THE IMPACT OF QUALITY INDICATORS ON THE RATING OF MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Miha Ravber, Marjan Mernik, Matej Črepinšek
Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia
miha.ravber@um.si, marjan.mernik@um.si, matej.crepinsek@um.si

Abstract Comparing the results of single objective optimizers is an easy task in comparison to multi-objective optimizers for which the result is usually an approximation of the Pareto optimal front. These approximation sets must first be evaluated. One of the most popular methods for evaluation is the use of quality indicators, for which the result is a real valued number that reflects a certain aspect of quality. Evaluating and comparing multi-objective optimizers is an important issue. It has been empirically proven that chess ranking can be successfully applied to ranking and comparing single objective evolutionary algorithms. In this paper, the method was adapted to multi-objective evolutionary algorithms (MOEAs). The comparison of several different quality indicators in the chess rating system was conducted in order to get a better insight on their characteristics and how they affect the ranking of MOEAs. Although it is expected that quality indicators with the same optimization goals would yield a similar ranking of MOEAs, it has been shown that results can be contradictory.

Keywords: Chess rating, Evolutionary algorithms, Multi-objective optimization, Performance assessment, Quality indicator.

1. Introduction

The goal of multi-objective optimization (MOO) is to obtain the Pareto optimal front that contains the best trade-off solutions. Since many multi-objective optimization problems (MOP) are difficult to solve, the outcome of the optimization is usually an approximation of the Pareto front. In order to compare these approximations, they need to be evaluated. Evaluating the quality of these approximations is itself an MOP. Zitzler et al. [20] suggested three optimization goals that need to be measured: the distance of the resulting nondominated set to

the Pareto optimal front should be minimized; a good (in most cases uniform) distribution of the solutions found is desirable; the extent of the obtained nondominated front should be maximized. Comparing the performance of MOEAs remains an open problem. The most popular measures are quality indicators (QI); the term "performance metric" is also used to quantify the differences between approximation sets.

Many different QIs for measuring the quality of approximation sets have been proposed in the literature [1, 8, 9, 10, 12, 14, 15, 19, 20, 23, 24]. Each QI has been designed with a standpoint that takes one or more previously mentioned optimization goals into consideration. This means that no single indicator alone can reliably measure MOEA performance. It should be noted that several surveys and experiments have been conducted to analyze individual indicators [7, 8, 16, 24]. The results have shown inconsistencies and contradictions in the assessment of various approximation sets. It was argued in [7] and [16], that without established comparison criteria, claims based on heuristically chosen QIs do little to determine a given MOEAs actual efficiency and effectiveness. In addition, the conclusions are useless for answering the question of which algorithms are superior to others. Can it be argued that one algorithm is better than another even though the outcome depends on the selected QI?

The aim of this paper is to obtain better insight into the impact of the selected QI for the comparison of MOEAs. The focus is on the analysis of different QIs with the help of a chess rating system.

The remainder of the paper is organized as follows. In Section 2, some basic concepts of quality indicators are introduced. The chess rating system with Glicko-2 is presented in Section 3. In Section 4, the execution of the experiment and results are presented. Finally, the paper concludes in Section 5.

2. Quality Indicators

Approximation sets can be compared using dominance relations. However, there are numerous limitations to using this approach. For example, the extent to which one algorithm is better than another cannot be expressed nor can it be expressed in which aspects this is so. Furthermore, when using dominance relations, there are cases in which approximation sets are incomparable. In order to overcome these limitations, QIs have been designed. These indicators quantitatively measure approximations of Pareto optimal fronts. Therefore, QIs are in essence functions that assign each approximation set a real number that reflects different aspects of quality or quality differences. Zitzler et al.

[24] defined a quality indicator I as an m-ary function $I : \Omega^m \to \mathbb{R}$ that assigns each vector $(A_1, A_2, \ldots, A_m)$ of m approximation sets a real value $I(A_1, \ldots, A_m)$. Once the approximation sets are evaluated by indicators, different conclusions can be drawn about their relations. For different aspects of quality, different indicators need to be used.

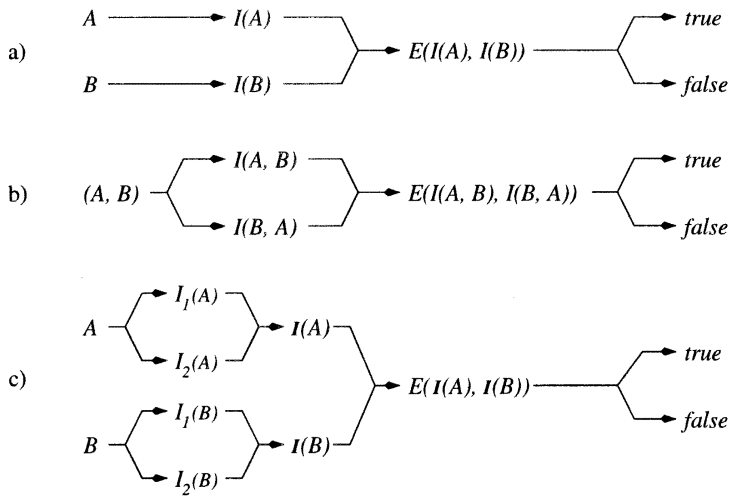

Figure 1: The concept of comparison methods adapted from [24].

Quality indicators have been categorized into different groups from different points of view to better understand their nature [24, 7, 10]. They are mainly categorized by the aspects of quality that they assess. These aspects include the closeness to the Pareto-optimal front, the number of elements of the Pareto-optimal front found, and the maximum spread of solutions. Quality indicators are also classified based on the number of approximation sets they take as an argument. Unary indicators accept one approximation and binary accept two. However, in principle indicators that accept an arbitrary number of arguments are also possible. When evaluating with unary indicators the resulting real values need to be compared in order to see which result set is better. Binary indicators, in contrast, compare two result sets to determine which one is better. Therefore, when comparing t sets using binary indicators $t(t1)$, comparisons need to be carried out to obtain the final ranking. Some unary indicators require a reference set to perform the evaluation, which must be taken into consideration since real-world problems have unknown Pareto-optimal fronts. When the reference set is available, any indicator can be converted from binary to unary. There are also other

categories that are not used as often, such as computational complexity, the sensitivity to scaling, the number of objectives, etc. It is also desirable that an indicator be compatible and complete with respect to dominance relations.
Quality indicators need interpretation, and different comparison methods can be used. This is best illustrated by Zitzler (Fig. 1) [24] where concepts of comparison methods using either only unary or only binary indicators are presented. Case (a) uses a single unary QI, (b) a single binary QI, and (c) a combination of two unary QIs. In cases (a) and (b), the indicator I evaluates the approximation sets A and B. The result is passed to the interpretation function E that decides the outcome. In case (c), two indicators are applied to A and B then the resulting two indicator values are combined in a vector, $I(A)$ for A and vector $I(B)$ for B. The vectors are passed to the interpretation function E that decides the outcome.

Table 1: Quality indicators and their properties.

Quality Indicator	Convergence	Uniformity	Spread	Requires reference set
CS [23]	√			
$I_{\epsilon+}$ [24]	√			√
GD [15]	√			√
HV [23]	√	√	√	
IGD [1], IGD+ [9]	√	√	√	√
MPFE [14]	√			√
MS [20]			√	√
R2 [8]	√			√
S [12]		√		
Generalized Spread Δ [19]			√	√

In this paper, eleven QIs are used, based on prevalence and different properties. Selected indicators are listed with their quality aspects in Table 1. When comparing algorithms, usually a handful of QIs are selected and then the experiment is performed and evaluated with selected statistical methodologies. In our case, the Chess Rating System for Evolutionary Algorithms (CRS4EAs) [13] is used. The outcome of the game was determined by methods a and b (Fig. 1), depending whether the indicator is unary or binary.

3. Chess Rating System for Evolutionary Algorithms (CRS4EAs)

In this paper, we use CRS4EAs based on the Glicko-2 system, in which each player receives his rating R, rating deviation RD and rating volatility σ [6]. The volatility measure indicates the degree of expected fluctuation in a players rating. When a player has an unpredictable performance such as exceptionally strong results after a period of stability, the volatility measure is high. If a player performs at a consistent level, the volatility measure is low. The rating deviation indicates how reliable a players rating is. A small rating deviation means a player plays often and has a reliable rating. In contrast, if the rating deviation is high, his rating is unreliable. A players strength can be summarized in the form of a 95% confidence interval. It can be said that we are 95% confident that the players rating R is within an interval $[R - 2RD, R + 2RD]$. To apply a rating, multiple games between multiple players within a rating period (tournament) need to be performed. Before the tournament starts the ratings, rating deviations and rating volatilities for all players need to be set. If a player is new or not established, his performance rating has to be defined first. The experiment was performed with the Evolutionary Algorithm Rating System (EARS) [5] framework that supports CRS4EAs. The codes for the different algorithms, problems, and calculation of quality indicators are available in the jMetal framework [4] and MOEA framework [11]. Each MOEA represents a chess player and searches for the best Pareto front approximation for a given problem represents a chess game. In a game, two MOEAs play against each other where the outcome is decided when each approximation set is evaluated with the given QI. Each player plays multiple games against all participants in the tournament.

4. Experiment

In this section, the experiment execution and results are presented. Chess ranking leaderboards of five algorithms with eleven QIs were compared.

4.1 Experimental Settings

In the experiment, five MOEAs were chosen for the tournament: IBEA [22], MOEA/D [17], NSGA-II [3], PESA-II [2] and SPEA2 [21]. The benchmark contains well-known unconstrained problems from the CEC 2009 special session and competition on the performance assessment of multi-objective optimization algorithms [18]. Population size

for all five MOEAs was set to be 100 for all of the 2-objective problems and 300 for the 3-objective problems, according to [16]. The rest of parameters setting of the algorithms are set according to the source code of [4, 11]. The maximum number of evaluations for a problem was set to 300,000. The number of independent runs of the tournament was set to 30. For the chess rating, Glickman recommended setting rating R to 1500, rating deviation RD to 350, and rating volatility σ to 0.06 [6]. A tournament was conducted for each QI. It should be noted that approximation sets were normalized prior to evaluation since different objective functions can have a different magnitude.

4.2 Experimental Execution

Figure 2 displays the flowchart of a single execution of the experiment in EARS. The experiment is conducted in the form of tournaments. Each tournament consists of $k = 5$ algorithms $\{a_1, a_2, \ldots, a_5\}$, $N = 10$ optimization problems and is performed in $n = 30$ independent runs. Each algorithm returns the best solution set for each optimization problem over n independent runs ($k * N * n$ results). These results are then evaluated with the QI that was given for the current tournament. After evaluation, the resulting two real values are passed to the interpretation function. The comparison methods use a single unary QI or a single binary QI (Fig. 1 a and b). A set $\{a_i, a_j\}_{l,m}$ is a single comparison or a single game between two algorithms a_i and a_j for the optimization problem F_l over run m where $i, j \in \{1, \ldots, k\}, i \neq j, l \in \{1, \ldots, N\}$ and $m \in \{1, \ldots, n\}$. The solution sets y_i and y_j from algorithms a_i and a_j for the problem F_l on run m are evaluated with the given quality indicator I and passed to the interpretation function E that defines the outcome of the comparison. Therefore, one tournament consists of $(k * (k-1)/2) * N * n$ games. At the end of the tournament, the results are gathered in the forms of wins, losses, and draws. Afterward, the ratings, rating deviations, and rating volatilities are updated. All the data is collected and presented on a leaderboard. The tournament was repeated for each QI, resulting in eleven leaderboards.

4.3 Results and discussion

The results for all QIs are presented in Table 2. All algorithms have played through the whole tournament for each indicator. For each indicator, there are two rows. The first row contains the final rating and rank for a given algorithm. The second row contains the 95% confidence intervals. For all players in all tournaments, the rating deviations reached their minimum value (50) [13]. The low value of RD was achieved with

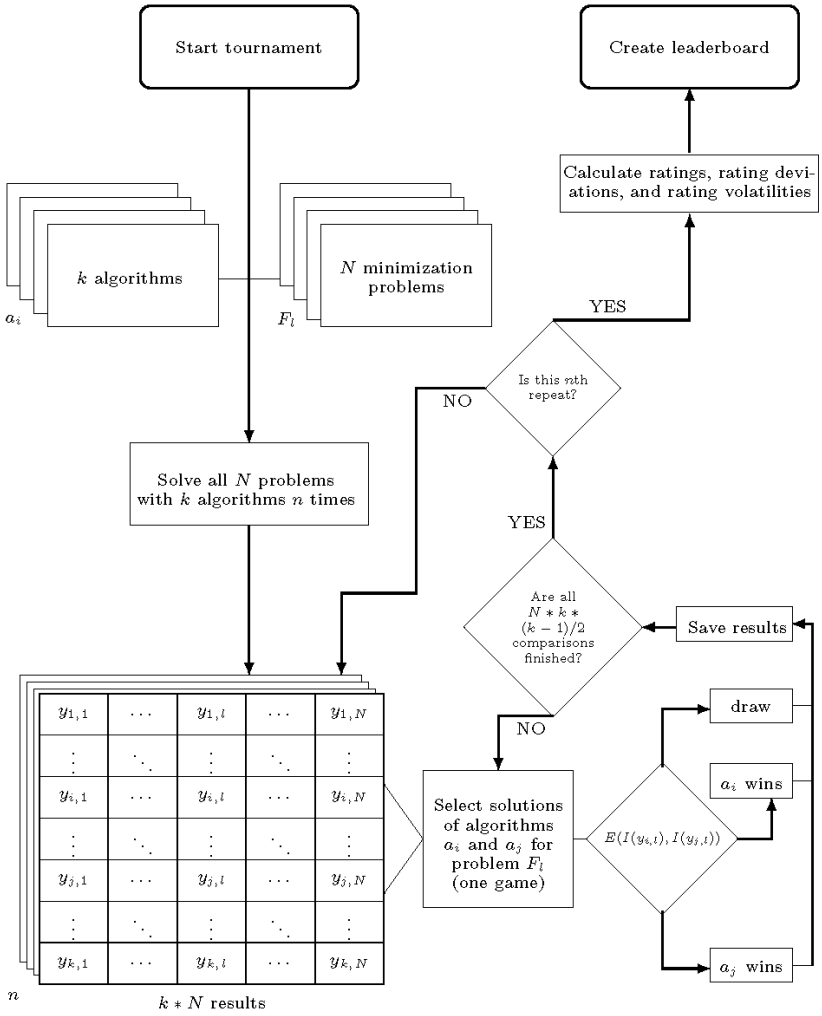

Figure 2: Flowchart of experiment execution in EARS [5].

an adequate number of tournaments, and it indicates that players competed frequently and have a stable rating. As expected, indicators are not unified in the ranking of algorithms, which is reflected in the deviation from the average rank displayed in the last row. Regardless of the incoherence in the ranking, some indicators assess the approximations similarly. Based on similarities in ranking, indicators can be divided into three groups. The biggest group contains seven indicators: HV, IGD, $IGD+$, $I_{\epsilon+}$, $R2$, MS and Δ. The first three indicators have the

Table 2: Leaderboards of five algorithms with eleven QIs on unconstrained CEC 2009 benchmark problems. For each indicator the rating intervals RI with 95% confidence ($RD \pm 2RD$) are presented.

	IBEA	MOEA/D	NSGA-II	PESAII	SPEA2
HV	*1380 (5)* [1280,1480]	**1649 (1)** [1549,1749]	1538 (2) [1438,1638]	1461 (4) [1361,1561]	1471 (3) [1371,1571]
IGD	*1228 (5)* [1128,1328]	**1700 (1)** [1600,1800]	1581 (2) [1481,1681]	1453 (4) [1353,1553]	1538 (3) [1438,1638]
$IGD+$	*1414 (5)* [1314,1514]	**1605 (1)** [1505,1705]	1566 (2) [1466,1666]	1437 (4) [1337,1537]	1478 (3) [1378,1578]
$I_{\epsilon+}$	*1390 (5)* [1290,1490]	**1599 (1)** [1499,1699]	1522 (3) [1422,1622]	1440 (4) [1340,1540]	1548 (2) [1448,1648]
$R2$	*1287 (5)* [1187,1387]	1598 (2) [1498,1698]	**1647 (1)** [1547,1747]	1400 (4) [1300,1500]	1567 (3) [1467,1667]
MS	*1218 (5)* [1118,1318]	1624 (2) [1524,1724]	**1770 (1)** [1670,1870]	1339 (4) [1239,1439]	1549 (3) [1449,1649]
Δ	*1266 (5)* [1166,1366]	1570 (3) [1470,1670]	1628 (2) [1528,1728]	1370 (4) [1270,1470]	**1667 (1)** [1567,1767]
CS	**1818 (1)** [1718,1918]	1399 (4) [1299,1499]	*1287 (5)* [1187,1387]	1525 (2) [1425,1625]	1471 (3) [1371,1571]
GD	**1848 (1)** [1748,1948]	1292 (4) [1192,1392]	*1291 (5)* [1191,1391]	1622 (2) [1522,1722]	1447 (3) [1347,1547]
$MPFE$	**1954 (1)** [1854,2054]	*1170 (5)* [1070,1270]	1339 (4) [1239,1439]	1644 (2) [1544,1744]	1392 (3) [1292,1492]
S	**1831 (1)** [1731,1931]	*1158 (5)* [1058,1258]	1421 (4) [1321,1521]	1633 (2) [1533,1733]	1457 (3) [1357,1557]
$\bar{x}$	3.9	2.9	3.1	3.6	3

same ranking. The remaining indicators ($I_{\epsilon+}$, $R2$, MS and Δ) have ranked differently, but there is no significant difference between the algorithms that switched ranks. The other two groups ranked the MOEAs very differently than the bigger group. If only the ranking is considered, two pairs of indicators are obtained, which differ only in the rank of $MOEA/D$ and $NSGA-II$. Since there is no significant difference between the fourth and fifth ranking algorithm with the $MPFE$ indicator, we grouped it with CS and GD. Although S achieved the same ranking as $MPFE$, it is in a separate group because there is a significant difference between $MOEA/D$ and $NSGA-II$. The bigger group contains all three compliant indicators: one strictly Pareto-compliant indicator (HV) and two weakly Pareto-compliant indicators ($IGD+$ and the unary $I_{\epsilon+}$). Since compliant indicators are deemed to be more reliable,

we conclude that the ranking of the bigger group is also more reliable. It is also interesting to note that indicators within the same group do not evaluate the same aspects of quality. This can be interpreted as indicating that the resulting approximation sets do not dominate only in one optimization goal. In Table 2, the rating was used to show the absolute power of the algorithm over other algorithms, however, the rating interval should also be considered. If the confidence intervals do not overlap, the algorithms have provided significantly different results, whereas the converse is not necessarily true.

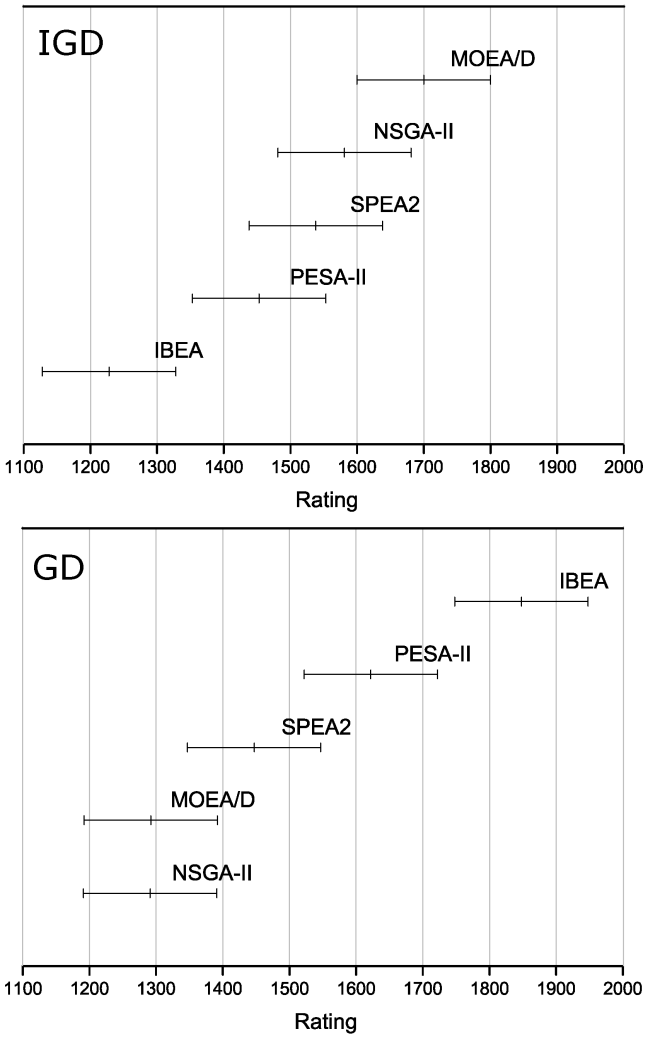

Figure 3: 95% confidence intervals for IGD (top) and GD (bottom) QI in table 2.

Due to space constraints we plotted the confidence intervals for IGD and GD (e.g., Fig. 3), which are some of the more popular QIs in literature. The results can be interpreted by observing ratings and rating interval. As we can see with the IGD indicator, $MOEA/D$ performed the best, being significantly better than $PESAII$ and $IBEA$. On the last place is $IBEA$, which performed the worst, being significantly outperformed by all other algorithms. In contrast, with GD $IBEA$ performed the best by significantly outperforming $MOEA/D$, $NSGAII$, $SPEA2$, and $PESAII$. On the second place is $PESAII$, outperforming $MOEA/D$ and $NSGAII$, which shares the last place with one point of difference. It is important to observe that the selected QI ranked the MOEAs almost in reverse order. This result can be explained by the property of GD indicator that measures only the convergence of the approximation set regardless of its spread and uniformity. Furthermore, the experiment was limited by selected set of problems, MOEAs, and QIs.

5. Conclusion

In this paper, eleven QIs were compared with CRS4EAs on five different MOEAs, solving unconstrained MOP from the CEC 2009 benchmark. For the given experiment, it has been shown that individual QIs differently rank algorithms even if they evaluate the same aspects of quality. Therefore, picking coherent indicators is very important. Selected QIs were categorized into three groups that have insignificant differences in MOEAs ranking. The biggest group with the state-of-the-art indicator contains Δ, $I_{\epsilon+}$, HV, IGD, $IGD+$, $R2$ and MS indicator. The other two groups containing CS, GD, $MPFE$ and S indicator returned very different ranking orders and are not recommended. Because of the disparity in rankings between indicators, a desired ranking of algorithms can be achieved with a carefully assembled set of indicators [16]. Therefore, in order to claim that one algorithm is better, a balanced and fair set of indicators is recommended. For future work, we would like to integrate this approach into CRS4EAs and test it on additional diverse problems for more detailed analysis of QIs.

References

[1] P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.

[2] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESAII: Region-based selection in evolutionary multiobjective optimization. *Proceedings of the*

Genetic and Evolutionary Computation Conference (GECCO), pages 124-130, 2001.

[3] K. Deb, A. Pratab, S. Agrawal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197, 2002.

[4] J. J. Durillo, and A. J. Nebro. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10): 760–771, 2011.

[5] Evolutionary Algorithms Rating System (Github). `https://github.com/matejxxx/EARS`, 2016.

[6] M.E. Glickman. Example of the Glicko-2 System. Boston University, 2012.

[7] D. Guoqiang, Z. Huang, and M. Tang. Research in the Performance Assessment of Multi-objective Optimization Evolutionary Algorithms. *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS)*, pages 915–918, 2007.

[8] M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the nondominated set. Technical Report IMM-REP-1998-7, 1998.

[9] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. *Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making*, pages 170–177, 2014.

[10] M. Li, S. Yang, and X. Liu. Diversity comparison of Pareto front approximations in many-objective optimization. *IEEE Transactions on Cybernetics*, 44(12): 2568–2584, 2014.

[11] MOEA Framework - A Free and Open Source Java Framework for Multiobjective Optimization. `http://www.moeaframework.org`, 2016.

[12] J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master Thesis, MA: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[13] N. Veček, M. Mernik, and M. Črepinšek. A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. *Information Sciences*, 277(1): 656–679, 2014.

[14] D. A. Van Veldhuizen. Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations. Ph.D. dissertation, Faculty of the Graduate School of Engineering , Air Force Institute of Technology, 1997.

[15] D. A. Van Veldhuizen and G. B. Lamont. Evolutionary computation and convergence to a Pareto front. *Proceedings of the Genetic Programming Conference*, pages 221-228, 1998.

[16] G. G. Yen and Z. He. Performance metric ensemble for multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 18(1):131–144, 2014.

[17] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[18] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487, 2009.

[19] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 892-899, 2006.

[20] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[21] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: improving the strength Pareto evolutionary algorithm. Technical Report TIK-Report 103, 2001.

[22] E. Zitzler and K. Simon. Indicator-based selection in multiobjective search. *Proceedings of International Conference on Parallel Problem Solving form Nature (PPSN)*, pages 832–842, 2004.

[23] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257-271, 1999.

[24] E. Zitzler and L. Thiele. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 117–132, 2003.