Efficiency and Effectiveness Metrics in Evolutionary Algorithms and Their Application

Guo-Sheng Hao[1](✉), Chang-Shuai Chen[1], Gai-Ge Wang[1],
Yong-Qing Huang[1,3], De-Xuan Zhou[2], and Zhao-Jun Zhang[2]

[1] School of Computer Science and Technology,
Jiangsu Normal University, Xuzhou 221116, Jiangsu, China
guoshenghaoxz@tom.com
[2] School of Electrical Engineering and Automation,
Jiangsu Normal University, Xuzhou 221116, Jiangsu, China
zzj921@163.com
[3] School of Mathematics and Computer, Tongling University,
Tongling 244000, Anhui, China
yongqinghuangbs@sina.com

Abstract. Efficiency and effectiveness are two important metrics for the evaluation of evolutionary algorithms (EAs). Firstly, there exist a number of efficiency metrics in EA, such as population size, number of termination generation, space complexity, and time complexity and so on. But the relationship of these metrics is left untouched. And evaluating or comparing EAs with one of these metrics or using them separately is unfair. Therefore it is necessary to consider their relationship and give proper metrics combination. We conclude that the product of population size and number of generation should be less than the value of search space size, and the product of time complexity and space complexity should also be less than a constant. Secondly, we study the relationship between efficiency and effectiveness. Based on these two metrics, we conclude that not only EAs can be compared, but also problems hardness can be measured. The results reveal important insights of EAs and problems hardness.

Keywords: Evolutionary algorithms · Performance · Efficiency · Effectiveness · Problems hardness · Population size · Stop generation

1 Introduction

Evolutionary algorithms (EAs) [1] are problem independent and have been reported to perform relatively well on problems such as: search, optimization, and artificial intelligence.

The analysis of problem hardness (difficulty) has been studied for decades [2]. Measuring hardness of an instance for a particular algorithm is typically done by comparing the optimization precision reached after a certain number of iterations compared to other algorithms, and/or by comparing the number of iterations taken to reach the best solution [3]. There have been numerous efforts identifying challenges

D.-S. Huang et al. (Eds.): ICIC 2015, Part II, LNCS 9226, pp. 1–12, 2015.
DOI: 10.1007/978-3-319-22186-1_1

such as isolation, deception, multi-modality, the size of basins of attraction [4], as well as landscape metrics [5], autocorrelation structures and distributions of local minima [6].

In optimization algorithms, performance measurement is an important topic and effectiveness and efficiency are two aspects of the performance of algorithms. Firstly, effectiveness refers to the quality of the obtained solution and most of its indicators are directly related to the optimized objective(s) [7], and there is also effective metric about the solution (population) diversity [8]. Secondly, efficiency is usually characterized by its runtime behavior, i.e., the order of its computation time and its memory requirements [9] and its indicators are mainly about the algorithm. Efficiency can be increased drastically with the use of domain knowledge which can be embedded in algorithms, while effectiveness highly depends on the properties of the problem [10]. To our knowledge, the issues that relationship between efficiency and effectiveness, the relationship among metrics of efficiency are still not analyzed deeply. And this is the purpose of this paper.

2 Efficiency Metrics of Evolutionary Algorithms

2.1 Population Size and Termination Generation Number

To calculate the efficiency, for the population size, only the number of individuals who get evaluation from environment is considered, but not from surrogate, in other words, who get real evaluation but not virtual evaluation. As shown in Fig. 1, there are two kinds of individuals' fitness evaluation: directly from environment and from the surrogate. Generally surrogate is developed for cheaper fitness functions, but at the cost of accuracy of fitness estimation [11].

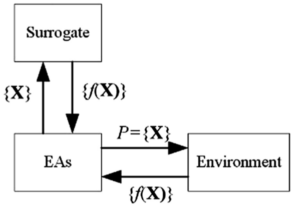

Fig. 1. Individuals evaluation structure in EAs.

The product of population size and the number of generation is the number of individuals whose information is used by the algorithm. In other words, the product is the number of individuals who get their evaluation from environment or whose evaluation is the real one but not the estimated one by surrogate. With the same obtained best solution X_{best}^{real}, the less the product is, the less resources the algorithm costs, and the higher the efficiency is.

Label the number of those individuals who have the real evaluation till the current evolutionary generation t as $N_{ind}(t)$, which can be simply explained as the product of the population size $|P|$ and t, and there should be:

$$N_{ind}(t) = |P| \times t \tag{1}$$

$N_{ind}(t)$ should not be more than $|S|$. It can be taken as granted that EAs are better than the traversal methods on searching optima, and the latter will have to visit all the individuals in search space. Therefore, for the termination generation T of each algorithm, there should be:

$$|S| \geq N_{ind}(T) \tag{2}$$

The relationship among $|P|$, T, $N_{Ind}(T)$ and $|S|$ is illustrated in Fig. 2, in which, the outer curve of the shadow area is $|P| \times T{=}|S|$, which is also the upper bound of $N_{ind}(T)$. The lower bounds for T and $|P|$ are labeled as T_0 and $|P|_0$ respectively.

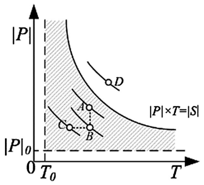

Fig. 2. Relationship of population size and generation number

According to (3), it is easy to know that the points A, B and C, in Fig. 2, are the examples of $N_{Ind}(T)$, while point D is not. The product value of $|P| \times T$ corresponding to points decreases with the order of D, A, B and C.

In fact, $N_{ind}(T)$ as a metrics of efficiency has been used many times [12, 13], but the relationship between $|P|$ and T was not given.

2.2 Space Complexity and Time Complexity

The definitions of $N_{ind}(T)$ does not include enough elements. For example, although $N_{ind}(T)$ consider the time-complexity with T and space-complexity with $|P|$, they are only the basic complexity.

Space-complexity of the algorithms is related with population size, but it can’t be simply represented with it. There are many memory-based algorithms, that use the same population size as other algorithms, but their space complexity is more than others.

Similarly, time-complexity of the algorithm is related with the termination generation number, but it can’t be simply represented with it. There are many data-mining embedded algorithms that spend more time to extract rules to guide the algorithms to explorer/exploit promising search space. In addition to the search process, establishing a high quality model in EAs might require longer time [14]. In a competing heuristics situation [15], time-complexity is also heavy, because the solutions are repeated generated and evaluated until it dominates the worst one in current population.

In order to compare the space-complexity, the memory cost per individual is considered. Label the memory cost of an algorithm during the evolutionary process as M_C, and define the memory cost per individual as:

$$m_C = \frac{M_C}{|S|} \tag{3}$$

Similarly, in order to compare the time-complexity, the time cost of an algorithm per individual is considered. Label the time cost during the evolutionary process as T_C, and define the time cost per individual as:

$$t_C = \frac{T_C}{|S|} \tag{4}$$

It has been mentioned many times in literature that under certain efficiency condition, time complexity is inversely proportion the space complexity. Sometimes, in order to save time, an algorithm will have to run with high space complexity, or in order to save memory to run in long time. Therefore, the relationship of m_C and t_C is shown in Fig. 3.

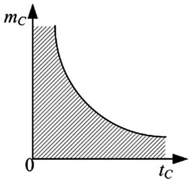

Fig. 3. Relationship between m_C and t_C

For a certain problem, algorithm with high m_C will be with the low t_C. Therefore, the product of power of m_C and power of t_C also has a upper bound. With this idea, a complexity metric can be defined as:

$$p_C = \frac{|S|}{|S| + \omega_{P1} M_C \times T_C} \tag{5}$$

where $\omega_{P1} = e^{\lfloor ln|S| \rfloor - \lfloor ln(M_C \times T_C) \rfloor}$, and it is for the balance of magnitudes. Therefore, there is $p_C \in (0,1)$.

2.3 Efficiency of Evolutionary Algorithms

The algorithm should not visit every individual. It is assumed that the increasing of M_C and T_C will result in the decreasing of $|P| \times T$. Therefore, m_C and t_C are inversely proportion to $N_{ind}(T)$ with the same algorithm and the same problem. Therefore, there is:

$$\frac{\partial N_{ind}(T)}{\partial M_C} < 0, \frac{\partial N_{ind}(T)}{\partial T_C} < 0 \tag{6}$$

Based on the relationship of $|P|$ and T, one of the metrics of the efficiency of algorithm can be defined as:

$$p_{N1} = 1 - \omega_{P2}\frac{N_{ind}(T)}{|S|} = 1 - \omega_{P2}\frac{|P| \times T}{|S|} \tag{7}$$

where $\omega_{P1} = e^{\lfloor ln|S| \rfloor - \lfloor ln(|P| \times T) \rfloor}$, which is for the balance of magnitudes. It is easy to know $p_{N1} \in [0,1]$. The bigger p_{N1} is, the better the efficiency is.

Based on the analysis of M_C, T_C and $N_{ind}(T)$, the following basic logistics can be given:

$$\begin{gathered} M_C \uparrow\Rightarrow T_C \downarrow, T_C \uparrow\Rightarrow M_C \downarrow, M_C \uparrow\Rightarrow N_{Ind}(T) \downarrow, \\ T_C \uparrow\Rightarrow N_{Ind}(T) \downarrow, N_{Ind}(T) \downarrow\Rightarrow P_{N1} \uparrow, M_C \uparrow\Rightarrow P_C \downarrow, T_C \uparrow\Rightarrow P_C \downarrow \end{gathered} \tag{8}$$

Based on the above logistics, the efficiency metric can be defined as:

$$p_{e1} = \omega_1 p_{N1} + \omega_2 p_C \tag{9}$$

where $\omega_1 + \omega_2 = 1$.

This efficiency metric includes parameters of population size, number of termination generation, time-complexity and memory-complexity. All the values of these parameters can be calculated prior to the running of the algorithm. Compared with effectiveness, this is the advantage of efficiency metrics.

3 Relationship of Efficiency and Effectiveness of Evolutionary Algorithms

3.1 Effectiveness in EAs

It is sure that, for two algorithms, the one, which obtains a solution nearer from the real optimum than that of another algorithm, has higher effectiveness. Therefore, a measurement of the effectiveness of the algorithm based on distance of obtained solution from the real optimum can be given. Generally, it is difficult to know the real optimum, but a solution better than or equal to the best obtained one can be given. Label this solution as $\mathrm{X}_{best}^{virtaul}$, the precision as δ. Label $d_b = \left|\mathrm{X}_{best}^{real} - \mathrm{X}_{best}^{virtual}\right|$, then d_b can be regarded as the metric for the improvement of population. According to Deb's necessary properties for metrics [8], the effectiveness p_{e2} of the performance should satisfy:

$$p_{e2} \in [0,1], \frac{\partial p_{e2}}{\partial d_b} < 0 \tag{10}$$

This means that with the decreasing of the distance, the effectiveness increases. The *progress rate* [16] is one of the metrics that satisfy this property, but it is complex to calculate.

For simplicity, the expression is as following:

$$p_{e2} = \frac{\delta}{\left|\left|\mathrm{X}_{best}^{real} - \mathrm{X}_{best}^{virtual}\right|\right| + \delta} = \frac{\delta}{d_b + \delta} \tag{11}$$

3.2 Relationship Between Efficiency and Effectiveness

Usually, the longer time an EA run (with a sufficient population size), the higher the solution quality will be [14]. However, in real-world scenarios, the computational resources are often limited, which leads to a tradeoff between efficiency and the effectiveness. This means that although efficiency p_{e1} and effectiveness p_{e2} are two elements of the performance **p**, they are not independent. If an algorithm is assigned with big value for c_P or T, it will be expected to have higher effectiveness, while large value of c_P or T also means low efficiency, in another word, an algorithm with low p_{e1} should have high p_{e2}.

Assuming that for a certain algorithm, p_{e1} could be inversely proportion to p_{e2}. Intuitively, for a certain algorithm, when the efficiency is improved, the effectiveness will be decreased, and vice versa. This means that for a certain algorithm, the improvement of efficiency will bring the low effectiveness and the improvement of effectiveness will bring the low efficiency. Therefore, there should be a tradeoff between p_{e1} and p_{e2}.

But for different algorithm variants, the performance will be different. Some of the variants can increase the efficiency and some of them can increase the effectiveness and some of them can increase both of the effectiveness and efficiency. The last kind of algorithm is the one that is preferred.

The relationship that p_{e1} is inversely proportional top_{e2} can be seen from the aspect of space precision δ. Denote the dimension or number of variants as d, which means that $\mathbf{X} = \{x^{\{i\}} | i = 1, 2, \ldots, d\}$, and D_i as the scope of i-th dimension. Label the upper and low bound of every dimension as u_{Di} and l_{Di}, and $d_D = \prod_{i=1}^{d} (u_{Di} - l_{Di})$, then there is:

$$|S| = \prod_{i=1}^{d} |D_i| = \prod_{i=1}^{d} \frac{U_{Di} - L_{Di}}{\delta} = \frac{d_D}{\delta^d} \tag{12}$$

where d_D is a constant value and $\delta > 0$. Therefore, according to (8) and (12), there is:

$$\frac{\partial p_{e1}}{\partial \delta} > 0, \frac{\partial p_{e2}}{\partial \delta} < 0 \tag{13}$$

Therefore, from the viewpoint of decision variable δ, p_{e1} is approximately inversely proportional top_{e2}, and they can be seen as conflict objectives. For a certain algorithm, the relationship of p_{e1} and p_{e2} is illustrated in Fig. 4, in which, the intersection point

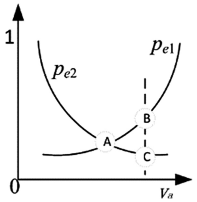

Fig. 4. Relationship between pe1 and pe2 for a certain algorithm

“A” of p_{e1} and p_{e2} is the tradeoff. And for the point of “B”, p_{e1} is increased while p_{e2} is decreased. Similarly, for the point of “C”, p_{e2} is increased while p_{e1} is decreased.

The relationship that p_{e1} is inversely proportional to p_{e2} also can be seen from the aspect of complexity metric p_C. It is easy to know that:

$$\frac{\partial p_C}{\partial M_C} < 0, \frac{\partial p_C}{\partial T_C} < 0 \tag{14}$$

And there is

$$\frac{\partial p_{e1}}{\partial M_C} < 0, \frac{\partial p_{e1}}{\partial T_C} < 0, \frac{\partial p_{e2}}{\partial M_C} > 0, \frac{\partial p_{e2}}{\partial T_C} > 0 \tag{15}$$

Based on the above analysis, the logistics relationship among the parameters and the algorithm’s performance is given in Fig. 5.

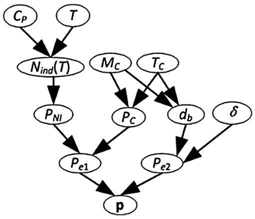

Fig. 5. Logistic relationships among the parameters and algorithm performance

Algorithm variants are always proposed in order to improve the performance. For example, in Fig. 6, it can be taken as granted that the original performance tradeoff is at the point Ⓓ. Then the tradeoff is improved to points “A”, “B” and “C”, during which “A” is the best tradeoff, which has both high p_{e1} and p_{e2}.

3.3 Definition of Performance

Based on the above analysis, a more sophisticated performance indicator is proposed as:

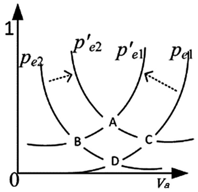

Fig. 6. Relationships among algorithms' efficiency and effectiveness

$$\mathrm{p} = (p_{e1}, p_{e2}) \tag{16}$$

Based on this indicator, the definition of dominance relationship on performance can be given.

Definition 1 (Dominance Relationship on Performance). For two performance vector $\mathrm{p}_1 = (p_{e1,1}, p_{e2,1})$ and $\mathrm{p}_2 = (p_{e1,2}, p_{e2,2})$, The preference orders on the set of performance vector can be defined.

- $\mathrm{p}_1 \succ \mathrm{p}_2(\mathrm{p}_1$ dominates $\mathrm{p}_2)$, if $\mathbf{p}_1$ is not worse than $\mathbf{p}_2$ in any element, and is better in at least one of the elements;
- $\mathbf{p}_1 \succeq \mathbf{p}_2(\mathbf{p}_1$ weakly dominates $\mathbf{p}_2)$, if $\mathbf{p}_1$ is not worse than $\mathbf{p}_2$ in any elements;
- $\mathrm{p}_1 = \mathrm{p}_2(\mathrm{p}_1$ equals $\mathrm{p}_2)$, if $\mathbf{p}_1 = \mathbf{p}_2$;
- $\mathrm{p}_1 \| \mathrm{p}_2(\mathrm{p}_1$ and p_2 are incomparable to each other), if neither $\mathbf{p}_1$ weakly dominates $\mathbf{p}_2$ nor $\mathbf{p}_2$ weakly dominates $\mathbf{p}_1$.

The relationship $\prec$, $\preceq$ are defined accordingly, i.e., $\mathrm{p}_1 \succ \mathrm{p}_2$ is equivalent to $\mathrm{p}_2 \prec \mathrm{p}_1$, etc.

4 Application of Performance Measurement

Performance **p** is determined by an algorithm *A* and the optimization problem *pl*. This from the definition of p_{e1} is the variables of c_p and *T*, and both c_p and *T* are related with the algorithm *A*. Similarly, from the definition of p_{e2}, it can be seen that the variables of δ and d_b are also related with *pl* and *A*. Therefore, for distinguishing performance, **p**(*pl*, *A*) can be used to explicitly illustrate that the performance is for the concrete problem *pl* and concrete algorithm *A*.

4.1 Algorithm Comparison

For the same problem *pl* with the same space size $|S|$, all EAs can be compared with performance **p**. the algorithms domination definition is given as following.

Definition 2 (Algorithms Domination Relationship on Certain Problem). For two EAs A_1 and A_2, suppose their performance is $\mathbf{p}_1(pl, A_1)$ and $\mathrm{p}_2(pl_2, A\neg_2)$ respectively. The performance orders can be defined as:

- $A_1 \succ_{pl} A_2 (A_1$ dominates A$_2)$, if $\mathbf{p}_1$ is not worse than $\mathbf{p}_2$ in any element, and is better in at least one of the elements;
- $A_1 \succeq_{pl} A_2 (A_1$ weakly dominates $A_2)$, if $\mathbf{p}_1$ is not worse than $\mathbf{p}_2$ in any elements;
- $A_1 =_{pl} A_2$ (A_1 equals A$_2)$, if $\mathbf{p}_1 = \mathbf{p}_2$;
- $A_1 ||_{pl} A_2 (A_1$ and A_2 are incomparable to each other), if neither $\mathbf{p}_1$ weakly dominates $\mathbf{p}_2$ nor $\mathbf{p}_2$ weakly dominates $\mathbf{p}_1$.

The relationship $\prec, \preceq$ are defined accordingly, i.e., $A_1 \succ_{pl} A_2$ is equivalent to $A_2 \prec_{pl} A_1$, etc.

Suppose that all the problems compose of a set $PL = \{pl_1, pl_2, \ldots\}$. Based on this set, the theoretic case for the algorithms comparison can be defined as $A_1 \succ_{PL} A_2$ and so on. For the same problem with the same precision δ, if three EAs: a_1, a_2 and a_3 obtain the same optima and the efficiency are corresponding to the points C, B and A respectively in Fig. 1, then there is $\mathrm{p}_1(A_1, pl) \succ \mathrm{p}_2(A_2, pl) \succ \mathrm{p}_3(A_3, pl)$ and $A_1 \succ_{pl} A_2 \succ_{pl} A_3$.

4.2 Problem Hardness Measurement

For different problems pl_1, pl_2 with the same space size $|S|$, if the same algorithm performs differently on them with $\mathrm{p}_1(pl_1, A\neg)$ and $\mathrm{p}_2(pl_2, A\neg)$, then the definition about problem hardness can be given in the following definition.

Definition 3 (Hardness Domination Relationship of Problems on Certain Algorithm). For a concrete EAs A, label the performance for two problems pl_1 and pl_2 as $\mathrm{p}_1(pl_1, A\neg)$ and $\mathrm{p}_2(pl_2, A)$ respectively. The preference orders can be defined as:

- $pl_1 \prec_A pl_2 (pl_1$ dominates $pl_2)$, if $\mathbf{p}_1$ is not worse than $\mathbf{p}_2$ in any element, and is better in at least one of the elements;
- $pl_1 \prec_A pl_2 (pl_1$ weakly dominates $pl_2)$, if $\mathbf{p}_1$ is not worse than $\mathbf{p}_2$ in any elements;
- $pl_1 =_A pl_2 (pl_1$ equals $pl_2)$, if $\mathbf{p}_1 = \mathbf{p}_2$;
- $pl_1 ||_A pl_2 (pl_1$ and pl_2 are incomparable to each other), if neither $\mathbf{p}_1$ weakly dominates $\mathbf{p}_2$ nor $\mathbf{p}_2$ weakly dominates $\mathbf{p}_1$.

The relationship $\prec, \preceq$ are defined accordingly, i.e., $pl_1 \succ_A pl_2$ is equivalent to $pl_2 \prec_A pl_1$, etc.

Suppose that all the algorithms compose of a set Al = $\{A_1, A_2, \ldots\}$. The theoretic case for the problems comparison can be defined as $pl_1 \succ_{Al} pl_2$ and so on.

Generally speaking, there are two kinds of measurements for the problems difficulty: prior estimation and post calculation. One can see that the above definition is based on the post calculation.

4.3 Comparison Design

Efficiency and effectiveness are metrics for the comparison of algorithms (and its parameters' value) and problems. These two metrics are related with at least three elements: algorithm, value of parameters of the algorithm and the optimized problem.

Based on the comparison of these three elements, experiments can be carried out to validate the relationship between efficiency and effectiveness. The comparison of these elements is shown in Table 1, in which, the last column gives the suitability to compare for the corresponding case. And in this table, 0 means the same case, 1 means the difference case.

Table 1. Comparison of elements

Number	Algorithms	Values of parameters	Optimized problems	Comparison
1	0	0	0	None
2	0	0	1	Hardness
3	0	1	0	Performance
4	0	1	1	Suitability
5	1	0	0	Performance/None
6	1	1	0	Performance
7	1	0	1	None
8	1	1	1	None

- For the case of No. 1, the three elements, algorithms, values of parameters and optimized problems, are all with the same situations; therefore, it is unnecessary to compare the performance. But generally, it is always to be utilized to calculate the mean and variance of algorithms' performance.
- For the case of No. 2, two elements, algorithms and its parameters' values are same and only the optimized problems are different. Therefore, by the performance of the algorithm, the hardness of the optimized problem can be compared.
- Similar to the case of No. 2, No. 3 can be used to compare the influence of parameters' values on the performance of algorithms. And the case of No. 4 can be used to compare the performance for different problems with different parameters' values.
- For the case of No. 5, when the compared algorithms have the same parameters, it can be used to compare the performance for different algorithms with the same parameters' value and for the same optimized problems. But generally speaking, since parameters are part of an algorithm, different algorithm may have parameters with different meaning resulting in their incomparability. Therefore, the value in the last column is performance/none. Similarly, for the case of No. 6, the performance can be compared.
- For the case of No. 7 and 8, both the algorithms and optimized problems are different; therefore, both of these two elements are incomparable.

Based on the analysis in Table 1, the experiments to compare performance can be as the cases of No. 3, 5 and 6, and the experiments to compare hardness can be as the case of No. 2.

5 Conclusion

As two important metrics for the evaluation of evolutionary algorithms (EAs), efficiency and effectiveness are studied in this paper. For efficiency metrics, population size, number of termination generation, space complexity, and time complexity and their relationship were studied. We conclude that the product of population size and number of generation should less than the search space size, and the product of time complexity and space complexity should also less than a constant. Secondly, we study the relationship between efficiency and effectiveness. Based on these two metrics, we conclude that not only EAs can be compared, but also problems hardness can be measured. The results reveal important insights of EAs and problems hardness.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grant No. 61305149, 61403174, by Jiangsu Provincial Natural Science Foundation under Grant No. BK20131130, by China Ministry of Education, Humanities and Social Sciences Youth Foundation under Grant No. 11YJC630074 and by Jiangsu Overseas Research & Training Program for University Prominent Young & Middle-aged Teachers and Presidents.

References

1. Yang, X.-S.: Nature-Inspired Optimization Algorithms. Elsevier, Oxford (2014)
2. Malan, K.M., Engelbrecht, A.P.: Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance. Paper presented at the 2013 IEEE Congress on Evolutionary Computation (CEC), 20–23 Jun 2013
3. Xin, B., Chen, J., Pan, F.: Problem difficulty analysis for particle swarm optimization: deception and modality. In: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, Shanghai, China, pp. 629–630 (2009)
4. Gras, R.: How efficient are genetic algorithms to solve high epistasis deceptive problems? Paper presented at the IEEE World Congress on Computational Intelligence, 1–6 Jun 2008
5. Smith-Miles, K., Lopes, L.: Measuring instance difficulty for combinatorial optimization problems. Comput. Oper. Res. **39**(5), 875–889 (2012). Elsevier B.V
6. Malan, K.M., Engelbrecht, A.P.: A survey of techniques for characterising fitness landscapes and some possible ways forward. Inf. Sci. **241**, 148–163 (2013). doi:10.1016/j.ins.2013.04.015
7. Jin, Y., Sendhoff, B.: Trade-off between performance and robustness: an evolutionary multiobjective approach. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 237–251. Springer, Heidelberg (2003)
8. Deb, K., Jain, S.: Running performance metrics for evolutionary multi-objective optimizations. Paper presented at the Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002), Singapore (2002)
9. Merz, P.: Advanced fitness landscape analysis and the performance of memetic algorithms. Evol. Comput. **12**(3), 303–325 (2004). doi:10.1162/1063656041774956
10. Lu, C.-C., Yu, V.F.: Data envelopment analysis for evaluating the efficiency of genetic algorithms on solving the vehicle routing problem with soft time windows. Comput. Ind. Eng. **63**(2), 520–529 (2012). doi:10.1016/j.cie.2012.04.005

11. Jin, Y.: Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol. Comput. **1**(2), 61–70 (2011)
12. Cooper, J., Hinde, C.: Improving genetic algorithms' efficiency using intelligent fitness functions. In: Chung, P.H., Hinde, C., Ali, M. (eds.) IEA/AIE 2003. LNCS, vol. 2718, pp. 636–643. Springer, Heidelberg (2003)
13. Sastry, K., Goldberg, D.E., Pelikan, M.: Efficiency enhancement of probabilistic model building genetic algorithms. In: Illinois Genetic Algorithms Laboratory (2004)
14. Sastry, K., Pelikan, M., Goldberg, D.: Efficiency enhancement of estimation of distribution algorithms. In: Pelikan, M., Sastry, K., CantúPaz, E. (eds.) Scalable Optimization via Probabilistic Modeling. Studies in Computational Intelligence, vol. 33, pp. 161–185. Springer, Heidelberg (2006)
15. Tvrdík, J., Misik, L., Krivy, I.: Competing heuristics in evolutionary algorithms. In: Intelligent Technologies-Theory and Applications, pp. 159–165 (2002)
16. Beyer, H.-G., Schwefel, H.-P., Wegener, I.: How to analyse evolutionary algorithms. Theoret. Comput. Sci. **287**(1), 101–130 (2002). doi:10.1016/S0304-3975(02)00137-8