All citations are in BibTex format. Comments are BELOW each link.

@incollection {springerlink:10.1007/0-387-28356-0_10,
  author = {Deb, Kalyanmoy},
  affiliation = {Indian Institute of Technology Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering Kanpur India},
  title = {Multi-Objective Optimization},
  booktitle = {Search Methodologies},
  editor = {Burke, Edmund K. and Kendall, Graham},
  publisher = {Springer US},
  isbn = {978-0-387-28356-2},
  keyword = {Mathematics and Statistics},
  pages = {273-316},
  url = {http://dx.doi.org/10.1007/0-387-28356-0_10},
  note = {10.1007/0-387-28356-0_10},
  year = {2005}
}

%1

%The most primary source of Multi-Objective Optimization. This is a chapter in the Book Search Methodologies. Outlines the nature of multi-objective optimization broadly with much information on the nature of Pareto dominance

@inproceedings{Harman:2007:CSF:1253532.1254729,
  author = {Harman, Mark},
  title = {The Current State and Future of Search Based Software Engineering},
  booktitle = {2007 Future of Software Engineering},
  series = {FOSE '07},
  year = {2007},
  isbn = {0-7695-2829-5},
  pages = {342--357},
  numpages = {16},
  url = {http://dx.doi.org/10.1109/FOSE.2007.29},
  doi = {10.1109/FOSE.2007.29},
  acmid = {1254729},
  publisher = {IEEE Computer Society},
}
This paper surveys the current state of search-based software engineering, which tries to improve software engineering by empowering users to understand the trade-offs inherent in software engineering. Surveys multiple ways to go about MOOP (Multi-objective optimization problems) using various techniques.

@ARTICLE{Deb00afast,  
author = {Kalyanmoy Deb and Amrit Pratap and Sameer Agarwal and T. Meyarivan},  
title = {A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II},  
journal = {IEEE Transactions on Evolutionary Computation},  
year = {2000},  
volume = {6},  
pages = {182--197}
}

Proposes NSGA-II, an $n^2$ order elitest genetic algorithm which improves upon NSGA by adding elitism and reducing the runtime. NSGA-II has become a de facto comparison in many MOOP algorithms. It uses Pareto non-dominance sorting with GA style combination/mutation, etc.

@ARTICLE{4455350,  
author={Nebro, A. J. and Luna, F. and Alba, E. and Dorronsoro, B. and Durillo, J. J. and Beham, A.},  
journal={Evolutionary Computation, IEEE Transactions on},  
title={AbYSS: Adapting Scatter Search to Multiobjective Optimization},  
year={2008},  
month={aug. },  
volume={12},  
number={4},  
pages={439 -457},  
keywords={},  
doi={10.1109/TEVC.2007.913109},
Proposes and new algorithm (AbYSS) to solve MOOPs. It uses scatter search methodology in multiobjective space, and compares its results to the state of the art algorithms NSGA-II and SPEA2.

Applies SPEA-2 (another MOOP problem) to a real world problem to find optimal waveforms for a single platform radar system performing multiple radar tasks concurrently. This paper shows an excellent example of defining multiple objectives in a real world problem and using MO optimizers to solve them.

@MISC{Zitzler02spea2:improving,  
  author = {E. Zitzler and K. Giannakoglou and D. Tsahalisi and J. Periaux and K. Papailiou and T. Fogarty (eds and Eckart Zit. Ler and Marco Laumanns and Lothar Thiele)},  
  title = {SPEA2: Improving the Strength Pareto}
The paper cited by #6 which lays out SPEA2. SPEA2 improves upon the oft cited (before the publication of SPEA2) SPEA, which adds in a fitness assignment strategy, density estimation, and manages a better past fit list (a truncated archive of previous solutions). Results are compared with PESA and NSGA-II.

Proposes PESA-II: another MO Optimizer. In this, hyperboxes in objective (or Pareto) space are tested for fitness, rather than individual solutions. This allows for a better spread along the Pareto frontier than individual selection.
Previous to this paper, most objective vectors were scalarized, which creates a set of solutions that tend to bias to particular reasons objective space. Deb & Srinivas show previously used test cases for a proof of concept for GAs including by adding in non-dominating sorting. They also show possible higher dimensionality usability for this.

@article {springerlink:10.1007/s00158-003-0368-6,  
  author = {Marler, R.T. and Arora, J.S.},  
  affiliation = {Optimal Design Laboratory, College of Engineering The University of Iowa 52242 Iowa City USA},  
  title = {Survey of multi-objective optimization methods for engineering},  
  journal = {Structural and Multidisciplinary Optimization},  
  publisher = {Springer Berlin / Heidelberg},  
  issn = {1615-147X},  
  keyword = {Engineering},  
  pages = {369-395},  
  volume = {26},  
  issue = {6},  
  url = {http://dx.doi.org/10.1007/s00158-003-0368-6},  
  note = {10.1007/s00158-003-0368-6},  
  year = {2004}  
}

Survey current multi-object optimization methodologies. Surveys several GAs as well as other MOO. Each individual method is explained with pros and cons as well as detailed explanations of their basic concepts. Ultimately no single approach found "best."

@ARTICLE{6081882,  
  author={Kocaguneli, E. and Menzies, T. and Keung, J.},  
  journal={Software Engineering, IEEE Transactions on},  
  title={On the Value of Ensemble Effort Estimation},  
  year={2011},  
  month={}  
}
No effort estimation (a MOOP) provides the most accurate models consistently. By combining 9 learning algorithms with 10 pre-processors, 90 solo-methods were generated. These were then tested across 20 effort estimation datasets and evaluated on 7 error metrics. This paper shows some combinations are better than others, even if there is no definitive best.

Defect prediction is an MOOP problem where you try to optimize metrics PD (closer to 1), PF (closer to 0). This tries to turn that into a single-dimension problem by using AUC (area under the optimal curve) as a metric. Proposes an algorithm WHICH (a 1-dimensional optimizer) which
I upgraded to multi-dimension for my master's these (which I called HOW). This uses static code features which can be relatively quickly gleaned from a file compared to less objective metrics.

@article{Fenton:1999:CSD:325392.325401,  
author = {Fenton, Norman E. and Neil, Martin},  
title = {A Critique of Software Defect Prediction Models},  
journal = {IEEE Trans. Softw. Eng.},  
issue_date = {September 1999},  
volume = {25},  
number = {5},  
month = sep,  
year = {1999},  
issn = {0098-5589},  
pages = {675--689},  
numpages = {15},  
url = {http://dx.doi.org/10.1109/32.815326},  
doi = {10.1109/32.815326},  
acmid = {325401},  
publisher = {IEEE Press},  
address = {Piscataway, NJ, USA},  
keywords = {Software faults and failures, defects, complexity metrics, fault-density, Bayesian Belief Networks.},
}

%Reference as a counterpoint in 12, Fenton very much argues against using static code features, since finding defects doesn't necessarily link to failures in a product. He reviews various attempts to learn from static code features, and points out the mistakes in them.

@article{Gay:2010:FRS:1670710.1670752,  
author = {Gay, Gregory and Menzies, Tim and Jalali, Omid and Mundy, Gregory and Gilkerson, Beau and Feather, Martin and Kiper, James},  
title = {Finding robust solutions in requirements models},  
journal = {Automated Software Engg.},  
issue_date = {March 2010},  
volume = {17},  
number = {1},  
month = mar,
This paper can illustrate how optimization solutions can be brittle, in that they may not have widespread efficacy due to slight changes from area to area. This tries to address this by establishing a robustness metric for requirements engineering (another MOOP).

@ARTICLE{cost-value,
author={Karlsson, J. and Ryan, K.},
journal={Software, IEEE}, title={A cost-value approach for prioritizing requirements},
year={1997},
month={sep/oct},
volume={14},
number={5},
pages={67-74},
keywords={commercial projects;competitive edge;cost-value approach;customer needs;project managers;requirement prioritizing;software development;software provider;user needs;cost-benefit analysis;formal specification;software cost estimation;software development management;systems analysis;},
doi={10.1109/52.605933},
ISSN={0740-7459},}

Getting back away from defect prediction, this article focuses on requirements engineering for the "next-release" problem. This paper shows how requirements engineering is multi-objective, and using a bi-objective model of optimizing value while minimizing cost. This divides areas into regions, although ultimately the algorithm
suggested is an aggregate fitness function rather than a Pareto optimal space finding solution.

@inproceedings{nextrelease, 
  author = {Zhang, Yuanyuan and Harman, Mark and Mansouri, S. Afshin}, 
  title = {The multi-objective next release problem}, 
  booktitle = {Proceedings of the 9th annual conference on Genetic and evolutionary computation}, 
  series = {GECCO '07}, 
  year = {2007}, 
  isbn = {978-1-59593-697-4}, 
  location = {London, England}, 
  pages = {1129--1137}, 
  numpages = {9}, 
  url = {http://doi.acm.org/10.1145/1276958.1277179}, 
  doi = {10.1145/1276958.1277179}, 
  acmid = {1277179}, 
  publisher = {ACM}, 
  address = {New York, NY, USA}, 
  keywords = {multi-objective genetic algorithms, next release problem, pareto optimality}, 
}

This paper shows an excellent example of how to breakdown a software engineering problem (specifically, next-release problem) into mathematic constructs with can be used find a Pareto optimal front. This specifically uses NSGA-II to solve the case and compares the results to a single-objective genetic algorithm. Multi-objective finds more solutions along the "important" part of the trade space (where the trade-offs are more visible). Single objective performs better on the extremes, though these aren't as useful.

@inproceedings{Veerappa, 
  author = {Veerappa, V. and Letier, E.}, 
  booktitle = {Requirements Engineering Conference (RE), 2011 19th IEEE International}, 
  title = {Understanding clusters of optimal solutions in multi-objective decision problems}, 
}
Connects multi-objective optimization input and output. In Pareto optimization, very different inputs may lead to similar outputs and vice versa. This would make selecting policies from a multi-objective optimizers to implement is complicated. By clustering input, we can understand the output better. Specifically works in requirements engineering.

@INPROCEEDINGS{Tanaka,
author={Tanaka, M. and Watanabe, H. and Furukawa, Y. and Tanino, T.},
title={GA-based decision support system for multicriteria optimization},
year={1995},
month={oct},
volume={2},
number={},
pages={1556 -1561 vol.2},
keywords={ GA-based decision support system; Kohonen's self organizing map; Pareto optimal solutions; final decision; genetic algorithm; multicriteria decision making; multicriteria optimization; radial basis function network; rough set; decision support systems; decision theory; feedforward neural nets; genetic algorithms; optimisation; self-organising feature maps;},
doi={10.1109/ICSMC.1995.537993},}
MOO (Multi-objective optimization) referenced as a decision making process (MCDM) using a genetic algorithm. This ends up proposing a very popular MOOP problem which finds the variables that best optimize a space. This frontier is non-continuous non-convex space. It was found using evolutionary GAs.

@article{zdt,
author = {Zitzler, Eckart and Deb, Kalyanmoy and Thiele, Lothar},
title = {Comparison of Multiobjective Evolutionary Algorithms: Empirical Results},
journal = {Evol. Comput.},
issue_date = {June 2000},
volume = {8},
number = {2},
month = jun,
year = {2000},
issn = {1063-6560},
pages = {173--195},
numpages = {23},
url = {http://dx.doi.org/10.1162/106365600568202},
doi = {10.1162/106365600568202},
acmid = {1108876},
publisher = {MIT Press},
address = {Cambridge, MA, USA},
}

A comparison of multiple evolutionary MOO using six proposed test functions (which have become very common tests). Also shows that test problems provide sufficient complexity to mimic real world efficacy of MOO. Shows Elitism is important for evolutionary MOOs.

@INPROCEEDINGS{kursawe,
author = {Frank Kursawe},
title = {A Variant of Evolution Strategies for Vector Optimization},
booktitle = {Parallel Problem Solving from Nature. 1st Workshop, PPSN I},
volume = {496 of}
An early example of a MOOP developed for Pareto optimization. Points out shortcomings of single dimension optimization and shows evolutionary strategies and illustrates Pareto Optimization on complex mathematical model.

@ARTICLE{fonseca,  
author={Fonseca, C.M. and Fleming, P.J.},  
title={Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. Application example},  
year={1998},  
month={jan},  
volume={28},  
number={1},  
pages={38 -47},  
keywords={Pegasus gas turbine engine;designer-oriented formulation;evolutionary algorithms;genetic algorithm;low-pressure spool speed governor;multiobjective optimization;multiple constraint handling.preference articulation.aerospace engines;decision theory;gas turbines;genetic algorithms;velocity control;},  
doi={10.1109/3468.650320},  
ISSN={1083-4427},}

Explores the affect of constraints on MOOs. Constraints must be handled within the optimizer and can't be separated from finding the best solutions. Evolutionary Search Algorithms essentially act as unconstrained optimizers, and constraint handling must be taken into account for each test.

@INPROCEEDINGS{comparison2,  
author={Godí andnez, A.C. and Espinosa, L.E.M.}
and Montes, E.M.},
booktitle={Electronics, Robotics and Automotive
Mechanics Conference (CERMA), 2010}, title={An
Experimental Comparison of Multiobjective
Algorithms: NSGA-II and OMOPSO},
year={2010},
month={28 2010-oct. 1},
volume={},
number={},
pages={28 -33},
keywords={MOPS;NSGA-II;Pareto front;ZDT test
functions;metaheuristics;multi objective particle
swarm optimization;research and
development;sorting genetic algorithm no
dominated-II;Pareto optimisation;genetic
algorithms;particle swarm optimisation;research
and development;},
doi={10.1109/CERMA.2010.13},
ISSN={},

%22

%This paper illustrates how to compare two MOO
tools. It does so using visual graphic
 comparision of results. Additionally numeric
comparisons are used such as adherence to the
Pareto frotier (convergence) spread across the
frontier (dispersion). The two tools were NSGA-II
and OMOPSO. This study claims OMOPSO performs
better than NSGA-II

@INPROCEEDINGS{heaven,
author={Heaven, W. and Letier, E.},
booktitle={Requirements Engineering Conference
(RE), 2011 19th IEEE International},
title={Simulating and optimising design decisions
in quantitative goal models},
year={2011},
month={29 2011-sept. 2},
volume={},
number={},
pages={79 -88},
keywords={ambulance service system;automated
techniques;multiobjective optimisation
algorithm;quantitative goal models;requirements
engineering;formal specification;optimisation;},
doi={10.1109/RE.2011.6051653},
ISSN={1090-705X},}
Using MOO principles to choose multiple design policies is the entire point of finding the Pareto frontier. Illustrating the Pareto Space is important to inform how the trade-offs of different client goals interact. Analysing all alternatives is expensive, and at the time this paper was written, dedicated tools didn't exist. MOO was used to study techniques by the London Ambulance service, using a simulation prototype. Showed a need to connect input space to output space.

@INPROCEEDINGS{comparison1, 
author={Kukkonen, S. and Lampinen, J.}, 
year={2005}, 
month={sept.}, 
volume={1}, 
number={}, 
pages={443 -450 Vol.1}, 
keywords={GDE3;generalized differential evolution;global optimization;multiobjective problems;constraint theory;evolutionary computation;optimisation;}, 
doi={10.1109/CEC.2005.1554717}, 
ISSN={},}

This is another example of improving an old algorithm, proposing GDE3. It extends differential evolution to allow for better constraint handling in MOOP. GDE3 is shown to outperform previous GDE's and be comparable to NSGA-II on a limited number of tests.

@INPROCEEDINGS{algo1, 
author={Bo Liu and Fernandez, F.V. and Qingfu Zhang and Pak, M. and Sipahi, S. and Gielen, G.}, 
booktitle={Evolutionary Computation (CEC), 2010 IEEE Congress on}, title={An enhanced MOEA/D-DE and its application to multiobjective analog cell sizing}, 
year={2010},}
Enhances an existing MOO. MOEA/D-DE combines the use of decomposition for evolutionary searching with differential evolution. This shows how to test on multiple benchmark problems and is then tested on several real-world problems. This uses a new replacement mechanism to find better children than the original random selection method. Scaling for DE is also randomized to enhance searching.