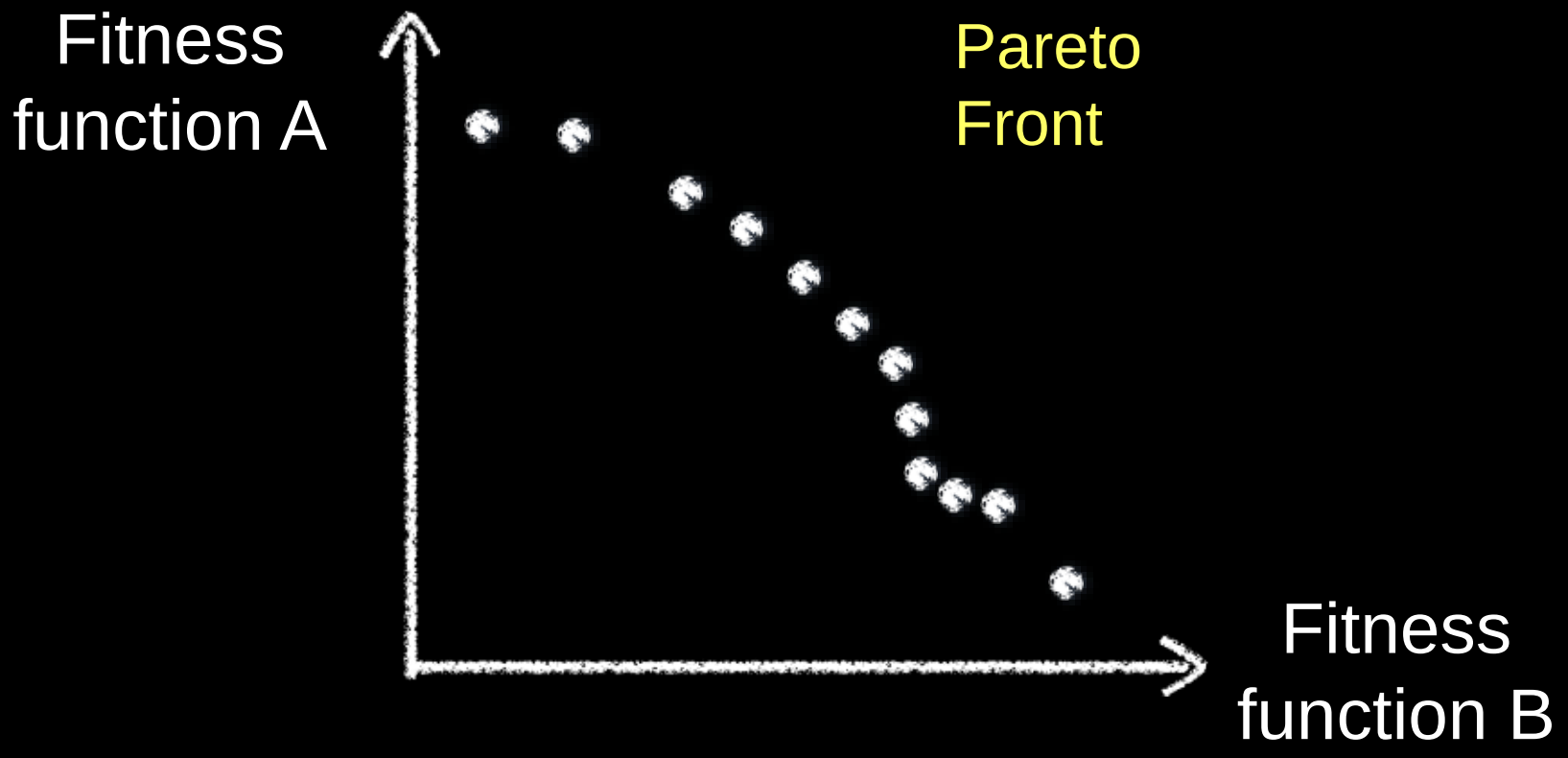
Multi Objective Search



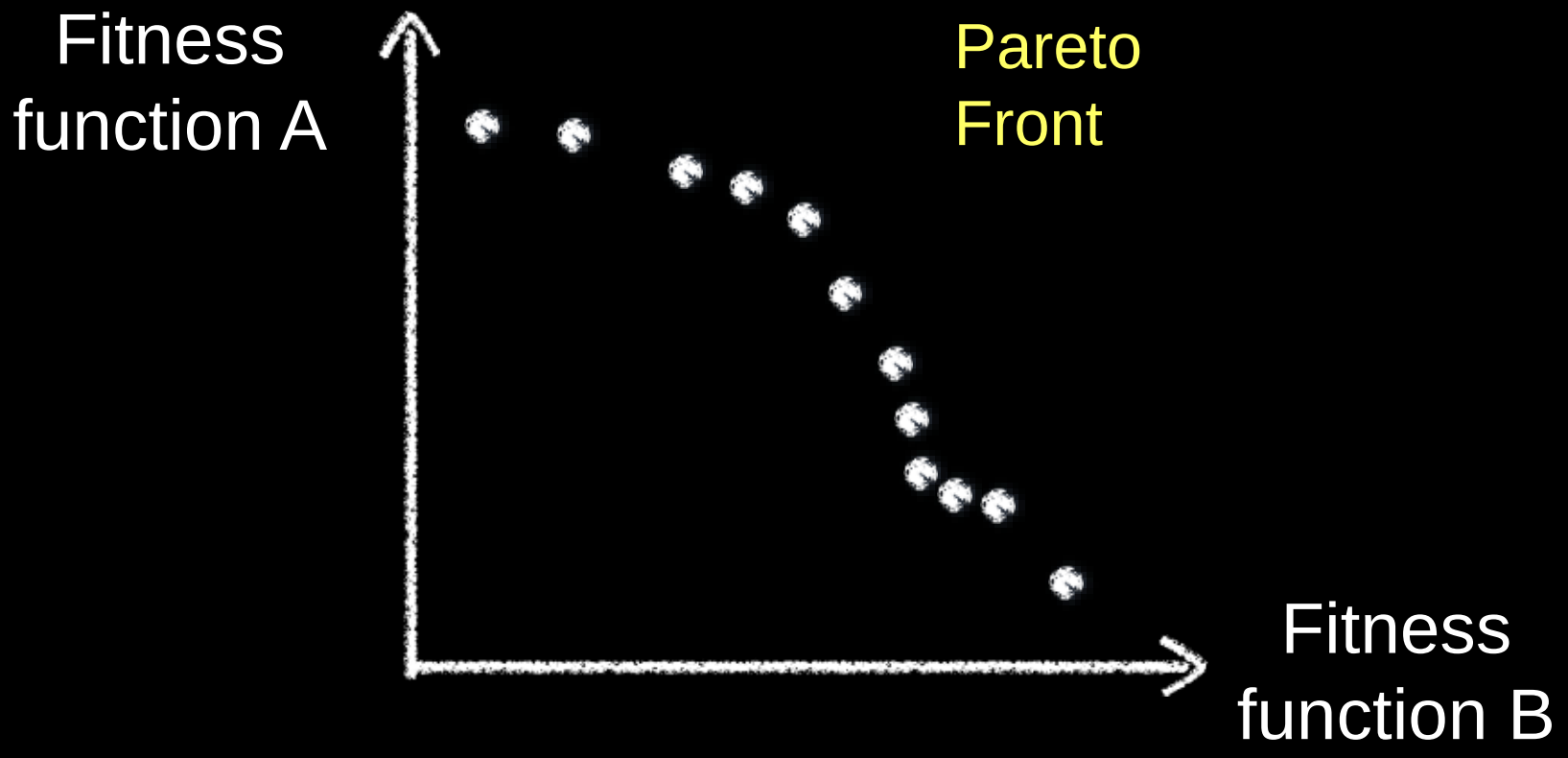
Multi Objective Search



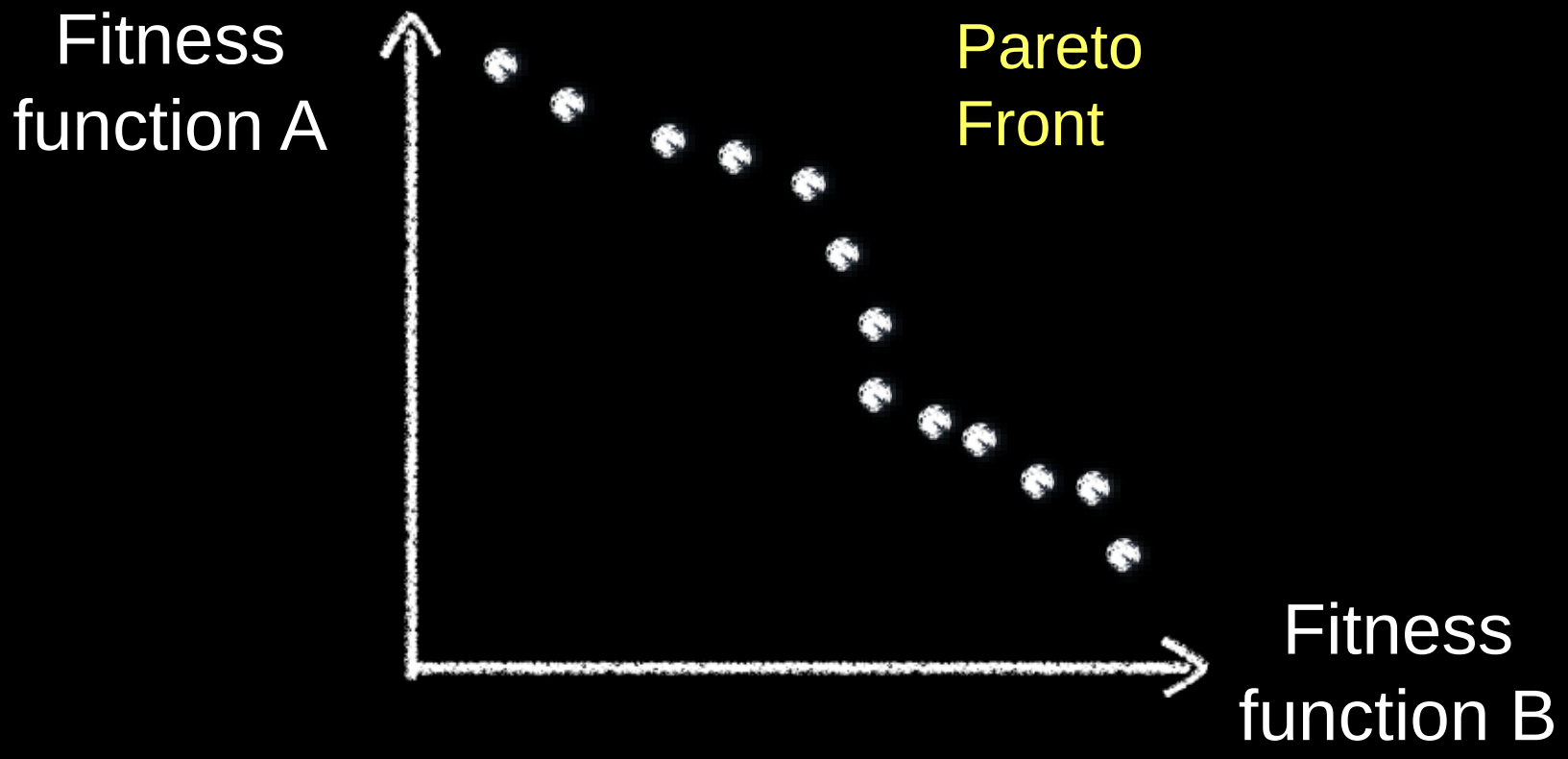
Multi Objective Search



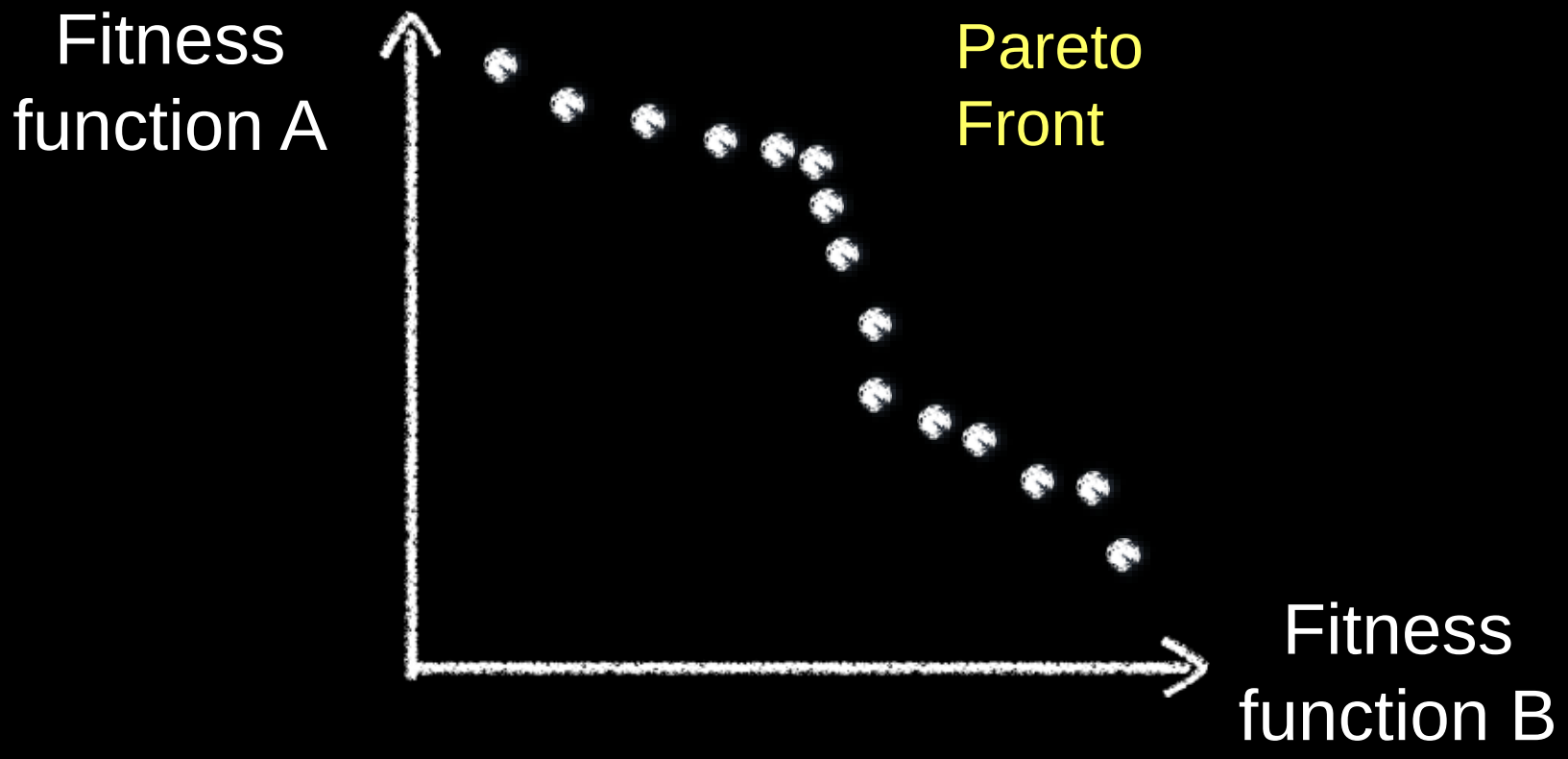
Multi Objective Search



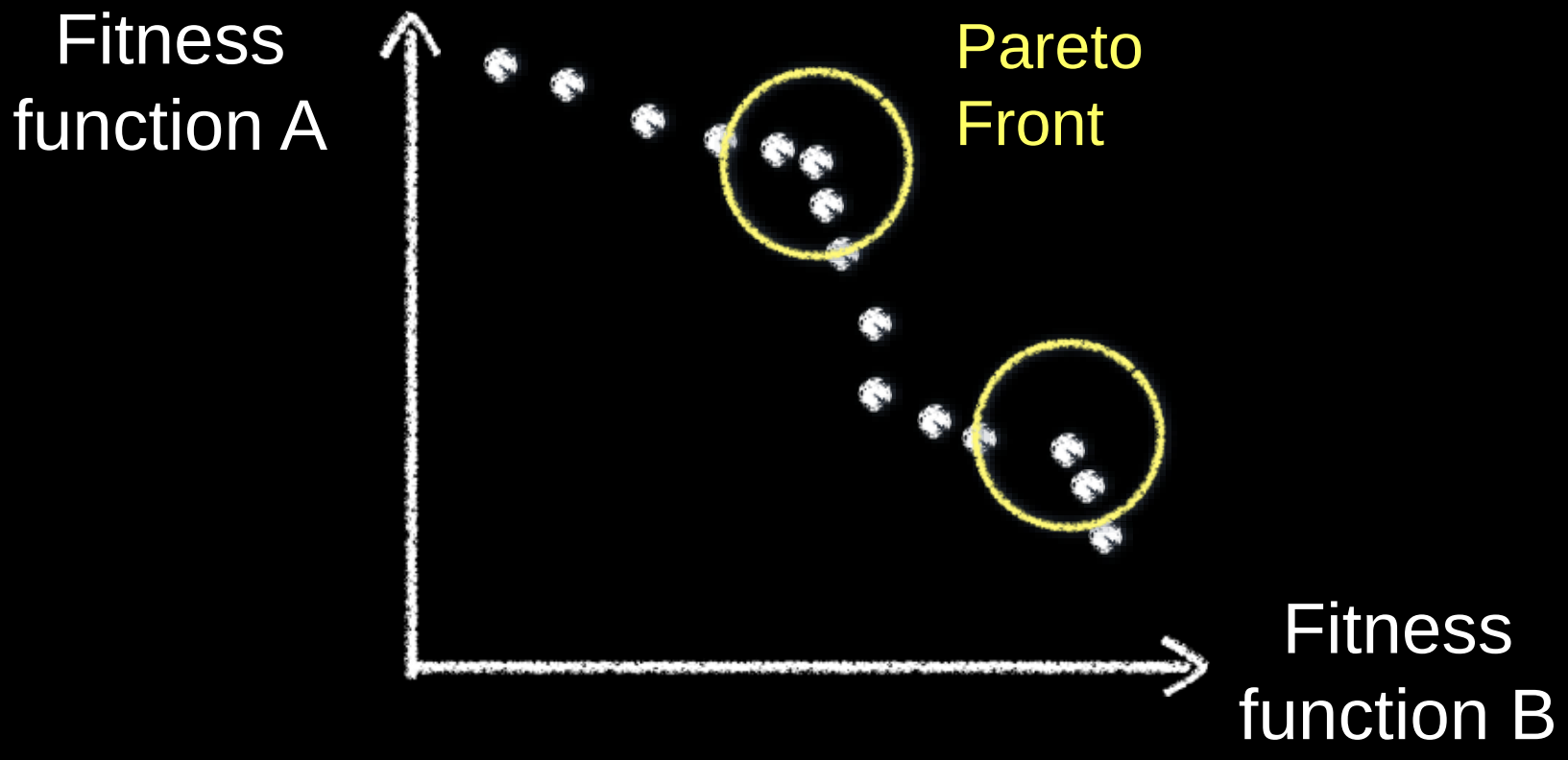
Multi Objective Search



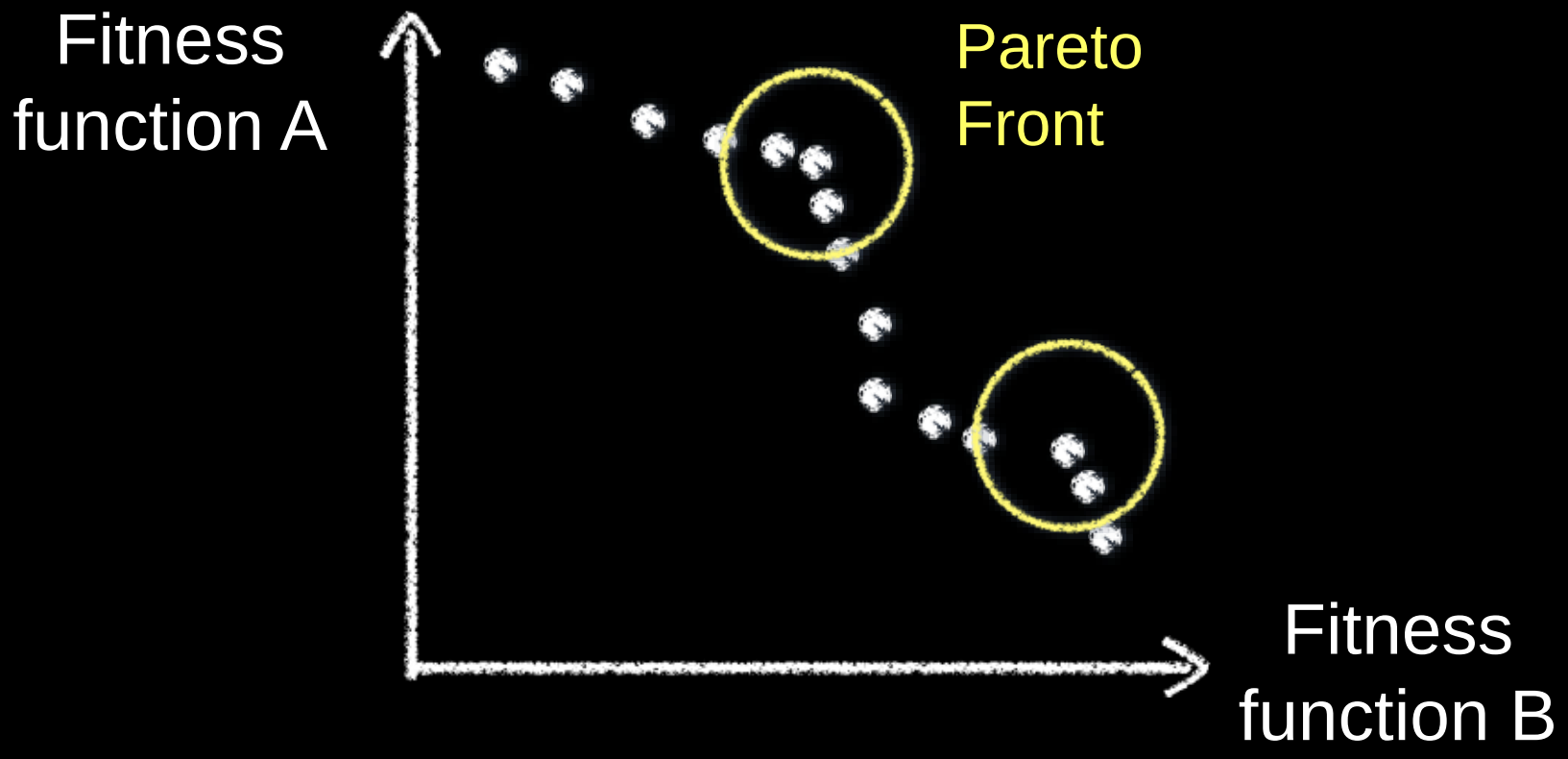
Multi Objective Search



Multi Objective Search



Direction of front growth



SBSE International

☺ SSBSE:
Symposium on Search-Based Software Engineering
http://ssbse.org/

☺ CSBSE
Chinese Search-based Software Engineering
http://csbse.org

☺ NasBASE
North American Search Based Software Engineering Symposium
http://nasbase.org

csbse.org

中国基于搜索的软件工程

Chinese Search-based Software Engineering

Workshops

- The Fourth Chinese SBSE Workshop, CSBSE 2015, NJU, NanJing, 2015
- The Third Chinese SBSE Workshop, CSBSE 2014, CUMT, Xuzhou, 2014
- The Second Chinese SBSE Workshop, CSBSE 2013, DLUT, Dalian, 2013
- The First Chinese SBSE Workshop, CSBSE 2012, BUCT, Beijing, 2012

Resources

- SBSE Repository
- Genetic Programming Bibliography

SBSE Surveys

- 李征,巩敦卫,聂长海,江贺,《基于搜索的软件工程研究进展与趋势》, 2013-2014,中国计算机科学技术年度报告, 2014,
- Mark Harman, Afshin Mansouri and Yuanyuan Zhang. **Search Based Software Engineering: Trends, Techniques and Applications** ACM Computing Surveys. 45(1): Article 11, 2012.
- Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: **A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation**. IEEE Transactions on Software Engineering, 36(6):742-762, 2010
- Outi Räihä: **A Survey on Search-Based Software Design**. Computer Science Review, Volume 4, Issue 4, November 2010, Pages 203-249
- Wasif Afzal, Richard Torkar and Robert Feldt: **A Systematic Review of Search-based Testing for Non-Functional System Properties**. Information and Software Technology, 51(6):957-976, 2009
- Mark Harman: **The Current State and Future of Search Based Software Engineering**. ICSE Future of Software Engineering: 342-357, 2007
- Phil McMinn: **Search-based software test data generation: a survey**. Software Testing, Verification and Reliability 14(2): 105-156, 2004

SBSE Tools

- [Austin] 针对C程序的，基于搜索的软件测试数据生成工具。
- [Cocotest] 利用搜索的方法对连续控制器(continuous controllers)进行自动化的MIL（Model-In-the-Loop）测试的工具。
- [PEX] 微软针对.NET的白盒测试开源框架。结合了符号执行与搜索技术。可以作为Visual Studio的插件，自动生成高覆盖率的测试数据。
- [GenProg] 利用遗传编程技术对C语言程序进行故障自动修复的工具。
- [EvoSuite] 针对java程序的测试数据生成工具。
- [JavaPathFinder] 对java程序进行执行、检验的工具，可用于探测多线程程序所有可能的执行路径。
- [Milu] 基于搜索的高阶变异测试(Higher Order Mutation Testing)工具。
- [FlagRemover] 用于替换程序中的一些控制变量，使搜索算法的运用能够达到更好的效果。

基于搜索的软件工程研究进展与趋势

2013-2014

中国计算机科学技术

发展报告

http://www.ccf.org.cn/sites/ccf/contndbg.jsp?contentId=2832949311672


Dalian Liaoning China
CSBSE2013
Chinese Search Based Software Engineering Workshop | Dalian, China | 2013
The First Chinese SBSE Workshop
Beijing
Dalian
Xuzhou


Beijing
Dalian
Nan Jing

CSBSE 2015

- ☺ **4th Chinese Search Based Software Engineering Workshop (CSBSE) 2015**
- ☺ 第四届中国基于搜索的软件工程研讨会，2015 年 6 月 13 日，南京

Keynote： **The Future of Search-Based Testing**

Phil McMinn

基于搜索的软件工程

软件学报专刊

软件学报 Journal of Software

基于搜索的软件工　研究　征文通知

【专刊题目】

基于搜索的
软件工程研究

【特约编辑】

李 征
北京化工大学

巩敦卫
中国矿业大学

聂长海
南京大学

江 贺
大连理工大学

【出版时间】

2016 年第 4 期

基于搜索的软件工程（SBSE），是软件工程学科发展的新方向。随着软件规模逐渐庞大与复杂，传统的从问题空间构造解决问题的方法已经变得越来越困难。基于搜索的软件工程，从问题的解空间出发，将传统的软件工程问题转化为优化问题，并使用高性能的搜索方法，在问题所有可能解的空间中，寻找最优解或者近似最优解，被 ICSE&2007 确立为软件工程领域发展的新方向。到目前为止，已经在软件需求分析、测试数据自动生成、程序错误自动修复等方面，取得了显著的研究成果，有效地促进了软件工程学科的发展和应用。&

为及时反映我国在本领域研究进展，《软件学报》将出版“基于搜索的软件工程研究”专刊，收录该领域近期取得的原创性高水平研究成果，进一步促进该领域的发展。本次专刊选题将突出以下几个研究热点：面向软件工程的搜索方法、面向软件工程的知识学习、基于搜索的方法在软件工程生命周期的应用，以及基于搜索的软件工程在工业界的应用等。&

专刊将与第四届中国基于搜索的软件工程研讨会（CSBSE&2015）和全国软件与应用学术会议（NASAC&2015）合作。CSBSE&2015 评审的优秀论文将优先推荐到专刊评审，同时向国内相关研究领域专家学者和科研人员广泛征文。收录论文经过专家评审后，将在 NASAC&2015 上报告，并根据论文修改情况和会议报告情况终审确定是否录用。欢迎踊跃投稿。&

一、征文范围（包括但不限于以下主题）！

!!1.!面向软件工程的搜索方法**!**

（1）面向软件工程的单目标优化算法&
&&&&&遗传算法、模拟退火、粒子群算法等&
（2）面向软件工程的多目标优化算法&&
（3）面向软件工程的搜索空间变换方法&

!!2.!面向软件工程的知识学习**!**

（1）面向软件工程的深度学习&
（2）面向软件工程的知识推理&
（3）软件工程仓库挖掘&

3.!基于搜索的方法在软件工程生命周期的应用**!**

（1）基于搜索的需求工程&
（2）基于搜索的软件测试&
（3）基于搜索的软件维护&
（4）基于搜索的软件设计&
（5）基于搜索的软件产品线&
（6）基于搜索的软件项目管理&

4.!基于搜索的软件工程在工业界的应用**!**

（1）基于搜索的自动化测试&
（2）基于搜索的测试数据生成应用&

二、重要日期!

第一轮截稿日期：2015 年 6 月 30 日，第一轮预录用通知发出日期：2015 年 7 月 31 日&
第二轮截稿日期：2015 年 8 月 31 日，第二轮预录用通知发出日期：2015 年 10 月 15 日&
NASAC&2015 报告日期：2015 年 11 月 6 日&–&8 日（武汉）&
修改稿提交日期：2015 年 11 月 31 日&
终审结果发出日期：2015 年 12 月 15 日&
最终稿提交日期：2015 年 12 月 31 日&

三、征文要求&

1.& 投稿方式：采用“软件学报在线投稿系统”(http://www.jos.org.cn)投稿。投稿时请选择稿件型为“专刊投稿”，同时在中文标题后面加上括号，在括号内注明专刊名称，即：(基于搜索的软件工程研究专刊)字样。&
2.& 稿件格式：参照《软件学报》论文模板给定的格式（见学报网站“下载区”）。&
3.& 投稿论文未在正式出版物上发表过，也不在其他刊物或会议的审稿过程中，不存在一稿多投现象；保证投稿论文的合法性（无抄袭、剽窃、侵权等不良行为）。&
4.& 其他事项请参阅投稿指南：http://www.jos.org.cn/ch/reader/view_fixed_content.aspx?id=instructions&
5.& 投稿作者需提交投稿声明；专刊投稿论文不收审理费。录用刊发论文收取软件学报标准版面费。发表之后，将按软件学报标准支付稿酬，并赠送样刊。&
6.& 通过初审的预录用论文，需在 NASAC&2015 会议（http://grid.hust.edu.cn/nasac2015/）上作报告，根据论文修改情况和会议报告情况终审确定是否录用。&

CCF TCSE 软件工程专业委员会

CSBSE
csbse.org

Email: lizheng@mail.buct.edu.cn
Web: http://cist.buct.edu.cn/staff/zheng/