BMOBench: Black-Box Multi-Objective Optimization Benchmarking Platform

Abdullah Al-Dujaili, Suresh Sundaram

May 14, 2018

This document briefly describes the Black-Box Multi-Objective Optimization Benchmarking (**BMOBench**) platform. It presents the test problems, evaluation procedure, and experimental setup. To this end, the BMOBench is demonstrated by comparing recent multi-objective solvers from the literature, namely `SMS-EMOA` (Beume et al., 2007), `DMS` (Custódio et al., 2011), and `MO-SOO` (Al-Dujaili and Suresh, 2015).

1 Test Problems

One-hundred multi-objective optimization problems from the literature are selected.[1] These problems have simple bound constraints, that is to say, $\mathcal{X} = [\mathbf{l}, \mathbf{u}] \subset \mathbb{R}^n$, where $\mathbf{u} \succeq \mathbf{l}$. Table 1 presents a brief list of these problems with number of dimensions/objectives. In order to have a better understanding of the algorithm strength/weakness, the benchmark problems are categorized (wherever possible) according to three key characteristics, namely *dimensionality*: low- or high-dimension decision space, *separability*: separable or non-separable objectives, and *modality*: uni-modal or multi-modal objectives. Each of these attributes imposes a different challenge in solving an MOO problem (Huband et al., 2006).

2 Evaluation Budget

`MO-SOO` is a deterministic algorithm producing the same approximation set in each run of the algorithm for a given problem, whereas the approximation sets produced by the compared stochastic algorithms: `DMS` and `SMS-EMOA` can be different every time they are run for a given problem. In practice, stochastic algorithms are run several times per problem. To this end and to ensure a fair comparison, given a computational budget of v function evaluations per run, the stochastic algorithms are allocated 10 runs per problem instance. On the other hand, the deterministic algorithms are run once per problem instance with the accumulated $10 \times v$ function evaluations.

In our experiments, the evaluation budget v is made proportional to the search space dimension n and is set to $10^2 \cdot n$. The overall computational budget used by an algorithm on BMOBench is the product of the evaluation budget per run, the number of problems, and the number of runs per problem.

[1] retrieved from http://www.mat.uc.pt/dms.

#	Problem Name	n	m	**D**	**S**	**M**	#	Problem Name	n	m	**D**	**S**	**M**	#	Problem Name	n	m	**D**	**S**	**M**
1	BK1 (Huband et al., 2006)	2	2	L	S	U	35	I5 (Huband et al., 2005)	8	3	H	NS	U	68	MOP3 (Huband et al., 2006)	2	2	L	×	×
2	CL1 (Cheng and Li, 1999)	4	2	L	×	×	36	IKK1 (Huband et al., 2006)	2	3	L	×	U	69	MOP4 (Huband et al., 2006)	3	2	L	S	×
3	Deb41 (Deb, 1999)	2	2	L	×	×	37	IM1 (Huband et al., 2006)	2	2	L	×	U	70	MOP5 (Huband et al., 2006)	2	3	L	NS	×
4	Deb512a (Deb, 1999)	2	2	L	×	×	38	Jin1 (Vicente and Custódio, 2012)	2	2	L	×	U	71	MOP6 (Huband et al., 2006)	2	2	L	S	×
5	Deb512b (Deb, 1999)	2	2	L	×	×	39	Jin2 (Vicente and Custódio, 2012)	2	2	L	×	U	72	MOP7 (Huband et al., 2006)	2	3	L	×	U
6	Deb512c (Deb, 1999)	2	2	L	×	×	40	Jin3 (Vicente and Custódio, 2012)	2	2	L	×	U	73	OKA1 (Nocedal and Wright, 2006)	2	2	L	×	×
7	Deb513 (Deb, 1999)	2	2	L	×	×	41	Jin4 (Vicente and Custódio, 2012)	2	2	L	×	U	74	OKA2 (Nocedal and Wright, 2006)	3	2	L	×	×
8	Deb521a (Deb, 1999)	2	2	L	×	×	42	Kursawe (Kolda et al., 2003)	3	2	L	×	×	75	QV1 (Huband et al., 2006)	10	2	H	S	M
9	Deb521b (Deb, 1999)	2	2	L	×	×	43	L1ZDT4 (Deb et al., 2006)	10	2	H	×	×	76	Sch1 (Huband et al., 2006)	1	2	L	×	×
10	Deb53 (Deb, 1999)	2	2	L	×	×	44	L2ZDT1 (Deb et al., 2006)	30	2	H	×	×	77	SK1 (Huband et al., 2006)	1	2	L	S	M
11	DG01 (Huband et al., 2006)	1	2	L	×	M	45	L2ZDT2 (Deb et al., 2006)	30	2	H	×	×	78	SK2 (Huband et al., 2006)	4	2	L	×	×
12	DPAM1 (Huband et al., 2006)	10	2	H	NS	×	46	L2ZDT3 (Deb et al., 2006)	30	2	H	×	×	79	SP1 (Huband et al., 2006)	2	2	L	NS	U
13	DTLZ1 (Deb et al., 2002)	7	3	H	×	M	47	L2ZDT4 (Deb et al., 2006)	30	2	H	×	×	80	SSFYY1 (Huband et al., 2006)	2	2	L	S	U
14	DTLZ1n2 (Deb et al., 2002)	2	2	L	×	M	48	L2ZDT6 (Deb et al., 2006)	10	2	H	×	×	81	SSFYY2 (Huband et al., 2006)	1	2	L	×	×
15	DTLZ2 (Deb et al., 2002)	12	3	H	×	U	49	L3ZDT1 (Deb et al., 2006)	30	2	H	×	×	82	TKLY1 (Huband et al., 2006)	4	2	L	×	×
16	DTLZ2n2 (Deb et al., 2002)	2	2	L	×	U	50	L3ZDT2 (Deb et al., 2006)	30	2	H	×	×	83	VFM1 (Huband et al., 2006)	2	3	L	S	U
17	DTLZ3 (Deb et al., 2002)	12	3	H	×	M	51	L3ZDT3 (Deb et al., 2006)	30	2	H	×	×	84	VU1 (Huband et al., 2006)	2	2	L	S	U
18	DTLZ3n2 (Deb et al., 2002)	2	2	L	×	M	52	L3ZDT4 (Deb et al., 2006)	30	2	H	×	×	85	VU2 (Huband et al., 2006)	2	2	L	S	U
19	DTLZ4 (Deb et al., 2002)	12	3	H	×	U	53	L3ZDT6 (Deb et al., 2006)	10	2	H	×	×	86	WFG1 (Huband et al., 2006)	8	3	H	S	U
20	DTLZ4n2 (Deb et al., 2002)	2	2	L	×	U	54	LE1 (Huband et al., 2006)	2	2	L	S	U	87	WFG2 (Huband et al., 2006)	8	3	H	NS	×
21	DTLZ5 (Deb et al., 2002)	12	3	H	×	U	55	lovison1 (Liuzzi et al., 2003)	2	2	L	×	×	88	WFG3 (Huband et al., 2006)	8	3	H	NS	U
22	DTLZ5n2 (Deb et al., 2002)	2	2	L	×	U	56	lovison2 (Liuzzi et al., 2003)	2	2	L	×	×	89	WFG4 (Huband et al., 2006)	8	3	H	S	M
23	DTLZ6 (Deb et al., 2002)	22	3	H	×	U	57	lovison3 (Liuzzi et al., 2003)	2	2	L	×	×	90	WFG5 (Huband et al., 2006)	8	3	H	S	×
24	DTLZ6n2 (Deb et al., 2002)	2	2	L	×	U	58	lovison4 (Liuzzi et al., 2003)	2	2	L	×	×	91	WFG6 (Huband et al., 2006)	8	3	H	NS	U
25	ex005 (Hwang and Masud, 1979)	2	2	L	×	U	59	lovison5 (Liuzzi et al., 2003)	3	3	L	×	×	92	WFG7 (Huband et al., 2006)	8	3	H	S	U
26	Far1 (Huband et al., 2006)	2	2	L	NS	M	60	lovison6 (Liuzzi et al., 2003)	3	3	L	×	×	93	WFG8 (Huband et al., 2006)	8	3	H	NS	U
27	FES1 (Huband et al., 2006)	10	2	H	S	U	61	LRS1 (Huband et al., 2006)	2	2	L	S	U	94	WFG9 (Huband et al., 2006)	8	3	H	NS	×
28	FES2 (Huband et al., 2006)	10	3	H	S	U	62	MHHM1 (Huband et al., 2006)	1	3	L	×	U	95	ZDT1 (Zitzler et al., 2000)	30	2	H	S	U
29	FES3 (Huband et al., 2006)	10	4	H	S	U	63	MHHM2 (Huband et al., 2006)	2	3	L	S	U	96	ZDT2 (Zitzler et al., 2000)	30	2	H	S	U
30	Fonseca (Fonseca and Fleming, 1998)	2	2	L	S	U	64	MLF1 (Huband et al., 2006)	1	2	L	×	M	97	ZDT3 (Zitzler et al., 2000)	30	2	H	S	×
31	I1 (Huband et al., 2005)	8	3	H	S	U	65	MLF2 (Huband et al., 2006)	2	2	L	NS	M	98	ZDT4 (Zitzler et al., 2000)	10	2	H	S	×
32	I2 (Huband et al., 2005)	8	3	H	NS	U	66	MOP1 (Huband et al., 2006)	1	2	L	S	U	99	ZDT6 (Zitzler et al., 2000)	10	2	H	S	M
33	I3 (Huband et al., 2005)	8	3	H	NS	U	67	MOP2 (Huband et al., 2006)	4	2	L	S	U	100	ZLT1 (Huband et al., 2006)	10	3	H	S	U
34	I4 (Huband et al., 2005)	8	3	H	NS	U														

Table 1: Test problems definition and properties. Symbols: **D** : dimensionality $\in$ {L : low-dimensionality, H : high-dimensionality}; **S** : separability $\in$ {S : separable, NS : non-separable}; **M** : modality $\in$ {U : uni-modal, M : multi-modal}; $\times$: uncategorized/mixed.

Quality Indicator (I)	Pareto-Compliant	Reference Set Required	Target
Hypervolume Difference (I_H^-)	Yes	Yes	Minimize
Generational Distance (I_{GD})	No	Yes	Minimize
Inverted Generational Distance (I_{IGD})	No	Yes	Minimize
Additive ϵ-Indicator ($I^1_{\epsilon+}$)	Yes	Yes	Minimize

Table 2: Employed Quality Indicators. Adapted from (Hadka, 2012) (for more details, see Knowles and Corne, 2002; Coello et al., 2002).

With $n = 2$, for instance, the overall computational budget used by `MO-SOO` on BMOBench is $10^3 \cdot 2 \cdot 100 \cdot 1 = 2 \times 10^5$ function evaluations. Each of the other algorithms uses also a computational budget of $10^2 \cdot 2 \cdot 100 \cdot 10 = 2 \times 10^5$ function evaluations.

3 Benchmark Procedure

Similar to (Brockhoff et al., 2015), a set of targets are defined for each problem in terms of four popular quality indicators (Knowles et al., 2006; Zitzler et al., 2003) listed in Table 2. A solver (algorithm) is then evaluated based on its runtime with respect to each target: the number of function evaluations used until the target is reached. We present the recorded runtime values in terms of *data profiles* (Moré and Wild, 2009). A data profile can be regarded as an empirical cumulative distribution function of the observed number of function evaluations in which the y-axis tells how many targets—over the set of problems and quality indicators—have been reached by each algorithm for a given evaluation budget (on the x-axis). Mathematically, a data profile for a solver s on a problem class

P has the form

$$d_s(\alpha) = \frac{1}{|P|} \left| \left\{ p \in P \;\middle|\; \frac{t_{p,s}}{n_p} \leq \alpha \right\} \right| ,$$

where $t_{p,s}$ is the observed runtime of solver s on solving problem p (hitting a target) over a decision space $\mathcal{X} \subseteq \mathbb{R}^{n_p}$. The data profile approach captures several benchmarking aspects, namely the convergence behavior over time rather than a fixed budget, which can as well be aggregated over problems of similar category (see, for more details, Brockhoff et al., 2015). In our experiments, 70 linearly spaced values in the logarithmic scale from $10^{-0.8}$ to 10^{-3} and from $10^{-0.1}$ to 10^{-2} were used as targets for I_H^-, I_{GD}, and I_{IGD}; and $I_{\epsilon+}^1$, respectively.

The I_H^-, I_{GD}, I_{IGD} and $I_{\epsilon+}^1$values are computed for each algorithm at any point of its run based on the set of all (normalized) non-dominated vectors found so far—i.e., the *archive*—with respect to a (normalized) *reference* set $R \in \Omega$. We computed the reference set for calculating the quality indicators by aggregating(union) the approximation sets generated by the evolutionary algorithms used in (Custódio et al., 2011).

As mentioned in the previous section, we aim to provide a fair comparison between deterministic and stochastic solvers and accommodate the multiple-run practice for stochastic algorithms, at the same time. This has been reflected in the evaluation budget allocation (see Section 2). Likewise, we need to adapt the data profiles. To this end, given a problem instance and for each one of the stochastic solvers, we consider the *best* reported runtime for each target from the solver's 10 runs, rather than the mean value. With this setting in hand, the data profile of `MO-SOO` at 10^3 function evaluations, for instance, can be compared to that of `SMS-EMOA` at 10^2 function evaluations.

4 Results

Figures 1 and 2 show the data profiles of the compared algorithms as a function of the number of functione evaluations used.

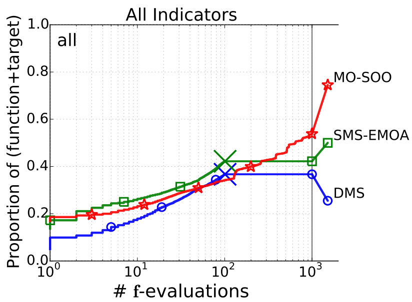

Figure 1: Data profiles aggregated over all the problems across all the quality indicators computed for each of the compared algorithms. The symbol $\times$ indicates the maximum number of function evaluations.

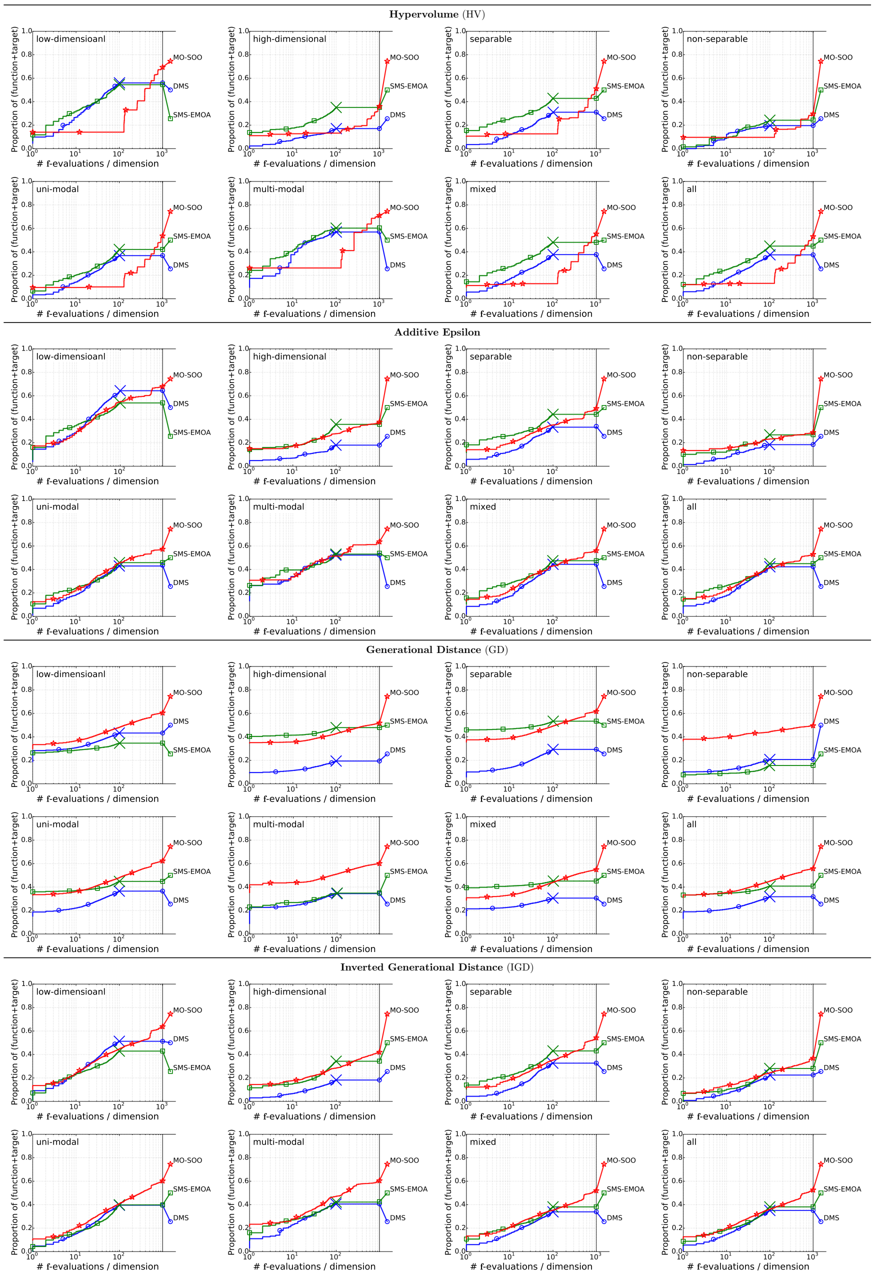

Figure 2: Data profiles aggregated over problem categories for each of the quality indicators computed. The symbol × indicates the maximum number of function evaluations.

5 Empirical Runtime Evaluation

In order to evaluate the complexity of the algorithms (measured in runtime), we have run the algorithms on a representative set of the problems. The empirical

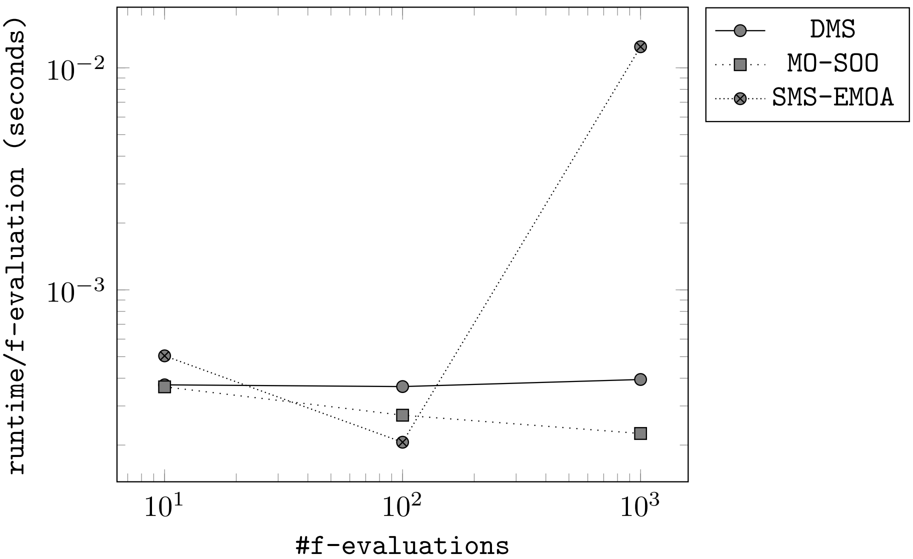

Figure 3: A *log*-log plot visualizing the runtime per one function evaluation (in seconds) of the compared algorithms. All the algorithms were run on a selected set of problems over a set of evaluation budgets, namely BK1, DPAM1, L3ZDT1, DTLZ3, and FES3; with an evaluation budget $\in \{10, 100, 1000\}$ per problem on a PC with: 64-bit Windows 7, Intel Xeon E5 CPU @ 3.20GHz, 16GB of memory.

complexity of an algorithm is then computed as the running time (in seconds) of the algorithm summed over all the problems given an evaluation budget (#FE). The results are shown in Figure 3.

References

Abdullah Al-Dujaili and S. Suresh. Multi-objective simultaneous optimistic optimization. *Manuscript submitted for publication to Journal of Machine Learning Research (JMLR)*, 2015.

Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

Dimo Brockhoff, Thanh-Do Tran, and Nikolaus Hansen. Benchmarking Numerical Multiobjective Optimizers Revisited. In *Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain, July 2015. doi: 10.1145/2739480.2754777. URL https://hal.inria.fr/hal-01146741.

FY Cheng and XS Li. Generalized center method for multiobjective engineering optimization. *Engineering Optimization*, 31(5):641–661, 1999.

Carlos A Coello Coello, David A Van Veldhuizen, and Gary B Lamont. *Evolutionary algorithms for solving multi-objective problems*, volume 242. Springer, 2002.

Ana Luísa Custódio, JF Aguilar Madeira, A Ismael F Vaz, and Luís N Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011.

Kalyanmoy Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation*, 7(3):205–230, 1999.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA)*, pages 825–830. Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA), 2002.

Kalyanmoy Deb, Ankur Sinha, and Saku Kukkonen. Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1141–1148. ACM, 2006.

Carlos M Fonseca and Peter J Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 28(1):26–37, 1998.

D Hadka. Moea framework a free and open source java framework for multiobjective optimization, 2012.

S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *Evolutionary Computation, IEEE Transactions on*, 10(5):477–506, Oct 2006. ISSN 1089-778X. doi: 10.1109/TEVC.2005.861417.

Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A scalable multi-objective test problem toolkit. In *Evolutionary multi-criterion optimization*, pages 280–295. Springer, 2005.

C.-L. Hwang and A. S. MD. Masud. Multiple objective decision making—methods and applications: A state-of-the-art survey. *Lecture Notes in Econom. Math. Systems*, 164, 1979. Springer-Verlag, Berlin.

J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 1, pages 711–716, May 2002. doi: 10.1109/CEC.2002.1007013.

J.D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multi-objective optimizers. TIK-Report 214, Computer Engineering and Networks Laboratory, ETH Zurich, Gloriastrasse 35, ETH-Zentrum, 8092 Zurich, Switzerland, February 2006.

Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.

Giampaolo Liuzzi, Stefano Lucidi, Francesco Parasiliti, and Marco Villani. Multiobjective optimization techniques for the design of induction motors. *IEEE Transactions on Magnetics*, 39(3):1261–1264, 2003.

Jorge J Moré and Stefan M Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

J. Nocedal and Stephen J Wright. *Numerical optimization*, volume 2. Springer-Verlag, Berlin, 2 edition, 2006.

Luís N Vicente and AL Custódio. Analysis of direct searches for discontinuous functions. *Mathematical programming*, 133(1-2):299–325, 2012.

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8 (2):173–195, 2000.

Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.