Mark Harman

List of Publications by Citations

279 papers	12,573 citations	53 h-index	101 g-index
300 ext. papers	15,300 ext. citations	2.2 avg, IF	7.06 L-index

#	Paper	IF	Citations
279	An Analysis and Survey of the Development of Mutation Testing. *IEEE Transactions on Software Engineering*, **2011**, 37, 649-678	3.5	818
278	Regression testing minimization, selection and prioritization: a survey. *Software Testing Verification and Reliability*, **2012**, 22, 67-120	0.9	623
277	Search-based software engineering. *Information and Software Technology*, **2001**, 43, 833-839	3.4	485
276	Search-based software engineering. *ACM Computing Surveys*, **2012**, 45, 1-61	13.4	431
275	Search Algorithms for Regression Test Case Prioritization. *IEEE Transactions on Software Engineering*, **2007**, 33, 225-237	3.5	420
274	An orchestrated survey of methodologies for automated software test case generation. *Journal of Systems and Software*, **2013**, 86, 1978-2001	3.3	341
273	The Oracle Problem in Software Testing: A Survey. *IEEE Transactions on Software Engineering*, **2015**, 41, 507-525	3.5	337
272	The Current State and Future of Search Based Software Engineering **2007**,		326
271	Software Module Clustering as a Multi-Objective Search Problem. *IEEE Transactions on Software Engineering*, **2011**, 37, 264-282	3.5	234
270	A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search. *IEEE Transactions on Software Engineering*, **2010**, 36, 226-247	3.5	234
269	Using formal specifications to support testing. *ACM Computing Surveys*, **2009**, 41, 1-76	13.4	207
268	Pareto efficient multi-objective test case selection **2007**,		186
267	Sapienz: multi-objective automated testing for Android applications **2016**,		169
266	Higher Order Mutation Testing. *Information and Software Technology*, **2009**, 51, 1379-1393	3.4	163
265	Testability transformation. *IEEE Transactions on Software Engineering*, **2004**, 30, 3-16	3.5	161
264	A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, **2017**, 126, 57-84	3.3	148
263	App store mining and analysis: MSR for app stores **2012**,		148

262	Using program slicing to assist in the detection of equivalent mutants **1999**, 9, 233-262		143
261	The multi-objective next release problem **2007**,		121
260	Pareto optimal search based refactoring at the design level **2007**,		120
259	Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	110
258	Mutation Testing Advances: An Analysis and Survey. *Advances in Computers*, **2019**, 275-378	2.9	101
257	**2008**,		100
256	Optimizing Existing Software With Genetic Programming. *IEEE Transactions on Evolutionary Computation*, **2015**, 19, 118-135	15.6	99
255	Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge **2009**,		97
254	Search Based Software Engineering: Techniques, Taxonomy, Tutorial. *Lecture Notes in Computer Science*, **2012**, 1-59	0.9	94
253	An overview of program slicing. *Software Focus*, **2001**, 2, 85-92		92
252	Efficient multi-objective higher order mutation testing with genetic programming. *Journal of Systems and Software*, **2010**, 83, 2416-2430	3.3	91
251	Searching for better configurations: a rigorous approach to clone evaluation **2013**,		90
250	Using hybrid algorithm for Pareto efficient multi-objective test suite minimisation. *Journal of Systems and Software*, **2010**, 83, 689-701	3.3	86
249	The plastic surgery hypothesis **2014**,		84
248	MILU: A Customizable, Runtime-Optimized Higher Order Mutation Testing Tool for the Full C Language **2008**,		84
247	A study of equivalent and stubborn mutation operators using human analysis of equivalence **2014**,		83
246	Strong higher order mutation-based test data generation **2011**,		83
245	Achievements, Open Problems and Challenges for Search Based Software Testing **2015**,		79

244	Search Based Approaches to Component Selection and Prioritization for the Next Release Problem. *Conference on Software Maintenance, Proceedings of the*, **2006**,		75
243	A multi-objective approach to search-based test data generation **2007**,		74
242	Multi-objective software effort estimation **2016**,		72
241	A Survey of Empirical Results on Program Slicing. *Advances in Computers*, **2004**, 62, 105-178	2.9	71
240	Comparing white-box and black-box test prioritization **2016**,		68
239	The GISMOE challenge: constructing the pareto program surface using genetic programming to find better programs (keynote paper) **2012**,		66
238	Using Genetic Improvement and Code Transplants to Specialise a C++ Program to a Problem Class. *Lecture Notes in Computer Science*, **2014**, 137-149	0.9	66
237	Fault localization prioritization. *ACM Transactions on Software Engineering and Methodology*, **2013**, 22, 1-29	3.3	65
236	Amorphous program slicing. *Journal of Systems and Software*, **2003**, 68, 45-64	3.3	64
235	A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirements Engineering*, **2009**, 14, 231-245	2.7	63
234	Automated software transplantation **2015**,		62
233	Using program slicing to simplify testing. *Software Testing Verification and Reliability*, **1995**, 5, 143-162	0.9	62
232	Search Based Requirements Optimisation: Existing Work and Challenges **2008**, 88-94		62
231	Trivial Compiler Equivalence: A Large Scale Empirical Study of a Simple, Fast and Effective Equivalent Mutant Detection Technique **2015**,		61
230	Testing and verification in service-oriented architecture: a survey. *Software Testing Verification and Reliability*, **2013**, 23, 261-313	0.9	58
229	Reducing Energy Consumption Using Genetic Improvement **2015**,		57
228	How to Overcome the Equivalent Mutant Problem and Achieve Tailored Selective Mutation Using Co-evolution. *Lecture Notes in Computer Science*, **2004**, 1338-1349	0.9	57
227	An empirical study of the robustness of two module clustering fitness functions **2005**,		54

226	A theoretical & empirical znalysis of evolutionary testing and hill climbing for structural test data generation **2007**,		53
225	Efficiency and early fault detection with lower and higher strength combinatorial interaction testing **2013**,		52
224	Practical Combinatorial Interaction Testing: Empirical Findings on Efficiency and Early Fault Detection. *IEEE Transactions on Software Engineering*, **2015**, 41, 901-924	3.5	51
223	Deep Parameter Optimisation **2015**,		51
222	The App Sampling Problem for App Store Mining **2015**,		51
221	Automated web application testing using search based software engineering **2011**,		51
220	An empirical study of static program slice size. *ACM Transactions on Software Engineering and Methodology*, **2007**, 16, 8	3.3	51
219	Threats to the validity of mutation-based test assessment **2016**,		51
218	Software Engineering Meets Evolutionary Computation. *Computer*, **2011**, 44, 31-39	1.6	50
217	**2015**,		49
216	Symbolic search-based testing **2011**,		49
215	A study of the bi-objective next release problem. *Empirical Software Engineering*, **2011**, 16, 29-60	3.3	49
214	Evolutionary testing in the presence of loop-assigned flags **2004**,		48
213	Experimental assessment of software metrics using automated refactoring **2012**,		46
212	The impact of input domain reduction on search-based test data generation **2007**,		45
211	Automated Unique Input Output Sequence Generation for Conformance Testing of FSMs. *Computer Journal*, **2005**, 49, 331-344	1.3	45
210	Empirical evaluation of pareto efficient multi-objective regression test case prioritisation **2015**,		44
209	Dynamic adaptive search based software engineering **2012**,		43

208	The relationship between search based software engineering and predictive modeling **2010**,		43
207	**2010**,		43
206	Provably Optimal and Human-Competitive Results in SBSE for Spectrum Based Fault Localisation. *Lecture Notes in Computer Science*, **2013**, 224-238	0.9	42
205	Empirical evaluation of search based requirements interaction management. *Information and Software Technology*, **2013**, 55, 126-152	3.4	41
204	An Integer Linear Programming approach to the single and bi-objective Next Release Problem. *Information and Software Technology*, **2015**, 65, 1-13	3.4	41
203	A Manifesto for Higher Order Mutation Testing **2010**,		41
202	An empirical investigation into branch coverage for C programs using CUTE and AUSTIN. *Journal of Systems and Software*, **2010**, 83, 2379-2391	3.3	41
201	Tool-Supported Refactoring of Existing Object-Oriented Code into Aspects. *IEEE Transactions on Software Engineering*, **2006**, 32, 698-717	3.5	40
200	**2009**,		38
199	The use of search-based optimization techniques to schedule and staff software projects: an approach and an empirical study. *Software - Practice and Experience*, **2011**, 41, 495-519	2.5	37
198	Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering **2009**,		36
197	SapFix: Automated End-to-End Repair at Scale **2019**,		35
196	Highly Scalable Multi Objective Test Suite Minimisation Using Graphics Cards. *Lecture Notes in Computer Science*, **2011**, 219-236	0.9	35
195	Predictive Mutation Testing. *IEEE Transactions on Software Engineering*, **2019**, 45, 898-918	3.5	35
194	The role of Artificial Intelligence in Software Engineering **2012**,		34
193	The species per path approach to SearchBased test data generation **2006**,		34
192	Improving CUDA DNA Analysis Software with Genetic Programming **2015**,		33
191	AUSTIN: An open source tool for search based software testing of C programs. *Information and Software Technology*, **2013**, 55, 112-125	3.4	33

190	An analysis of the relationship between conditional entropy and failed error propagation in software testing **2014**,		33
189	ORBS: language-independent program slicing **2014**,		33
188	Babel Pidgin: SBSE Can Grow and Graft Entirely New Functionality into a Real World System. *Lecture Notes in Computer Science*, **2014**, 247-252	0.9	33
187	Input Domain Reduction through Irrelevant Variable Removal and Its Effect on Local, Global, and Hybrid Search-Based Structural Test Data Generation. *IEEE Transactions on Software Engineering*, **2012**, 38, 453-477	3.5	32
186	Feature lifecycles as they spread, migrate, remain, and die in App Stores **2015**,		32
185	Coverage and fault detection of the output-uniqueness test selection criteria **2014**,		32
184	Cloud engineering is Search Based Software Engineering too. *Journal of Systems and Software*, **2013**, 86, 2225-2241	3.3	32
183	Detecting Trivial Mutant Equivalences via Compiler Optimisations. *IEEE Transactions on Software Engineering*, **2018**, 44, 308-333	3.5	31
182	Learning Combinatorial Interaction Test Generation Strategies Using Hyperheuristic Search **2015**,		31
181	Reducing qualitative human oracle costs associated with automatically generated test data **2010**,		31
180	Conditioned slicing supports partition testing. *Software Testing Verification and Reliability*, **2002**, 12, 23-28	1.89	31
179	Pricing crowdsourcing-based software development tasks **2013**,		30
178	Empirical evaluation of a nesting testability transformation for evolutionary testing. *ACM Transactions on Software Engineering and Methodology*, **2009**, 18, 1-27	3.3	30
177	Theoretical foundations of dynamic program slicing. *Theoretical Computer Science*, **2006**, 360, 23-41	1.1	30
176	GPGPU test suite minimisation: search based software engineering performance improvement using graphics cards. *Empirical Software Engineering*, **2013**, 18, 550-593	3.3	28
175	Exact scalable sensitivity analysis for the next release problem. *ACM Transactions on Software Engineering and Methodology*, **2014**, 23, 1-31	3.3	28
174	FloPSy - Search-Based Floating Point Constraint Solving for Symbolic Execution. *Lecture Notes in Computer Science*, **2010**, 142-157	0.9	28
173	**2013**,		27

172	Genetic improvement for adaptive software engineering (keynote) **2014**,		27
171	Test data regeneration: generating new test data from existing test data. *Software Testing Verification and Reliability*, **2012**, 22, 171-201	0.9	27
170	Evolutionary testing of autonomous software agents. *Autonomous Agents and Multi-Agent Systems*, **2012**, 25, 260-283	2	27
169	Why Source Code Analysis and Manipulation Will Always be Important **2010**,		27
168	Automated test data generation for aspect-oriented programs **2009**,		27
167	Making the Case for MORTO: Multi Objective Regression Test Optimization **2011**,		27
166	Fairness Analysis in Requirements Assignments **2008**,		27
165	A formalisation of the relationship between forms of program slicing. *Science of Computer Programming*, **2006**, 62, 228-252	1.1	27
164	An empirical investigation of the influence of a type of side effects on program comprehension. *IEEE Transactions on Software Engineering*, **2003**, 29, 665-670	3.5	27
163	State-based model slicing. *ACM Computing Surveys*, **2013**, 45, 1-36	13.4	26
162	A parallel algorithm for static program slicing. *Information Processing Letters*, **1995**, 56, 307-313	0.8	26
161	Improving 3D medical image registration CUDA software with genetic programming **2014**,		25
160	Control Dependence for Extended Finite State Machines. *Lecture Notes in Computer Science*, **2009**, 216-230	0.9	25
159	Human Competitiveness of Genetic Programming in Spectrum-Based Fault Localisation. *ACM Transactions on Software Engineering and Methodology*, **2017**, 26, 1-30	3.3	24
158	Comparing the performance of metaheuristics for the analysis of multi-stakeholder tradeoffs in requirements optimisation. *Information and Software Technology*, **2011**, 53, 761-773	3.4	24
157	Dependence clusters in source code. *ACM Transactions on Programming Languages and Systems*, **2009**, 32, 1-33	1.6	24
156	Search based data sensitivity analysis applied to requirement engineering **2009**,		24
155	Estimating the feasibility of transition paths in extended finite state machines. *Automated Software Engineering*, **2010**, 17, 33-56	1.5	24

154	Deploying Search Based Software Engineering with Sapienz at Facebook. *Lecture Notes in Computer Science*, **2018**, 3-45	0.9	24
153	Adaptive Multi-Objective Evolutionary Algorithms for Overtime Planning in Software Projects. *IEEE Transactions on Software Engineering*, **2017**, 43, 898-917	3.5	23
152	Mutation-aware fault prediction **2016**,		23
151	Automatically generating realistic test input from web services **2011**,		23
150	**2009**,		23
149	Handling dynamic data structures in search based testing **2008**,		23
148	Search Based Software Engineering for Program Comprehension **2007**,		23
147	Why the Virtual Nature of Software Makes It Ideal for Search Based Optimization. *Lecture Notes in Computer Science*, **2010**, 1-12	0.9	23
146	Cooperative Co-evolutionary Optimization of Software Project Staff Assignments and Job Scheduling. *Lecture Notes in Computer Science*, **2011**, 127-141	0.9	23
145	Genetic programming for Reverse Engineering **2013**,		22
144	Angels and monsters **2014**,		22
143	Regression test suite prioritization using system models. *Software Testing Verification and Reliability*, **2012**, 22, 481-506	0.9	22
142	Search Based Optimization of Requirements Interaction Management **2010**,		22
141	A new algorithm for slicing unstructured programs. *Journal of Software: Evolution and Process*, **1998** , 10, 415-441		22
140	Robust next release problem **2014**,		21
139	. *IEEE Transactions on Software Engineering*, **2004**, 30, 715-735	3.5	21
138	An Empirical Study of Meta- and Hyper-Heuristic Search for Multi-Objective Release Planning. *ACM Transactions on Software Engineering and Methodology*, **2018**, 27, 1-32	3.3	20
137	Genetic improvement of GPU software. *Genetic Programming and Evolvable Machines*, **2017**, 18, 5-44	2	20

136	Automated patching techniques. *Communications of the ACM*, **2010**, 53, 108-108	2.5	20
135	Search Based Software Engineering: Introduction to the Special Issue of the IEEE Transactions on Software Engineering. *IEEE Transactions on Software Engineering*, **2010**, 36, 737-741	3.5	20
134	Automated Test Data Generation using Search Based Software Engineering **2007**,		20
133	CONSIT: a fully automated conditioned program slicer. *Software - Practice and Experience*, **2004**, 34, 15-46	2.5	20
132	Genetically Improved CUDA C++ Software. *Lecture Notes in Computer Science*, **2014**, 87-99	0.9	20
131	The Relationship Between Program Dependence and Mutation Analysis **2001**, 5-13		20
130	The importance of accounting for real-world labelling when predicting software vulnerabilities **2019**,		19
129	Search-Based Software Project Management **2014**, 373-399		19
128	A unifying theory of control dependence and its application to arbitrary program structures. *Theoretical Computer Science*, **2011**, 412, 6809-6842	1.1	19
127	AUSTIN: A Tool for Search Based Software Testing for the C Language and Its Evaluation on Deployed Automotive Systems **2010**,		19
126	Empirical study of optimization techniques for massive slicing. *ACM Transactions on Programming Languages and Systems*, **2007**, 30, 3	1.6	19
125	Branch-Coverage Testability Transformation for Unstructured Programs. *Computer Journal*, **2005**, 48, 421-436	1.3	19
124	Computing Unique Input/Output Sequences Using Genetic Algorithms. *Lecture Notes in Computer Science*, **2004**, 164-177	0.9	19
123	Applying Genetic Improvement to MiniSAT. *Lecture Notes in Computer Science*, **2013**, 257-262	0.9	19
122	Grow and Graft a Better CUDA pknotsRG for RNA Pseudoknot Free Energy Calculation **2015**,		18
121	A trajectory-based strict semantics for program slicing. *Theoretical Computer Science*, **2010**, 411, 1372-1386	0.6	18
120	Automated Session Data Repair for Web Application Regression Testing **2008**,		18
119	Less is More: Temporal Fault Predictive Performance over Multiple Hadoop Releases. *Lecture Notes in Computer Science*, **2014**, 240-246	0.9	18

118	Specialising Software for Different Downstream Applications Using Genetic Improvement and Code Transplantation. *IEEE Transactions on Software Engineering*, **2018**, 44, 574-594	3.5	17
117	Transformed Vargha-Delaney Effect Size. *Lecture Notes in Computer Science*, **2015**, 318-324	0.9	17
116	Finding the Optimal Balance between Over and Under Approximation of Models Inferred from Execution Logs **2012**,		17
115	A theoretical and empirical study of EFSM dependence **2009**,		17
114	Assessing the impact of global variables on program dependence and dependence clusters. *Journal of Systems and Software*, **2010**, 83, 96-107	3.3	17
113	The Effect of Communication Overhead on Software Maintenance Project Staffing: a Search-Based Approach **2007**,		17
112	. *IEEE Transactions on Software Engineering*, **2020**, 46, 302-320	3.5	17
111	Augmenting test suites effectiveness by increasing output diversity **2012**,		16
110	Model projection **2011**,		16
109	FlagRemover. *ACM Transactions on Software Engineering and Methodology*, **2011**, 20, 1-33	3.3	16
108	Memory mutation testing. *Information and Software Technology*, **2017**, 81, 97-111	3.4	15
107	Test oracle assessment and improvement **2016**,		15
106	Today/future importance analysis **2010**,		15
105	Evolutionary testing in the presence of loop-assigned flags. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **2004**, 29, 108-118	0.4	15
104	Testability Transformation □Program Transformation to Improve Testability **2008**, 320-344		15
103	An empirical study on dependence clusters for effort-aware fault-proneness prediction **2016**,		14
102	Amorphous Slicing of Extended Finite State Machines. *IEEE Transactions on Software Engineering*, **2013**, 39, 892-909	3.5	14
101	Theory and algorithms for slicing unstructured programs. *Information and Software Technology*, **2006**, 48, 549-565	3.4	14

100	Automated Transplantation of Call Graph and Layout Features into Kate. *Lecture Notes in Computer Science*, **2015**, 262-268	0.9	14
99	App Store Effects on Software Engineering Practices. *IEEE Transactions on Software Engineering*, **2021**, 47, 300-319	3.5	14
98	Crawlability metrics for automated web testing. *International Journal on Software Tools for Technology Transfer*, **2011**, 13, 131-149	1.3	13
97	Allowing Overlapping Boundaries in Source Code using a Search Based Approach to Concept Binding **2006**,		13
96	Syntax-Directed Amorphous Slicing. *Automated Software Engineering*, **2004**, 11, 27-61	1.5	13
95	Static Program Slicing Algorithms are Minimal for Free Liberal Program Schemas. *Computer Journal*, **2005**, 48, 737-748	1.3	13
94	Automatic testing and improvement of machine translation **2020**,		13
93	Customer Rating Reactions Can Be Predicted Purely using App Features **2018**,		13
92	The Value of Exact Analysis in Requirements Selection. *IEEE Transactions on Software Engineering*, **2017**, 43, 580-596	3.5	12
91	Empirical Study on the Efficiency of Search Based Test Generation for EFSM Models **2010**,		12
90	Refactoring as Testability Transformation **2011**,		12
89	. *IEEE Transactions on Software Engineering*, **2019**, 45, 1150-1169	3.5	11
88	ORBS and the limits of static slicing **2015**,		11
87	Dependence Anti Patterns **2008**,		11
86	Unifying program slicing and concept assignment for higher-level executable source code extraction. *Software - Practice and Experience*, **2005**, 35, 977-1006	2.5	11
85	Software engineering using metaheuristic innovative algorithms: workshop report. *Information and Software Technology*, **2001**, 43, 905-907	3.4	11
84	The SEMINAL workshop. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **2001**, 26, 62-66	0.4	11
83	An experimental search-based approach to cohesion metric evaluation. *Empirical Software Engineering*, **2017**, 22, 292-329	3.3	10

82	Identifying 'Linchpin Vertices' That Cause Large Dependence Clusters **2009**,		10
81	Equivalence of conservative, free, linear program schemas is decidable. *Theoretical Computer Science*, **2003**, 290, 831-862	1.1	10
80	ConSUS: a light-weight program conditioner. *Journal of Systems and Software*, **2005**, 77, 241-262	3.3	10
79	Grow and Serve: Growing Django Citation Services Using SBSE. *Lecture Notes in Computer Science*, **2015**, 269-275	0.9	10
78	App store mining and analysis **2015**,		9
77	Issues in clone classification for dataflow languages **2010**,		9
76	Transition coverage testing for simulink/stateflow models using messy genetic algorithms **2011**,		9
75	Measuring and Improving Latency to Avoid Test Suite Wear Out **2009**,		9
74	Generating feasible input sequences for extended finite state machines (EFSMs) using genetic algorithms **2005**,		9
73	Slicing programs in the presence of errors. *Formal Aspects of Computing*, **1996**, 8, 490-497	1.2	9
72	Search Based Transformations. *Lecture Notes in Computer Science*, **2003**, 2511-2512	0.9	9
71	Inferring Test Models from Kate's Bug Reports Using Multi-objective Search. *Lecture Notes in Computer Science*, **2015**, 301-307	0.9	8
70	Generalized observational slicing for tree-represented modelling languages **2017**,		8
69	The executable experimental template pattern for the systematic comparison of metaheuristics **2014**,		8
68	Coherent clusters in source code. *Journal of Systems and Software*, **2014**, 88, 1-24	3.3	8
67	Heuristics for fault diagnosis when testing from finite state machines. *Software Testing Verification and Reliability*, **2007**, 17, 41-57	0.9	8
66	An empirical study of the relationship between the concepts expressed in source code and dependence. *Journal of Systems and Software*, **2008**, 81, 2287-2298	3.3	8
65	Locating dependence structures using search-based slicing. *Information and Software Technology*, **2008**, 50, 1189-1209	3.4	8

64	Improving test quality using robust unique input/output circuit sequences (UIOCs). *Information and Software Technology*, **2006**, 48, 696-707	3.4	8
63	Mutation testing of memory-related operators **2015**,		7
62	Search--based approaches to the component selection and prioritization problem **2006**,		7
61	A formal relationship between program slicing and partial evaluation. *Formal Aspects of Computing*, **2006**, 18, 103-119	1.2	7
60	FITTEST: A new continuous and automated testing process for future Internet applications **2014**,		6
59	Equivalence hypothesis testing in experimental software engineering. *Software Quality Journal*, **2014**, 22, 215-238	1.2	6
58	Equivalence of linear, free, liberal, structured program schemas is decidable in polynomial time. *Theoretical Computer Science*, **2007**, 373, 1-18	1.1	6
57	**2006**,		6
56	An Empirical Study of Executable Concept Slice Size **2006**,		6
55	API-Constrained Genetic Improvement. *Lecture Notes in Computer Science*, **2016**, 224-230	0.9	6
54	Testing Web Enabled Simulation at Scale Using Metamorphic Testing **2021**,		6
53	Automated search for good coverage criteria **2016**,		6
52	Search Based Software Engineering. *Lecture Notes in Computer Science*, **2006**, 740-747	0.9	6
51	Genetic Improvement using Higher Order Mutation **2015**,		5
50	We Need a Testability Transformation Semantics. *Lecture Notes in Computer Science*, **2018**, 3-17	0.9	5
49	Dynamic adaptive Search Based Software Engineering needs fast approximate metrics (keynote) **2013**,		5
48	Inferring Automatic Test Oracles **2017**,		5
47	Analysis of Procedure Splitability **2008**,		5

46	Espresso **2000**,		5
45	Input Sequence Generation for Testing of Communicating Finite State Machines (CFSMs). *Lecture Notes in Computer Science*, **2004**, 1429-1430	0.9	5
44	Evaluation of estimation models using the Minimum Interval of Equivalence. *Applied Soft Computing Journal*, **2016**, 49, 956-967	7.5	5
43	Comparative Analysis of Constraint Handling Techniques for Constrained Combinatorial Testing. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	5
42	Are mutants really natural? **2018**,		5
41	Automated generation of state abstraction functions using data invariant inference **2013**,		4
40	Multi-objective Module Clustering for Kate. *Lecture Notes in Computer Science*, **2015**, 282-288	0.9	4
39	Coherent dependence clusters **2010**,		4
38	Multi objective higher order mutation testing with GP **2009**,		4
37	A non-standard semantics for program slicing and dependence analysis. *The Journal of Logic and Algebraic Programming*, **2007**, 72, 191-206		4
36	Characterising, Explaining, and Exploiting the Approximate Nature of Static Analysis through Animation **2006**,		4
35	Guaranteed inconsistency avoidance during software evolution. *Journal of Software: Evolution and Process*, **2003**, 15, 393-416		4
34	WES **2020**,		4
33	Testing of Future Internet Applications Running in the Cloud. *Advances in Computer and Electrical Engineering Book Series*,305-321	0.3	4
32	HOMI: Searching Higher Order Mutants for Software Improvement. *Lecture Notes in Computer Science*, **2016**, 18-33	0.9	4
31	"Ignorance and Prejudice" in Software Fairness **2021**,		4
30	. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	4
29	A Survey of Performance Optimization for Mobile Applications. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	4

28	GI4GI **2015**,		3
27	Regression Test Case Prioritisation for Guava. *Lecture Notes in Computer Science*, **2015**, 221-227	0.9	3
26	Empirical answers to fundamental software engineering problems (panel) **2013**,		3
25	Future Internet Testing with FITTEST **2011**,		3
24	A new algorithm for slicing unstructured programs **1998**, 10, 415		3
23	Some challenges for software testing research (invited talk paper) **2019**,		2
22	Crawlability Metrics for Web Applications **2012**,		2
21	Improving Web Application Testing using testability measures **2009**,		2
20	An alternative characterization of weak order dependence. *Information Processing Letters*, **2010**, 110, 939-943	0.8	2
19	Evaluating Key Statements Analysis **2008**,		2
18	Cost measures matter for mutation testing study validity **2020**,		2
17	Using Genetic Algorithms to Search for Key Stakeholders in Large-Scale Software Projects **2013**, 118-134		2
16	Optimised Realistic Test Input Generation Using Web Services. *Lecture Notes in Computer Science*, **2012**, 105-120	0.9	2
15	Agent-Based Modelling of Stock Markets Using Existing Order Book Data. *Lecture Notes in Computer Science*, **2013**, 101-114	0.9	2
14	A Study of Bug Resolution Characteristics in Popular Programming Languages. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	2
13	Learning From Mistakes: Machine Learning Enhanced Human Expert Effort Estimates. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	2
12	Enhancing Genetic Improvement of Software with Regression Test Selection **2021**,		2
11	Facebook's Cyber□Cyber and Cyber□Physical Digital Twins **2021**,		2

10	OASIs: oracle assessment and improvement tool **2018**,		2
9	Efficient Identification of Linchpin Vertices in Dependence Clusters. *ACM Transactions on Programming Languages and Systems*, **2013**, 35, 1-35	1.6	1
8	1st International workshop on combining modelling and search-based software engineering (CMSBSE 2013) **2013**,		1
7	SBSelector: Search Based Component Selection for Budget Hardware. *Lecture Notes in Computer Science*, **2015**, 289-294	0.9	1
6	The FITTEST Tool Suite for Testing Future Internet Applications. *Lecture Notes in Computer Science*, **2014**, 1-31	0.9	1
5	Fairea: a model behaviour mutation approach to benchmarking bias mitigation methods **2021**,		1
4	Foreword to the invited impact paper on automatic software repair. *Software Quality Journal*, **2013**, 21, 419-419	1.2	
3	TAIC PART 2007 and Mutation 2007 special issue editorial. *Journal of Systems and Software*, **2009**, 82, 1753-1754	3.3	
2	Source code analysis and manipulation. *Information and Software Technology*, **2002**, 44, 717-720	3.4	
1	The FITTEST Tool Suite for Testing Future Internet Applications. *Lecture Notes in Computer Science*, **2014**, 1-31	0.9	