Genetic Improvement

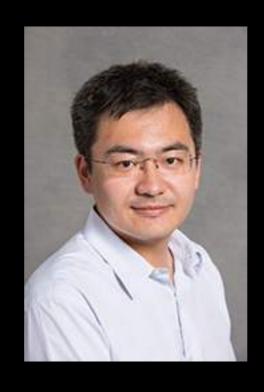
Justyna Petke

Centre for Research in Evolution, Search and Testing University College London



Thank you







Mark Harman

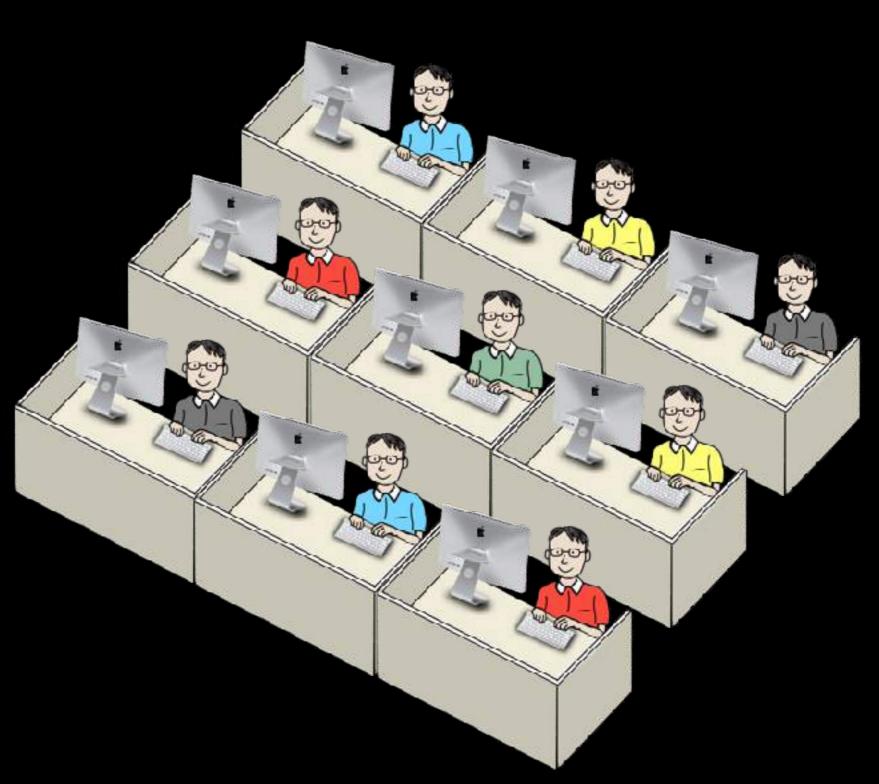
Yue Jia

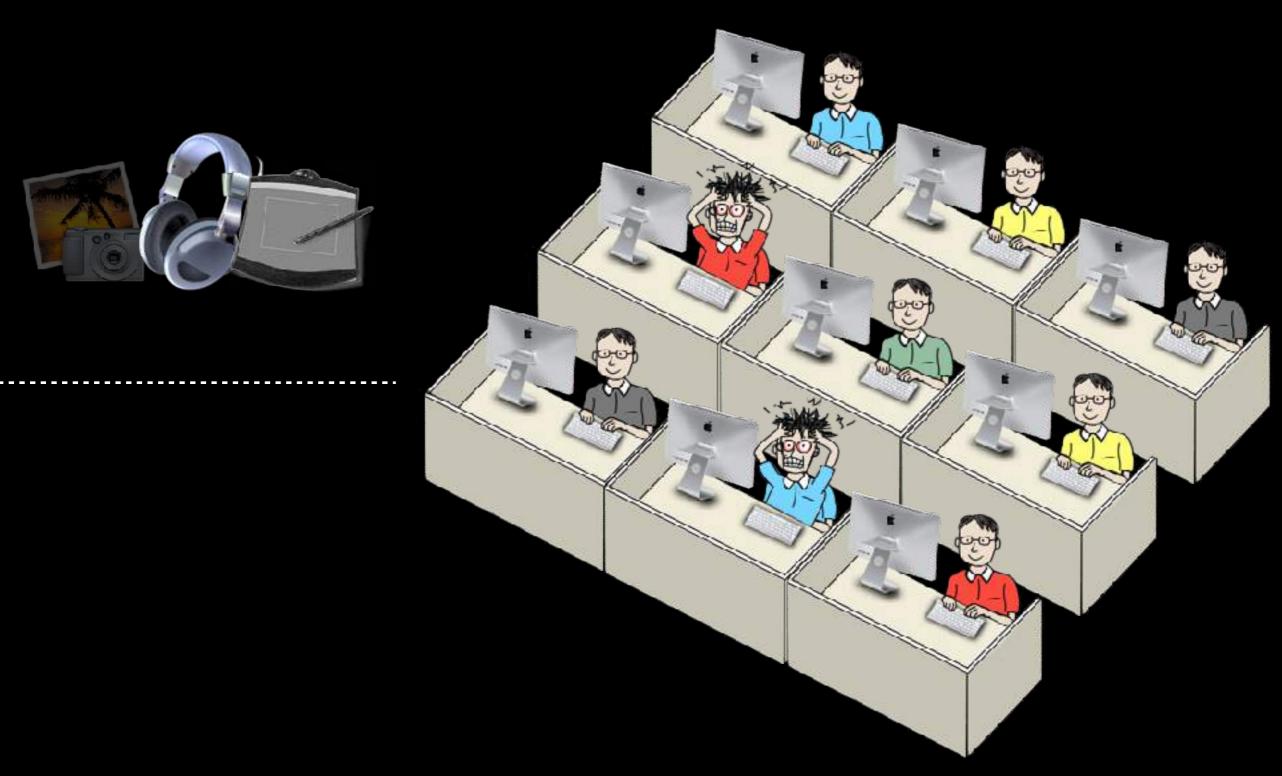
Alexandru Marginean

Motivation

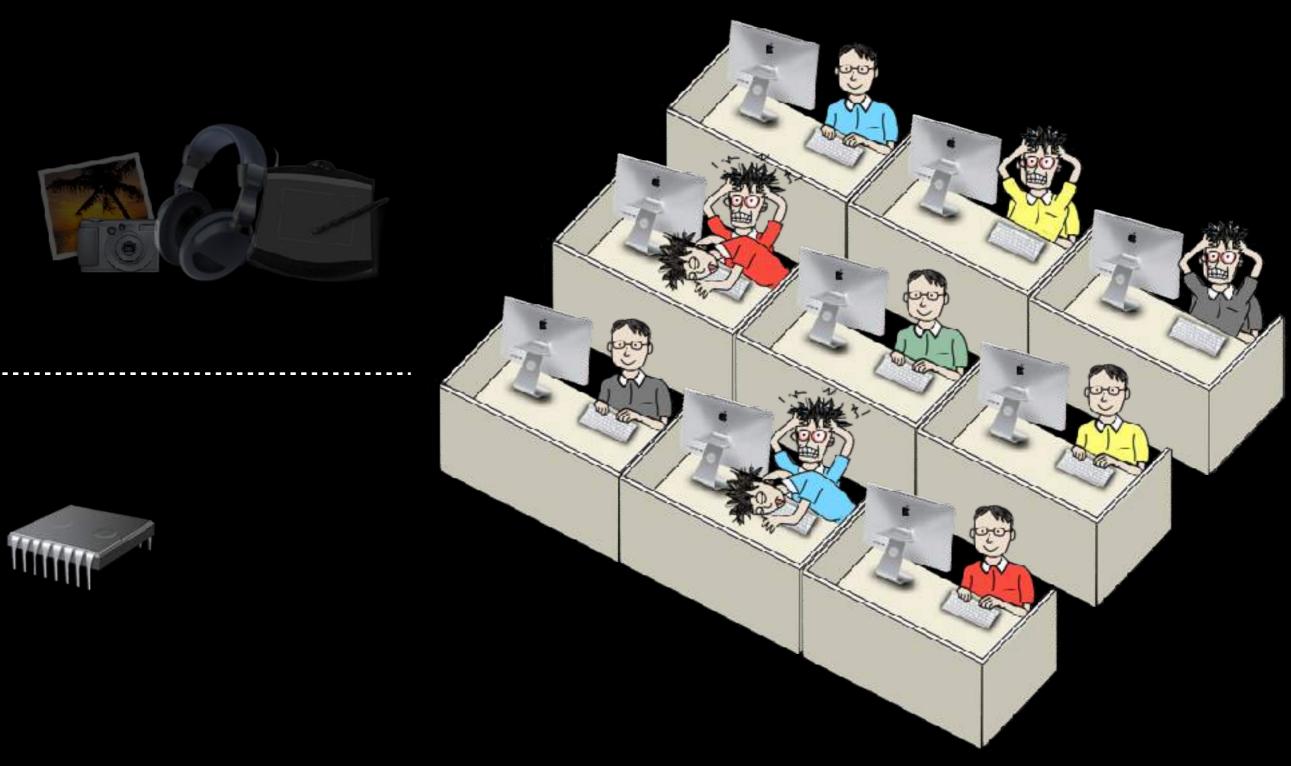


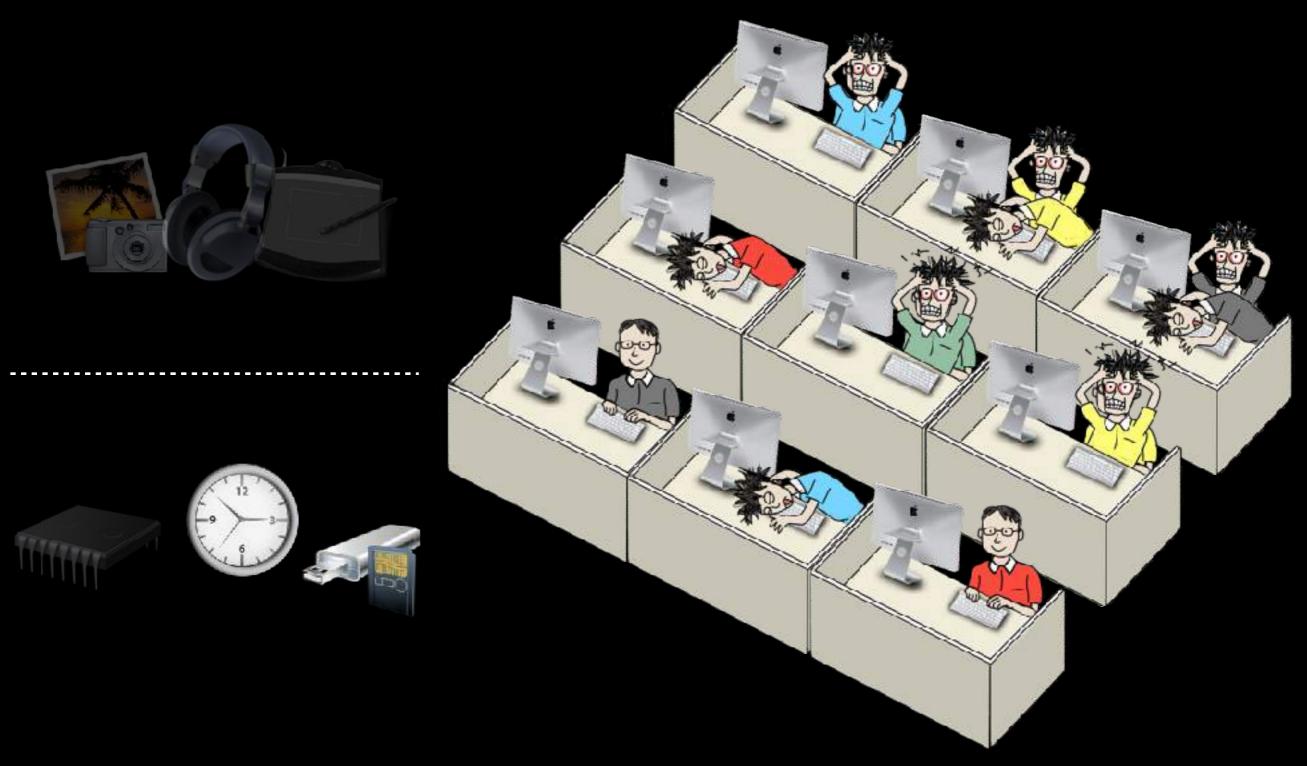


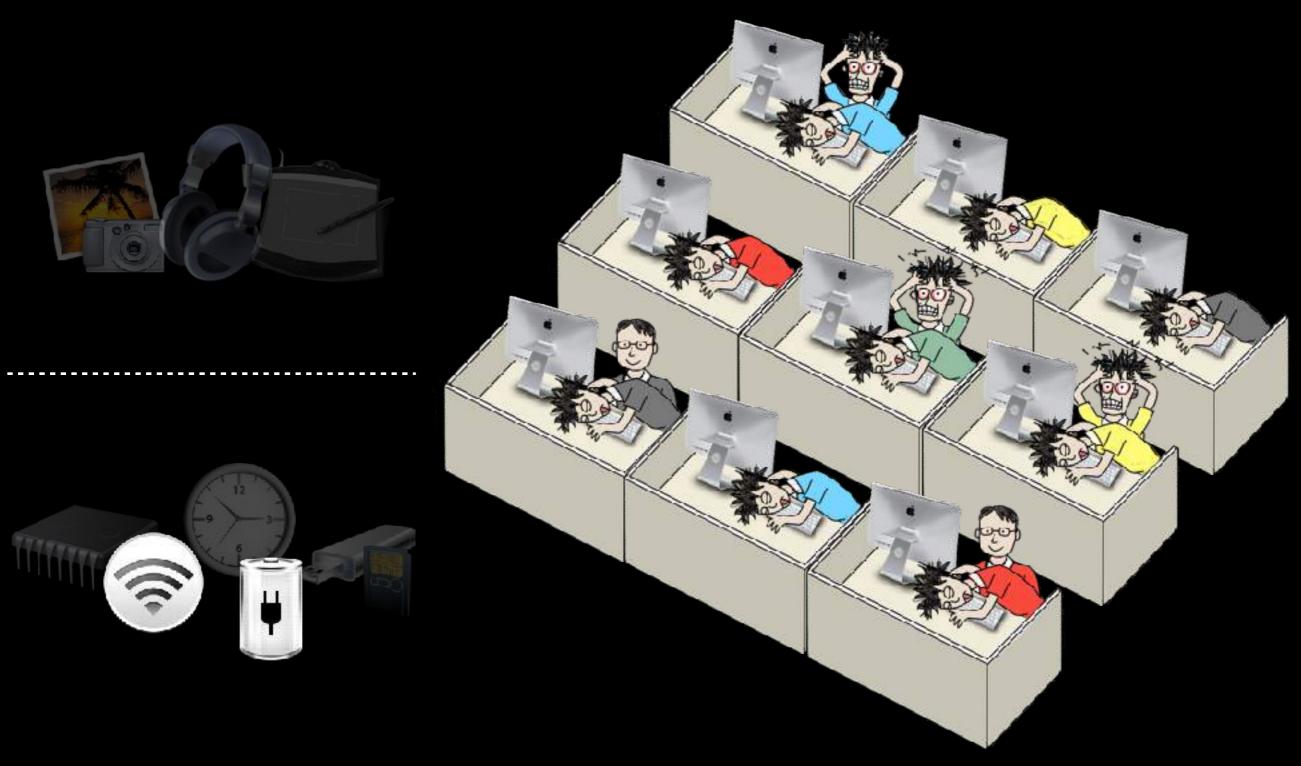




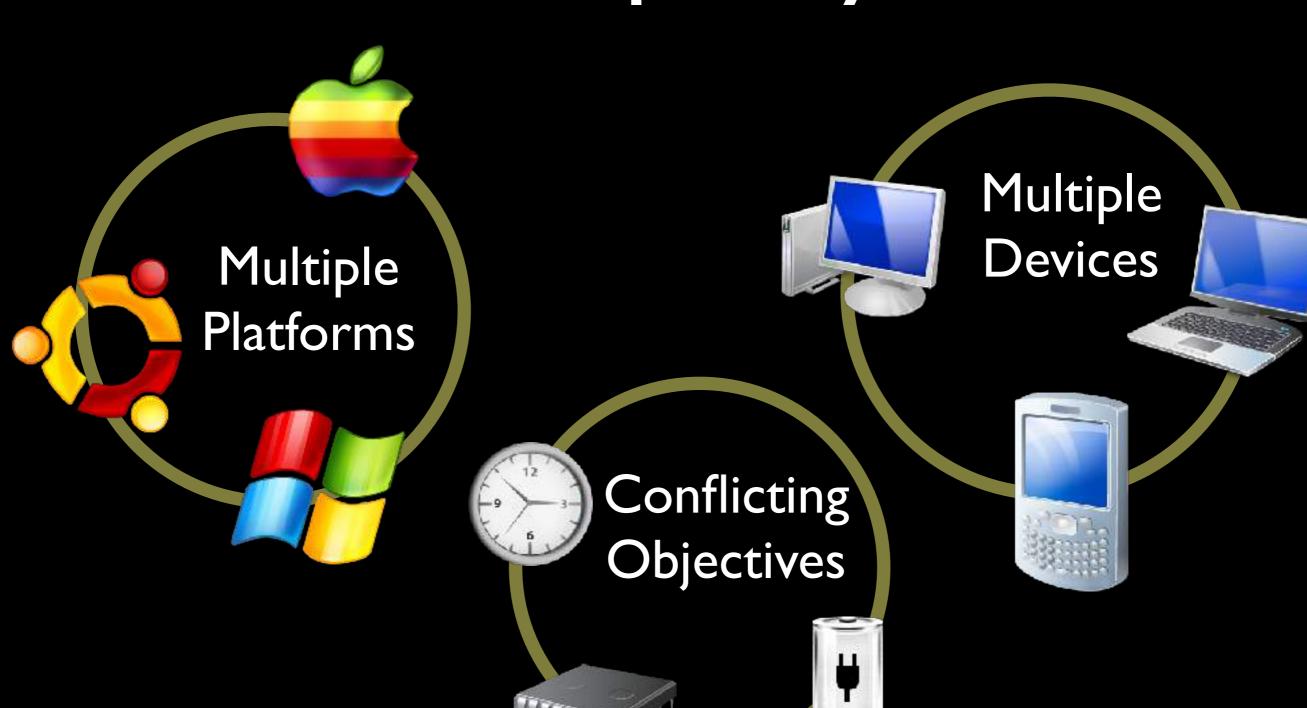








Multiplicity





Programming is Hard!

y). *!metion(a)("ase strict"; function b(b) (return this each (function()); he[b]())))var c=function(o)(this.element=a(b));c.version= 3.3.7 ,c.TRANSITION_DURATION=150,c.prot w-meno("),c-b.fets("target");df(d)[(d-b.ettr("href"),d=d&&d.replace(/."(?=#[^\s]"5)/,"")),! st a"), f-a. Frent("nice. bs. teb", {relutedTarget:b[0]}), g-a. Event("show. bs. teb", {relatedTarget:e[0] indefrounted())[var h-a(d);this.activete(b.closest("Ii"),c),this.activete(h,h.parent(),function) rigger((type:"shown, bs. tab", related(arget:o[8]))))))), c.prototype.activate=function(b, d, e){func .p.active").resoveClass("active").end().find("[data-toggle="tah"]").attr("ario-expanded",!1),| ta-equated, ie), b)(s(e).offsetbildth, b. addClass("in")):b. resoveClass("fade"), b. parent(".dropdox ().find('[data-toggle-"tab"]')_attr("aria-expanded", [0), e&&e()]var g=d.find("> .active").h=c&& w")[[Hid.Hid](").fase").length);g.length58h2g.one("bsTronsitionEnd",f).envlateTransitionEnd pur d-a.fa.tab;a.fa.tab-b,a.fa.tab.Constructor-c,a.fn.tab.noConflict=function(){return a.fn.t 'shor')];a(document).oo("click.bs.tab.datn-api", "[data-toggle="tab"]",e).on("click.bs.tab.data se strict ; function b(b) (return this.each(function() (war des(this), e=d.data("bs.effix"), f="ob. ripped base(b)()));war c-function(b,d)(this.options=a.extend({}},c.DEFAULTS,d),this.\$target=a ",a.proxy(this.checkPosition,this)).on("click.bs.affix.deta-api",a.proxy(this.checkPositionWi m23, this.phreedoffset-oull, this.checkPosition());c.VERSION="3.3.7",c.RESET="affix affix-top State-faction(a,b,c,d)(var e-this.\$target.scrollTop(),f=this.\$clement.offset(),g=this.\$targ "bottom" == the affixed) return nell/=c?!(e+this.unpinc-f.top)&& "bottom":!(e+gc-a-d)&& "bottom" reservance and a star a training star get scroll Top(), b this felement offset(); return ethEventLogs function()(setTimeout(e.proxy(this.checkPosition, this) 100



Automated Programming

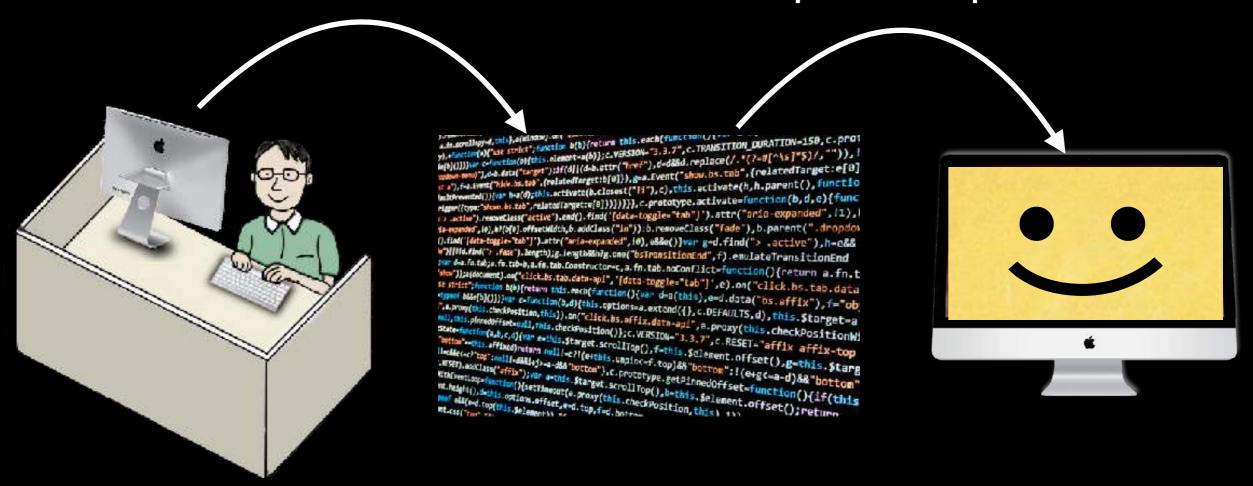
y). *faction(a)("ast strict"; function b(b) {return this.each(function()); ** le[b](1)))var c=function(o)(this, element=a(b));c.version= 3.3.7°,c.TRANSITION_DURATION=150,c.prot person"), d-a. fats("target"); 3f(d)|(d-b.ettr("firef"), d=d&&d.replace(/."(?=#[^\s]"\$)/, "")), [st a"), f-a. Frent("nice.bs. teb", (relutedTarget:b[0])), g-a. Event("show.bs. teb", (relatedTarget:e[0]) leabfrownted())[var h-a(d);this.activate(b.closest("Ii"),c),this.activate(h,h.parent(),function) rigger((type:"show, bs. tab", related(argette[8]))))))), c.prototype.activate=function(b, d, e){func ... active").resoveClass("active").end().find("[data-toggle="tah"]").attr("ario-expanded", !1),| ta-equaded, ie), b)(s(e).offsethildth, b. addClass("in")):b. removeClass("fade"), b. parent(".dropdox ().find([data-toggle="tab"]")_attr("aria-expanded", i0), e&2e())var g=d.find("> .active"), h=e&& w"\\\\(\text{ind(").fase"\).length);g.length&&in2g.cne("bsTransitionEnd",f).envlateTransitionEnd pur d-a.fa.tab;a.fa.tab-b,a.fa.tab.Constructor-c,a.fn.tab.noConflict=function(){return a.fn.t 'show"));a(document).co("click.bs.tab.data-api", "[data-toggle="tab"]",e).on("click.bs.tab.data se strict"; function b(b) (return this.each(function() (war des(this), e=d.data("bs.offix"), f="ob. ripped base(b)()));war c-function(b,d)(this.options=a.extend({}},c.DEFAULTS,d),this.\$target=a ",a.proxy(this.checkPosition,this)).on("click.bs.affix.deta-api",a.proxy(this.checkPositionWi m23, this.phreedoffset-oull, this.checkPosition());c.VERSION="3.3.7",c.RESET="affix affix-top State-faction(a,b,c,d)(van e-this.\$target.scrollTop(),f-this.\$clement.offset(),g-this.\$targ "bettom"-=0ds.affixed)return nell/=c?!(e+this.umpinc=f.top)&&"bottom":!(e+gc=a-d)&&"bottom" Hedde(+c)*top::malli-daki+j>=-dda*bottom*).c.prototype.getPinnedOffset=function(){if(this Leser).assrlass("effia"); war a=tnds_\$ta~get.scrollTop(),b=this.\$element.offset(); return #ithSweetLoop=function()(setFinesur(e.proxy(this.checkPosition,this) 15)



Genetic Improvement of Software

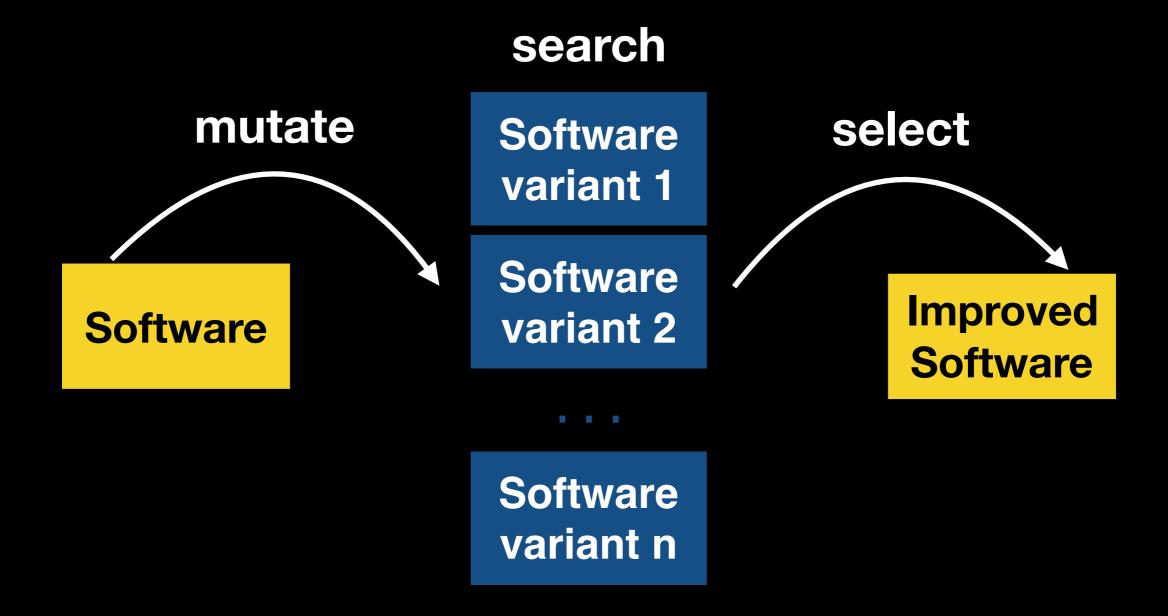
human writes code

computer improves it



Genetic Improvement

uses search in order to improve existing software



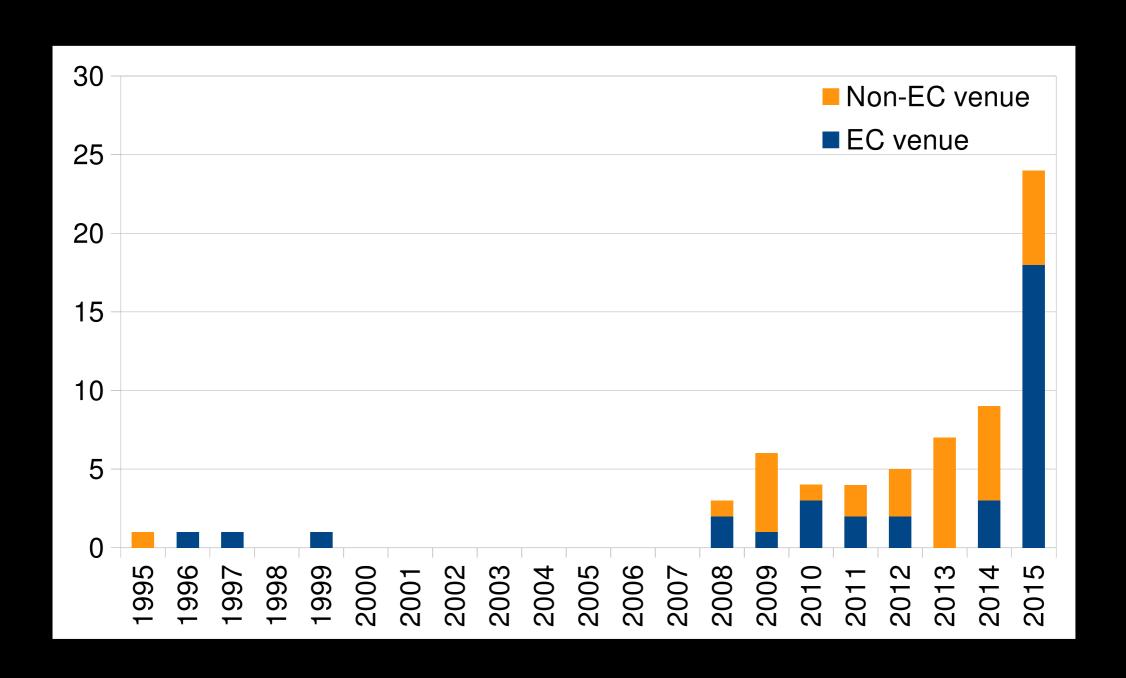


GI History



Genetic Improvement of Software: A Comprehensive Survey

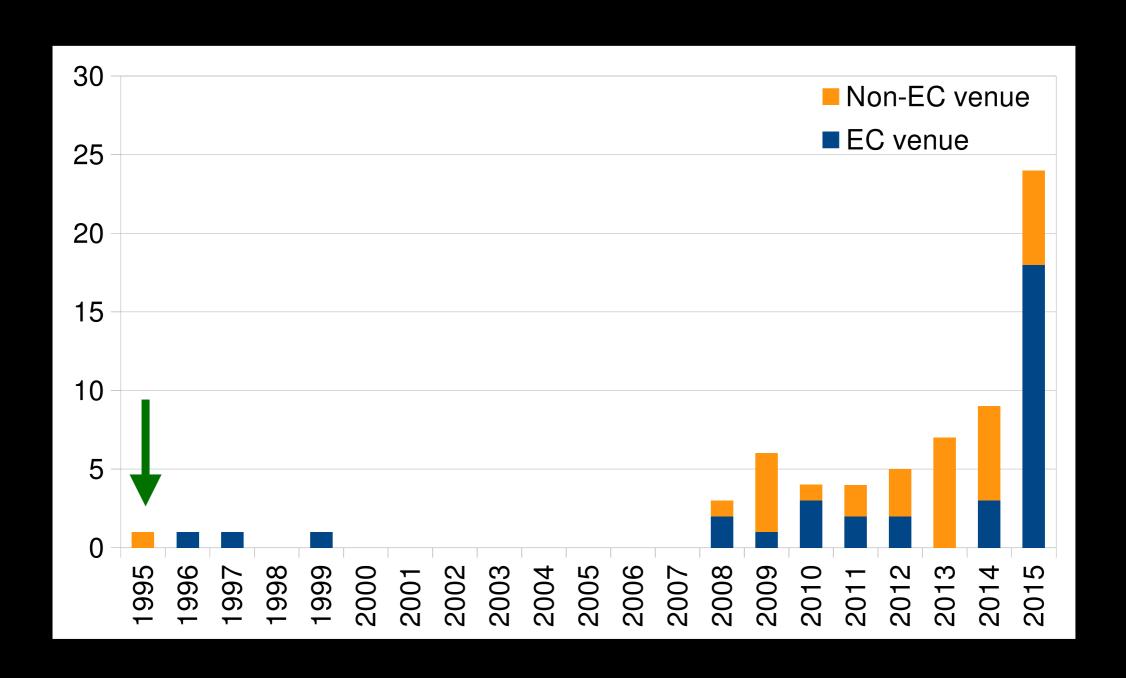
Petke et al., Transactions on Evolutionary Computation 2017





Genetic Improvement of Software: A Comprehensive Survey

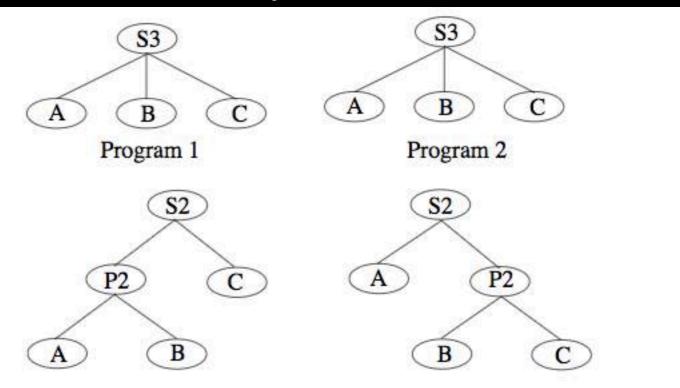
Petke et al., Transactions on Evolutionary Computation 2017





Automatic conversion of programs from serial to parallel using Genetic Programming The Paragen System

Walsh & Ryan, ParCo 1995



Parallel version of each program

Figure 3: Two example programs, shown in serial and parallel form

```
      Stmt A: a = b + 2
      Stmt A: a = a + b

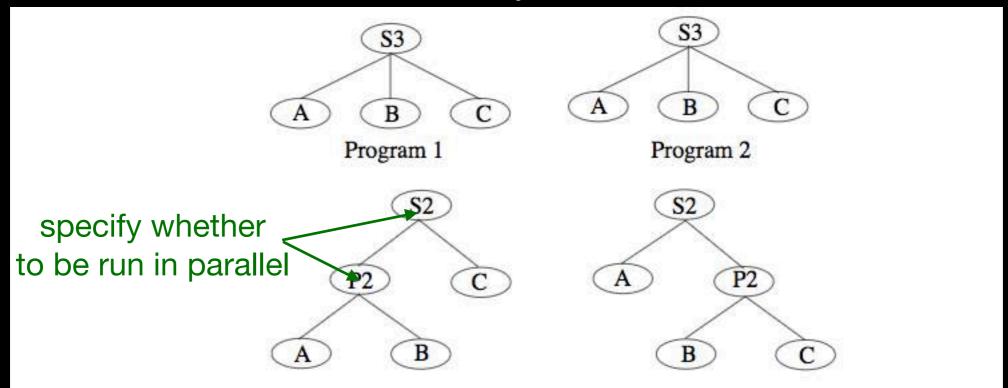
      Stmt B: c = b + 5
      Stmt B: b = a + 1

      Stmt C: c = e + f
      Stmt C: c = a + 3
```

Two example sequential programs, shown in parse tree form in Figure 3

Automatic conversion of programs from serial to parallel using Genetic Programming The Paragen System

Walsh & Ryan, ParCo 1995



Parallel version of each program

Figure 3: Two example programs, shown in serial and parallel form

```
      Stmt A: a = b + 2
      Stmt A: a = a + b

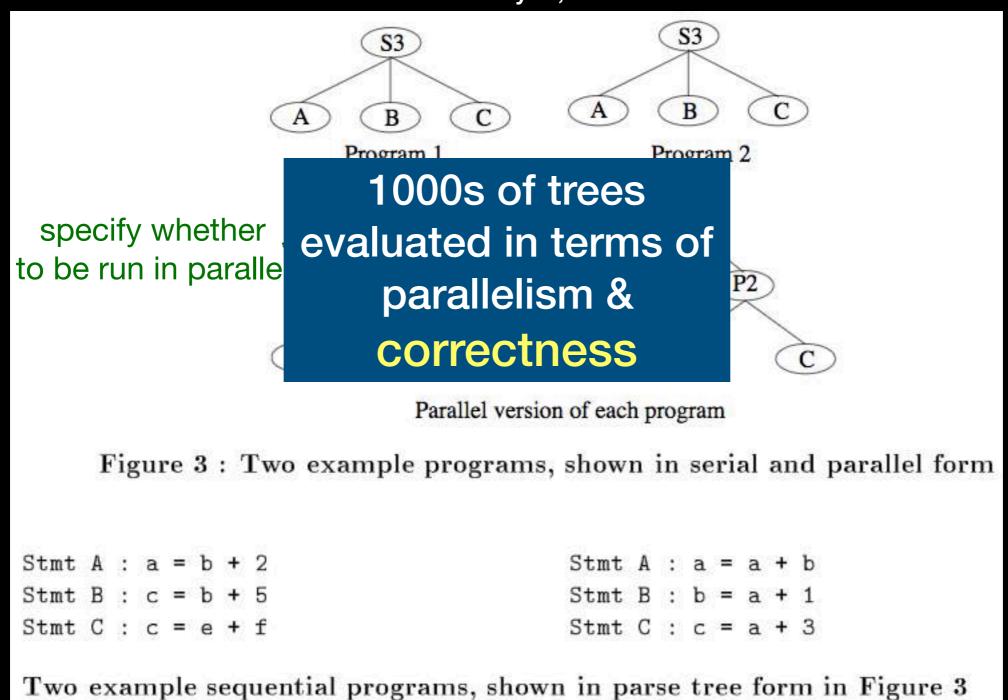
      Stmt B: c = b + 5
      Stmt B: b = a + 1

      Stmt C: c = e + f
      Stmt C: c = a + 3
```

Two example sequential programs, shown in parse tree form in Figure 3

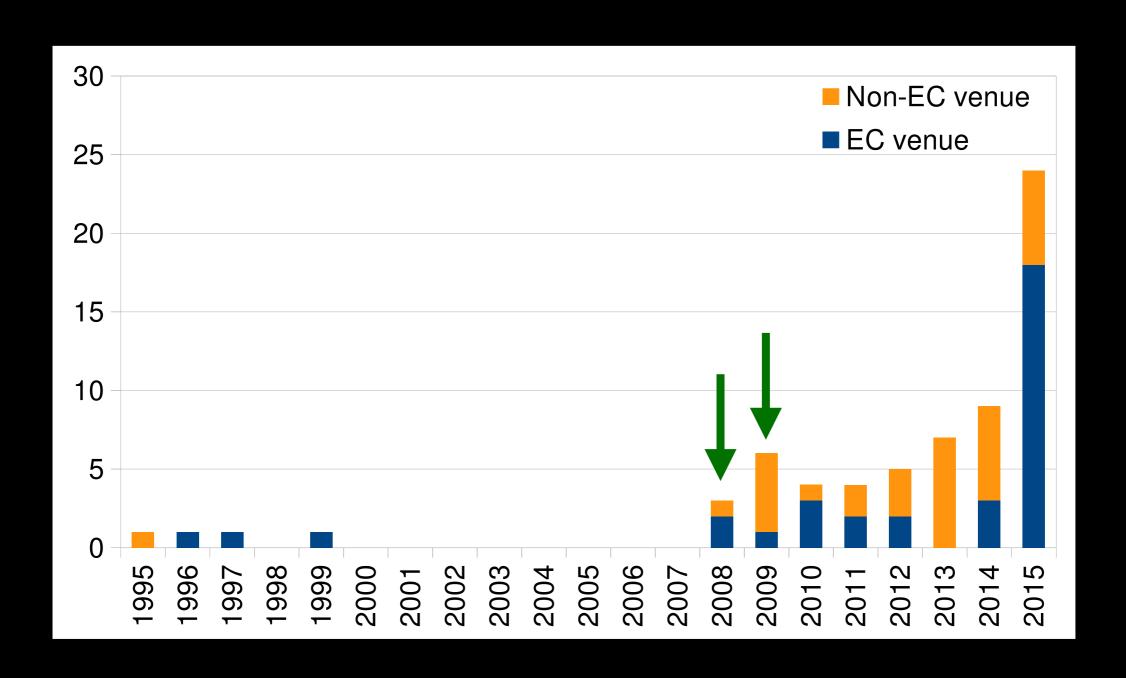
Automatic conversion of programs from serial to parallel using Genetic Programming The Paragen System

Walsh & Ryan, ParCo 1995



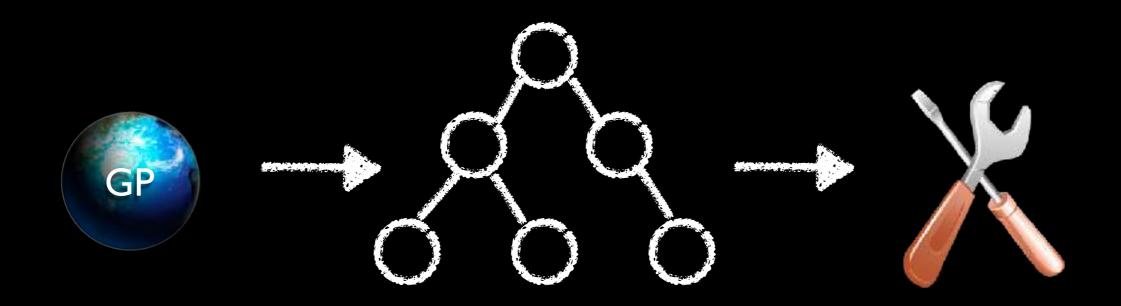
Genetic Improvement of Software: A Comprehensive Survey

Petke et al., Transactions on Evolutionary Computation 2017





Genetic Improvement milestones in 2008 & 2009



Genetic Improvement milestones in 2008 & 2009

Automated softwal formed a basis for the

first GP-based tool

Work by Arcuri et a for automated

e.g., "On the Automa software repair called s", ICSE 2008

GenProg

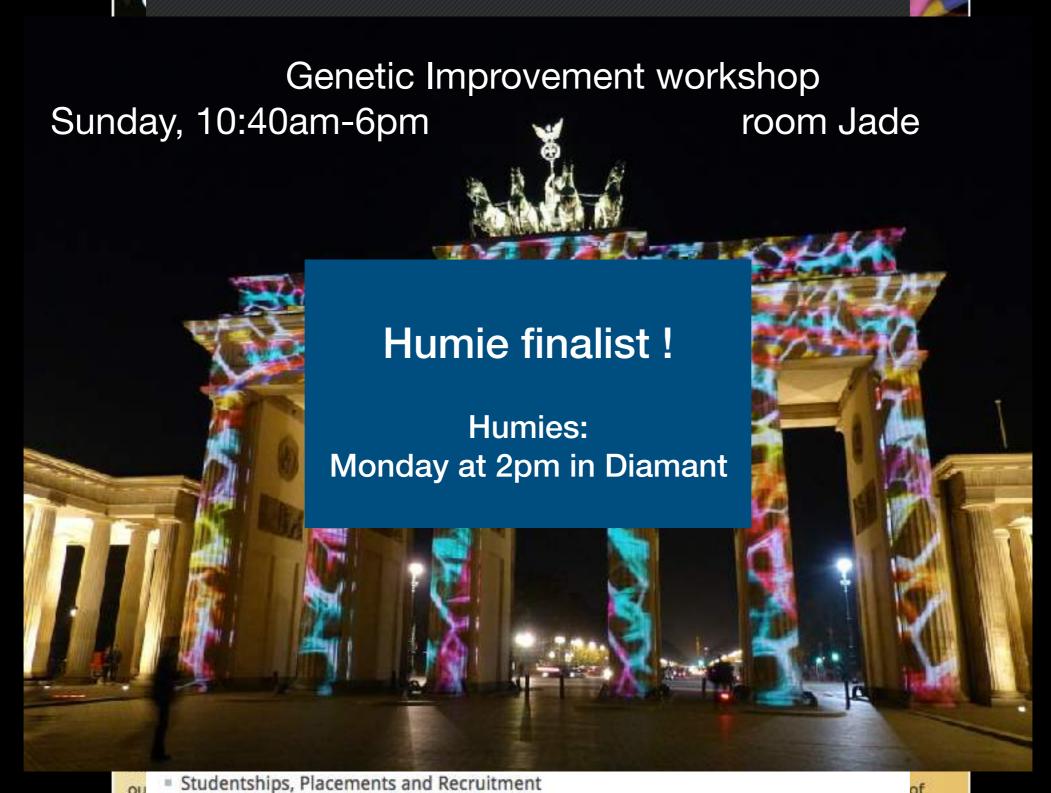
Work by Weimer, N

rest

e.g., "Automatically finding patches using genetic programming", ICSE 2009







CREST

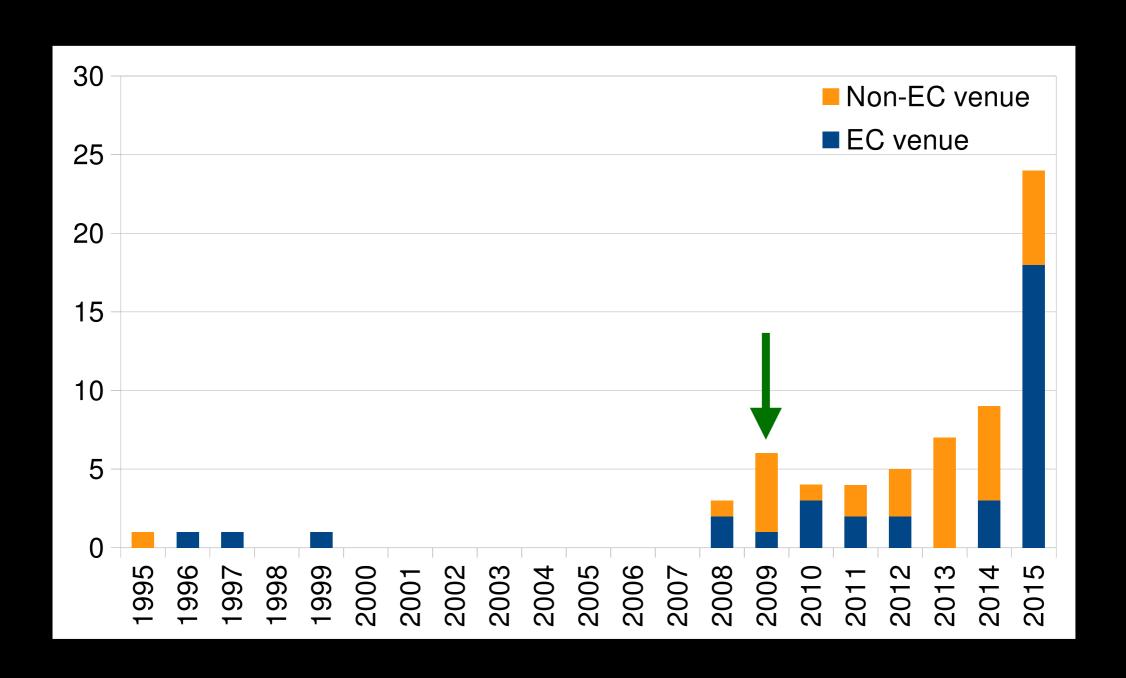
OI.

Technology Licensing & Consultancy

Feasibility & Proof of Concept Studies

Genetic Improvement of Software: A Comprehensive Survey

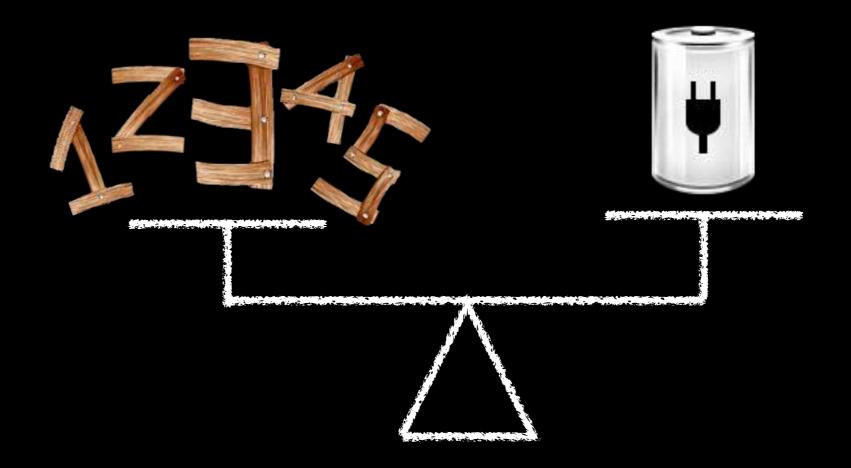
Petke et al., Transactions on Evolutionary Computation 2017





Proposed extension to non-functional properties

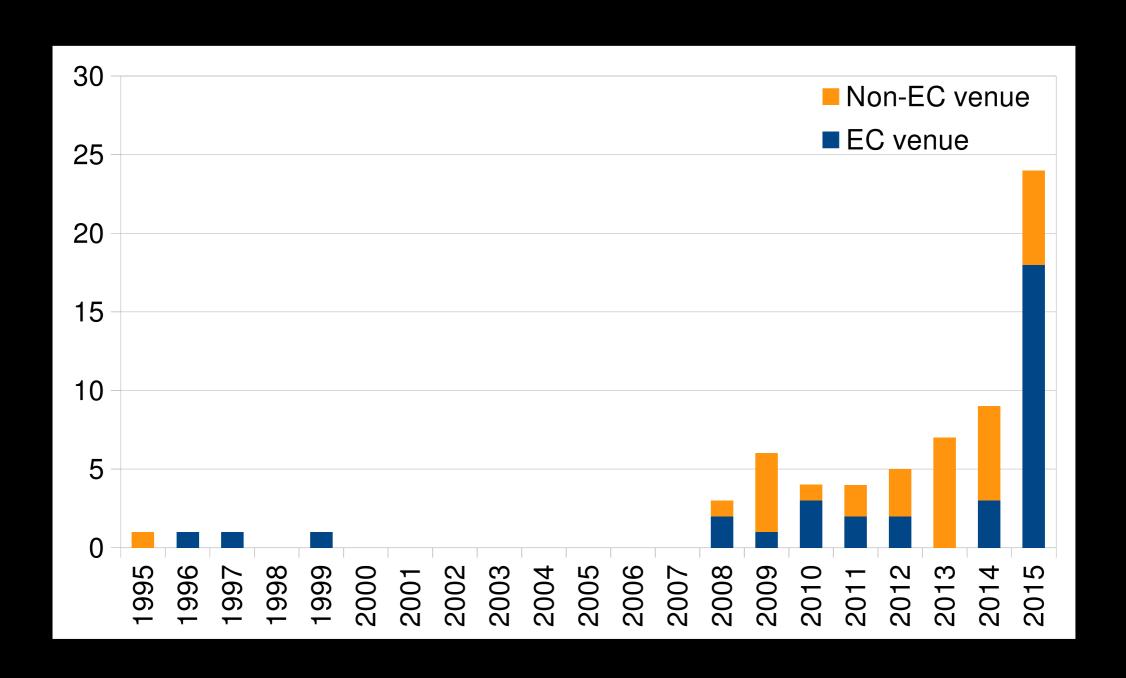
Work by David R. White et al. "Genetic Programming for Low-Resource Systems"





Genetic Improvement of Software: A Comprehensive Survey

Petke et al., Transactions on Evolutionary Computation 2017

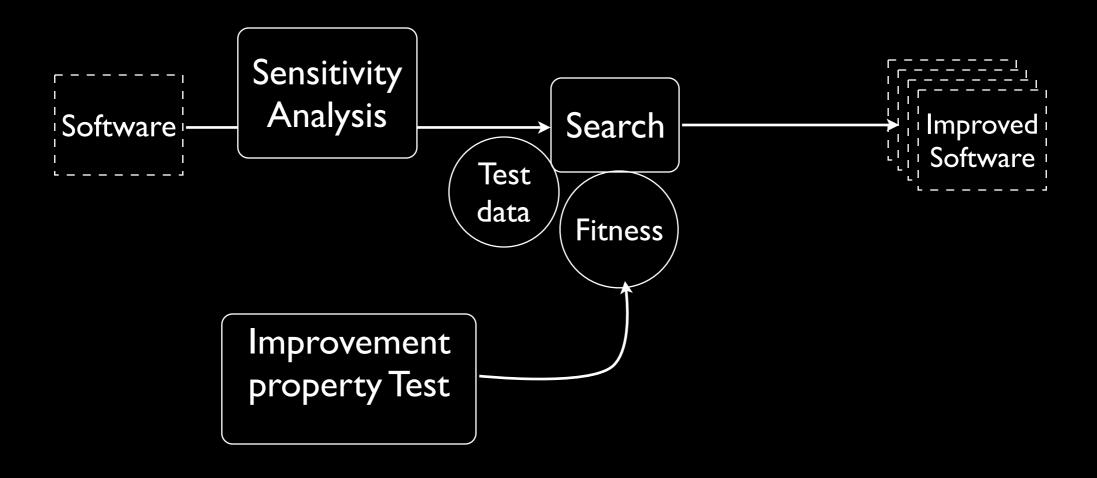




GI Framework

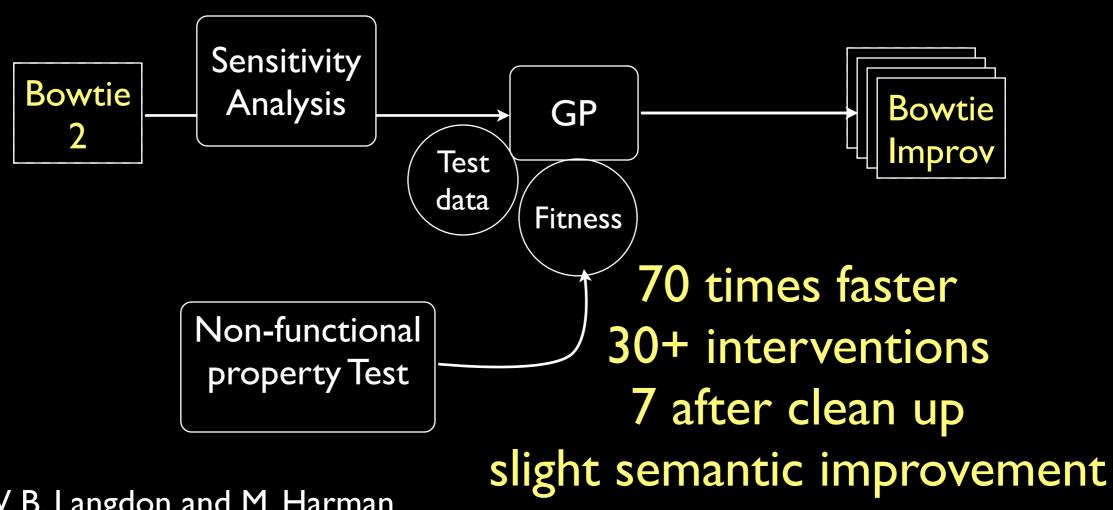


Genetic Improvement Framework





Efficiency Improvement



W. B. Langdon and M. Harman

Optimising Existing Software with Genetic Programming.

Transactions on Evolutionary Computation (TEC) 2015

Mutation	Source file	Line (type)	Orig. code	New code
replaced	bt2_io.cpp	622 (for2)	i < offsLenSampled	i < this->_nPat
replaced	sa_rescomb.cpp	50 (for2)	i < satup>offs.size()	O
Runtime reduction from 12 days to 4 hours				
replaced	aligner_swsse_ee_u8 .cpp	707	vh = _mm_max_epu8(vh, vf);	vmax = vlo;
deleted	aligner_swsse_ee_u8 .cpp	766	pvFStore += 4;	
replaced	aligner_swsse_ee_u8	772	_mm_store_si128(pv	vh = mm max epu8(vh.

772

778

HStore, vh);

ve =

_mm_max_epu8(ve,

vh);



replaced

deleted

_mm_max_epu8(vh,

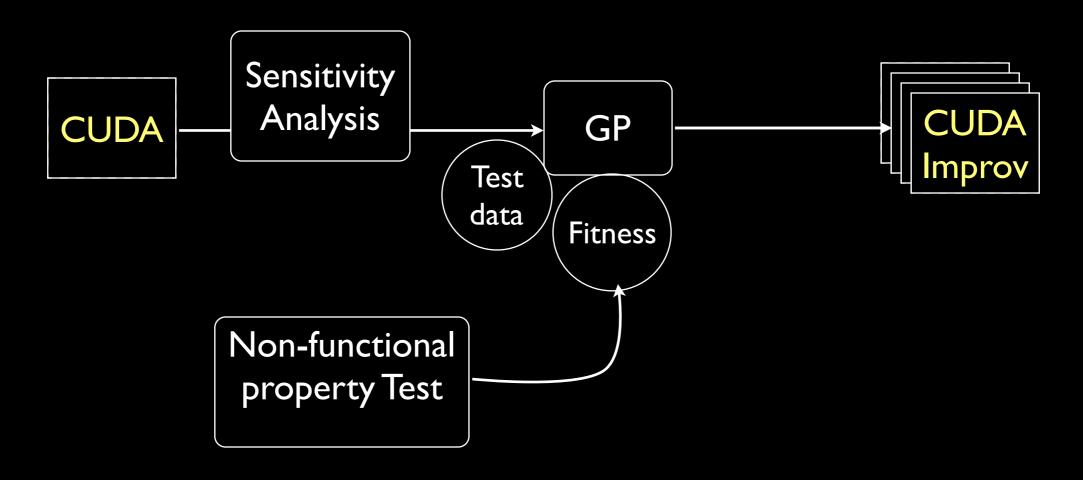
vf);

.cpp

aligner_swsse_ee_u8

.cpp

Efficiency Improvement



W.B. Langdon, B.Y.H. Lam, J. Petke & M. Harman Improving CUDA DNA Analysis Software with Genetic Programming Genetic and Evolutionary Computation Conference (GECCO) 2015

Efficiency Improvement

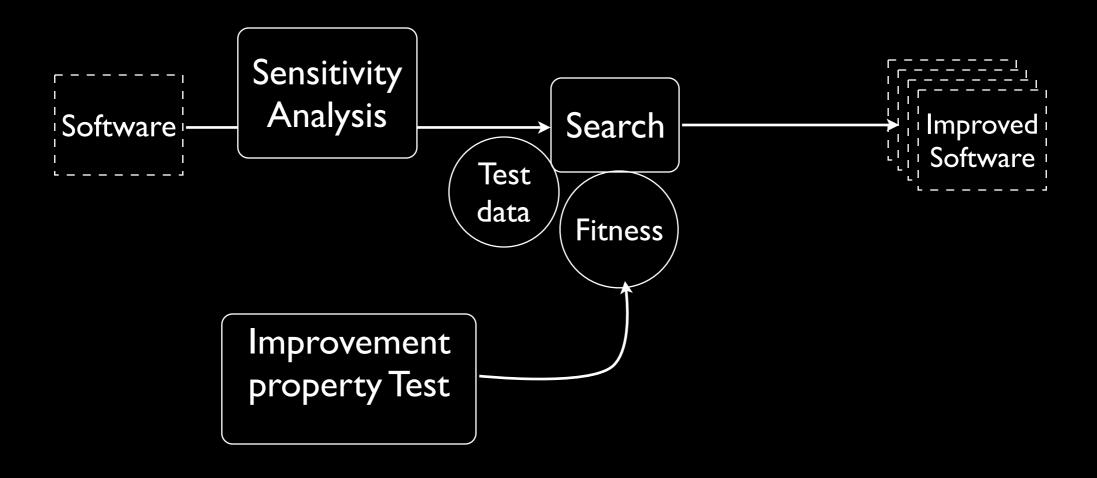
Challenge: Use genetic improvement to improve program efficiency of a state-of-the-art bioinformatics program for DNA sequence mapping called BarraCUDA, consisting of 8,000+ lines of code.

Results: The improved version of BarraCUDA is up to 3x faster than the original on large real-world datasets. The new version has been adopted into development and has been downloaded over 1,000 times so far. Ported by IBM to one of their super computers and adopted by Lab7.

W.B. Langdon, B.Y.H. Lam, J. Petke & M. Harman Improving CUDA DNA Analysis Software with Genetic Programming Genetic and Evolutionary Computation Conference (GECCO) 2015

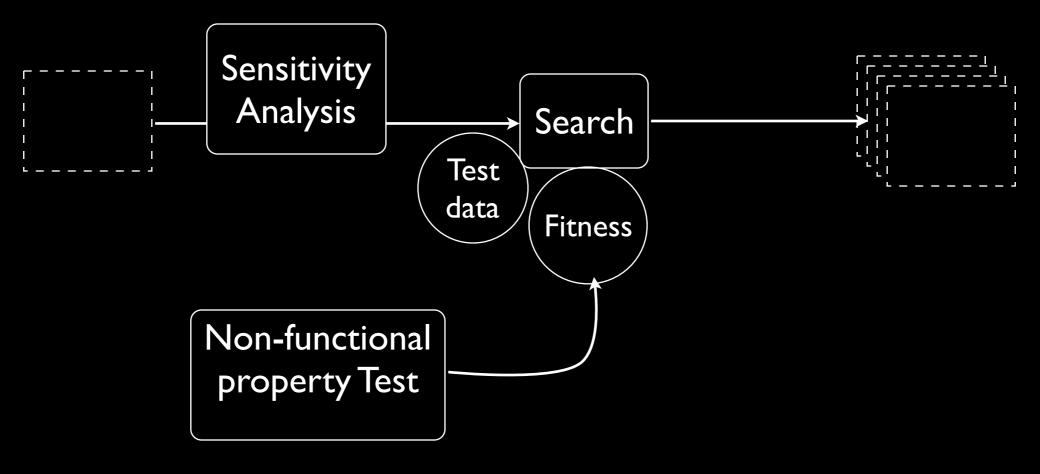


Genetic Improvement Framework



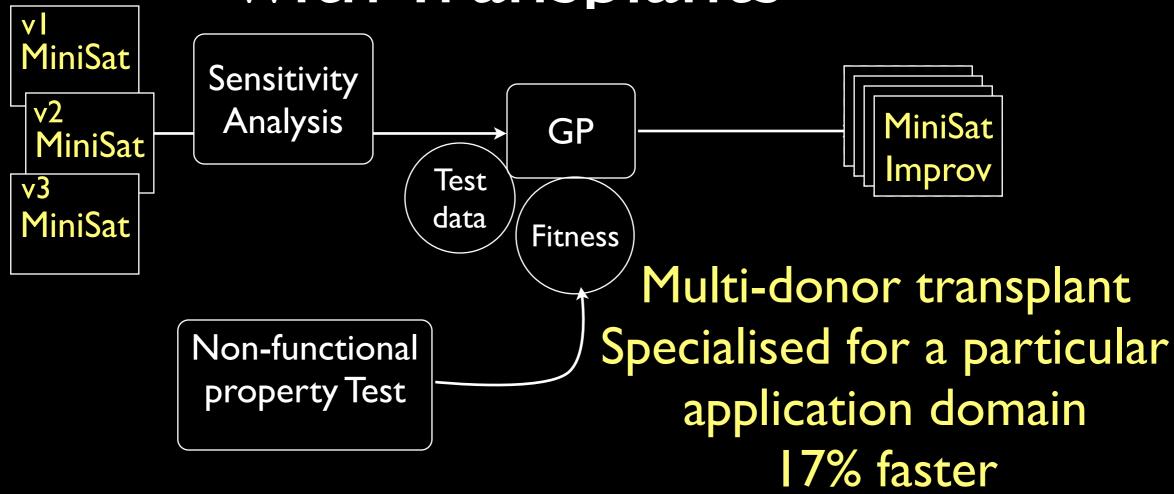


Software Specialisation with Transplants





Software Specialisation with Transplants



Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer Using Genetic Improvement & Code Transplants to Specialise a C++ program to a Problem Class European Conference on Genetic Programming (EuroGP) 2014



Genetic and Evolutionary Computation Conference

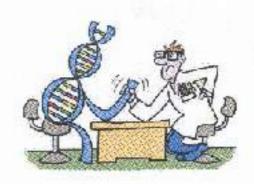
July 12-16, 2014 Vancouver, British Columbia

Winners of the 2014 Humies Silver Award:

Justyna Petke, Mark Harman, William B. Langdon, Westley Weimer

Using Genetic Improvement and Code Transplants to Specialize a C++ Program to a Problem Class





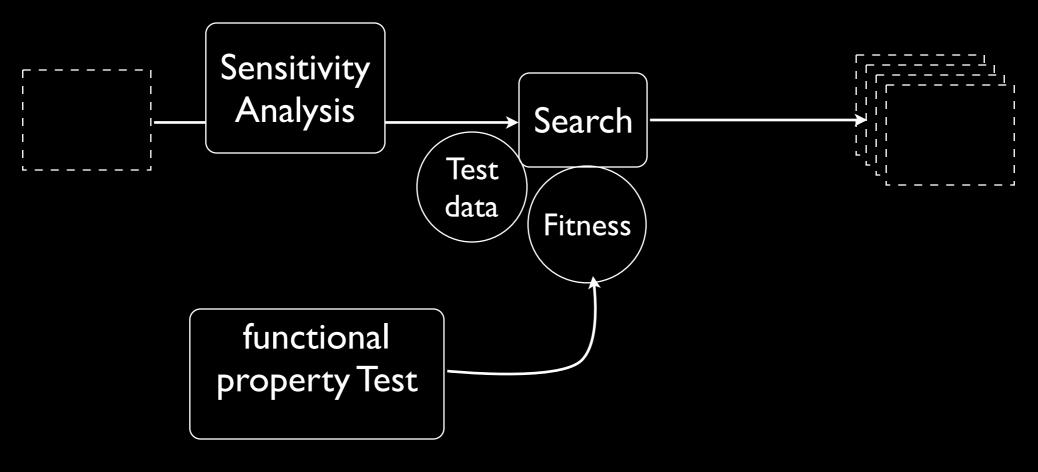


Africa's Constituted and immediately



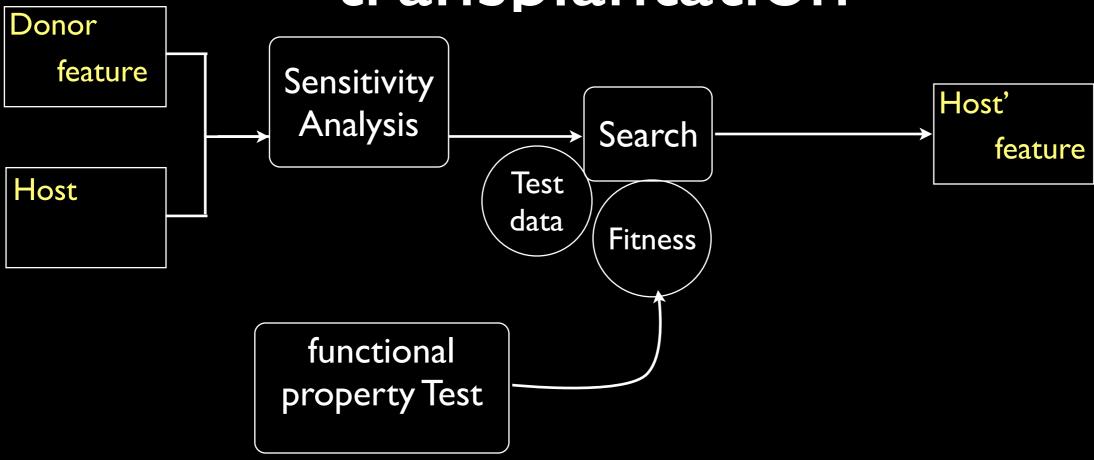


Real-world cross-system transplantation





Real-world cross-system transplantation



Earl T. Barr, Mark Harman, Yue Jia, Alexandru Marginean, and Justyna Petke Automated Software Transplantation International Symposium on Software Testing and Analysis (ISSTA) 2015

Why A ~100 players ansplantation?



Video Player

Start from scraich

Check open source repositories





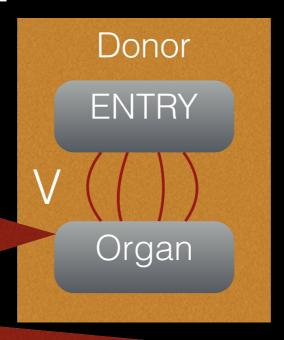
Automated Software Transplantation



Organ Entry

Organ's Test Suite

Implantation Point

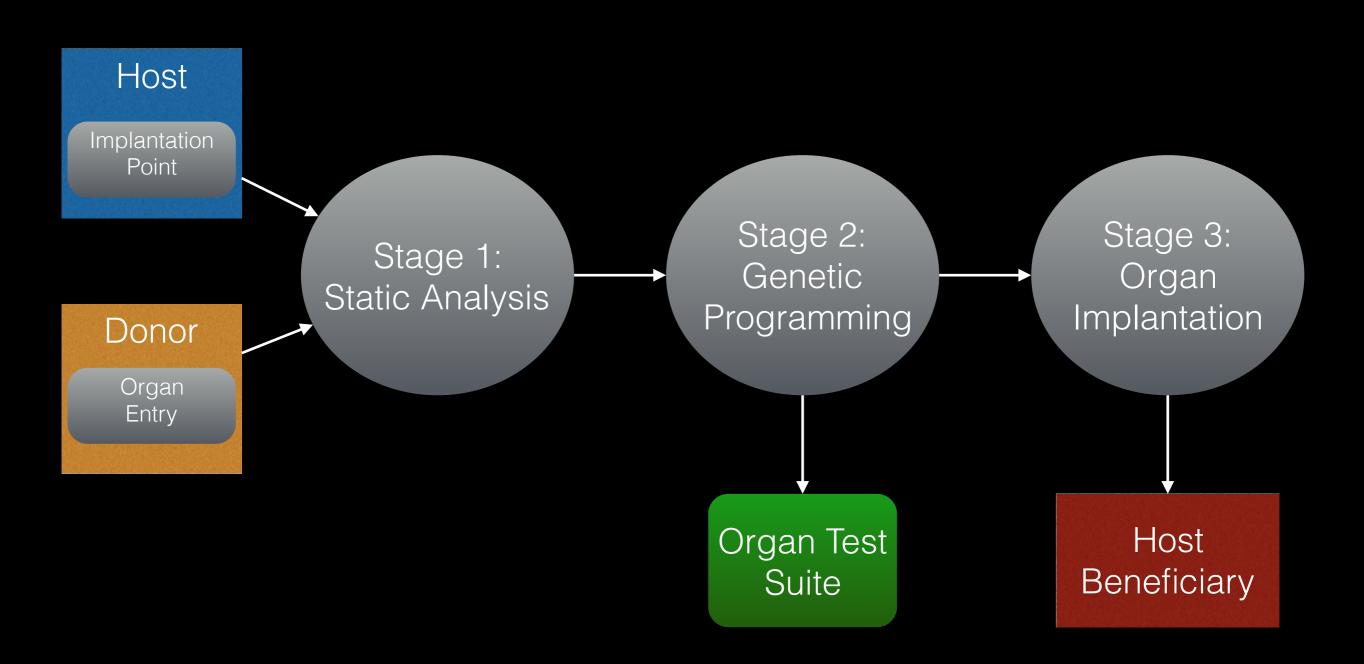


Host





uTrans





Empirical Study

Donor	Host	All Passed	Regression	Regression++	Acceptance
Idct	Pidgin	16	20	17	16
Mytar	Pidgin	16	20	18	20
Web	Pidgi			0	18
Cflow	Pidg			5	16
Tux	Pidg	:- 12	4 -£ F	7	16
Idct	Cflo [,]		t of 15 exp		16
Mytar	Cflo' W	e success	sfully autot	cransplanted 7	20
Web	Cflo'	nev	w function	ality	17
Cflow	Cflo'			'	20
Tux	Cflo _'			4	16
Idct	SoX			17	16
Mytar	SoX	17	17	17	20
Web	SoX	0	0	0	17
Cflow	SoX	14	16	15	14
Tux	SoX	13	13	13	14
TOTAL		188/300	233/300	196/300	256/300



Case Study

H.264

within 26 hours performed a task that took developers an avg of 20 days of elapsed time

+ Acceptance % 100%





Automated Software Transplantation

E.T. Barr, M. Harman, Y. Jia, A. Marginean & J. Petke

ACM Distinguished Paper Award at ISSTA 2015

Gold 'Humie' Award Winner at GECCO 2016





Code 'transplant' could revolutionise programming

PROGRAMMING F ROLLIGT IS F BY LAMES TEMPERTOR

Code has been automatically "transplanted" from one piece of software to another for the first time, with researchers claiming the breakthrough could radically change how computer programs are

The process, demonstrated by researchers at University College London, has been likened to organ transplantation in humans. Known as Muscalpel, it works by isolating the code of a useful leasure in a 'donor' program and transplanting this "organ" to the right "vein" in software lacking the feature. Almost all of the transplant is automated, with minimal human involvement.

```
void loop()
 the article in WIRED.CO.UK
   if ((millis() - fn[mum_ma_task_o
    LITCINUM_HILTASK_OIT] .....
             or not) -(m)():
```

over 2,000 shares of



coverage in



Click

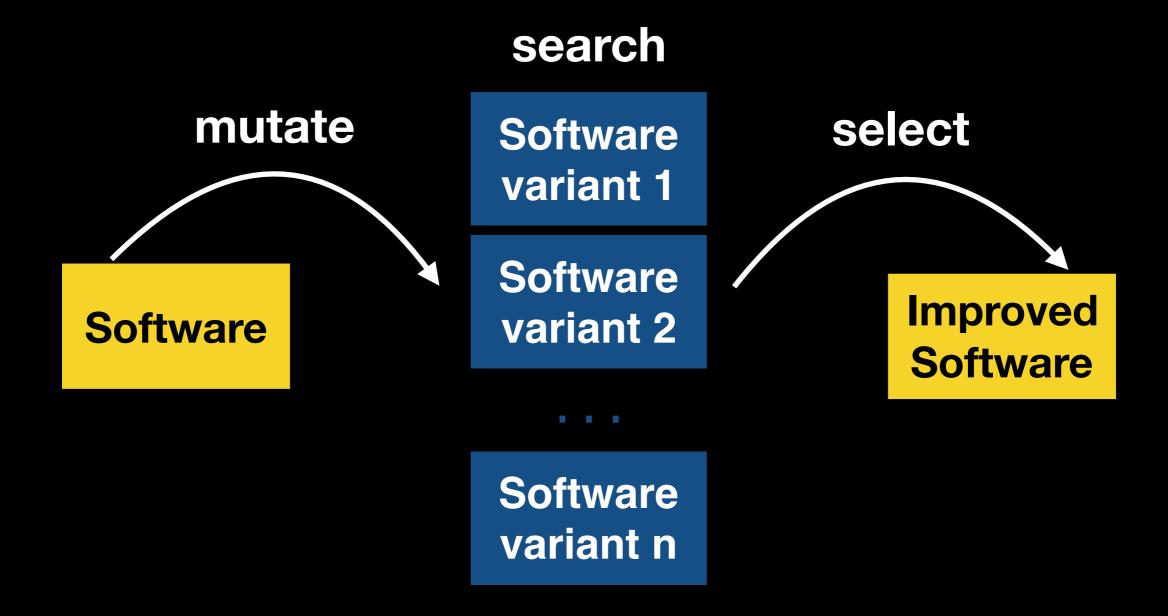


Summary



Genetic Improvement

uses search in order to improve existing software





Challenge



Challenge

Which non-functional and which functional properties of software can be improved using genetic improvement? Please base your answers based on the talks you attend in this session.

Sunday 10:50-12:30 Genetic Improvement Workshop (Jade)



Challenge

Which non-functional and which functional properties of software can be improved using genetic improvement? Please base your answers based on the talks you attend in this session.

Sunday 10:50-12:30 Genetic Improvement Workshop (Jade)



Summaries

3 papers from the GI workshop

including 1 Humie contender

2 out of 3 presented at the session related to the challenge

available from: http://geneticimprovementofsoftware.com



Summaries

Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success

Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir

Embedding Genetic Improvement into Programming Languages Shin Yoo

Genetic Improvement of Runtime in a Bioinformatics Application Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, Albert V. Smith and Vilmundur Gudnason





Summary: Genetic Improvement is the application of evolutionary and search-based optimisation methods to the improvement of existing software. For example, it may be used to automate the process of bug-fixing or execution time optimisation.

Assignment 1: Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions: How did the initial framework look like? How was it extended to handle large real-world cases? Is it industry ready? The following papers should help you with the review (feel free to include others):

- [1] Andrea Arcuri and Xin Yao, "A novel co-evolutionary approach to automatic software bug fixing". CEC 2008, pp. 162-168
- [2] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer: "A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each". ICSE 2012: 3-13
- [3] Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. "Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success". GI workshop at GECCO 2017

Assignment 2: Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/papers/gi.zip. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.

Assignment 3: Choose a paper presented at GECCO. Present the work and explain how would you extend it. For example, if it's a tool paper, show how to run it and propose how to extend it or apply to a new application (perhaps write a prototype if appropriate). If it's a research paper, for example, either state what are the limitations and how would you overcome them or propose an extension that would yield to a new or /wider application of the approach or what are the milestones that need to be overcome to make the approach industry-ready.

Expected output and deliverable: A short report for assignments 1 and 2 and presentation for all.



Summary: Genetic Improvement is the application of evolutionary and search-based optimisation methods to the improvement of existing software. For example, it may be used to automate the process of bug-fixing or execution time optimisation.

Assignment 1: Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions: How did the initial framework look like? How was it extended to handle large real-world cases? Is it industry ready? The following papers should help you with the review (feel free to include others):

- [1] Andrea Arcuri and Xin Yao, "A novel co-evolutionary approach to automatic software bug fixing". CEC 2008, pp. 162-168
- [2] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer: "A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each". ICSE 2012: 3-13
- [3] Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. "Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success". GI workshop at GECCO 2017

Assignment 2: Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/acmschool. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.

Assignment 3: Choose a paper presented at GECCO. Present the work and explain how would you extend it. For example, if it's a tool paper, show how to run it and propose how to extend it or apply to a new application (perhaps write a prototype if appropriate). If it's a research paper, for example, either state what are the limitations and how would you overcome them or propose an extension that would yield to a new or /wider application of the approach or what are the milestones that need to be overcome to make the approach industry-ready.

Expected output and deliverable: A short report for assignments 1 and 3 and presentation for all.



Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions: How did the initial framework look like? How was it extended to handle large real-world cases? Is it industry ready? The following papers should help you with the review (feel free to include others):

- [1] Andrea Arcuri and Xin Yao, "A novel co-evolutionary approach to automatic software bug fixing". CEC 2008, pp. 162-168
- [2] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer: "A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each". ICSE 2012: 3-13
- [3] Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. "Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success". Gl workshop at GECCO 2017



Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions: How did the initial framework look like? How was it extended to handle large real-world cases? Is it industry ready? The following papers should help you with the review (feel free to include others):

[1] Andrea Arcuri and Xin Yao, "A novel co-evolutionary approach to automatic software bug fixing". CEC 2008, pp. 162-168

[2] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer: "A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each". ICSE 2012: 3-13

[3] Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. "Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success". Gl workshop at GECCO 2017



Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions: How did the

initial framework look like? How was it ereal-world cases? Is it industry ready? Thelp you with the review (feel free to income the look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? Is it industry ready? The look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? How was it ereal-world cases? It is it industry the look like? It is it indust

Humie winner

[1] Andrea Arcuri and Xin Yao, "A novel co-evolutionary approach to automatic software bug fixing". CEC 2008, pp. 162-168

[2] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer: "A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each". ICSE 2012: 3-13

[3] Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. "Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success". Gl workshop at GECCO 2017



Humie finalist

[1] Andrea Arcuri and Xin Yao, "A novel co-evolution bug fixing". CEC 2008, pp. 162-168[2] Claire Le Goues, Michael Dewey-Vogt, Stephanie

Humies: Monday at 2pm in Diamant

systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each". ICSE 2012: 3-13

[3] Saemundur Oskar Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. "Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success". Gl workshop at GECCO 2017



Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions:

How did the initial framework look like? What was the novelty?

How was it extended to handle large real-world cases? What were the developments?

Is it industry ready? What's next for GP-based automated program repair?



Literature survey: Summarise and present the key developments in the field of automated program repair using genetic programming. In your report try to answer the following questions: How did the initial framework look like? How was it extended to handle large real-world cases? Is it industry ready?

Deliverables: a 3-4 page summary and a 10min talk



Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/acmschool. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.



Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/acmschool. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.

two options: minisatCIT (Linux) or minisat-2.2.0-gi (Linux and macOS X)

prerequisite: gawk



Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/acmschool. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.

What is the software representation?

What are the mutation and crossover operators?

What is the fitness value?



Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/acmschool. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.

How much runtime improvement was achieved?

What are the fitness values throughout generations?



Coding exercise: Download and run the GI framework used in the following paper: Justyna Petke, Mark Harman, William B. Langdon and Westley Weimer. "Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class". EuroGP 2014: 137-149. It is available from: http://www0.cs.ucl.ac.uk/staff/J.Petke/acmschool. Please report on your experience. In particular, try to answer the following question: What are the main components of the framework? Please describe and comment on the improvements obtained with the framework.

Deliverabels: a 10min talk



Research proposal: Choose a paper presented at GECCO. Present the work and explain how would you extend it. For example, if it's a tool paper, show how to run it and propose how to extend it or apply to a new application (perhaps write a prototype if appropriate). If it's a research paper, for example, either state what are the limitations and how would you overcome them or propose an extension that would yield to a new or /wider application of the approach or what are the milestones that need to be overcome to make the approach industry-ready.



Research proposal: Choose a paper presented at GECCO. Present the work and explain how would you extend it. For example, if it's a tool paper, show how to run it and propose how to extend it or apply to a new application (perhaps write a prototype if appropriate). If it's a research paper, for example, either state what are the limitations and how would you overcome them or propose an extension that would yield to a new or /wider application of the approach or what are the milestones that need to be overcome to make the approach industry-ready.

What is the motivation? What was done? What were the results?



Research proposal: Choose a paper presented at GECCO. Present the work and explain how would you extend it. For example, if it's a tool paper, show how to run it and propose how to extend it or apply to a new application (perhaps write a prototype if appropriate). If it's a research paper, for example, either state what are the limitations and how would you overcome them or propose an extension that would yield to a new or /wider application of the approach or what are the milestones that need to be overcome to make the approach industry-ready.

Why would you extend it?

What is your proposal?

Is it feasible? Show some initial investigation.



Research proposal: Choose a paper presented at GECCO. Present the work and explain how would you extend it. For example, if it's a tool paper, show how to run it and propose how to extend it or apply to a new application (perhaps write a prototype if appropriate). If it's a research paper, for example, either state what are the limitations and how would you overcome them or propose an extension that would yield to a new or /wider application of the approach or what are the milestones that need to be overcome to make the approach industry-ready.

Deliverables: a report or tool demo and a 10min talk



Summary

Assignment 1: Report on developments in GP-based automated software repair.

Assignment 2: Report on a run of a GI framework.

Assignment 3: Propose an extension of work presented at GECCO.

geneticimprovementofsoftware.com

