On the Approximation Ability of Evolutionary Optimization with Application to Minimum Set Cover: Extended Abstract*

Yang Yu[1] and **Xin Yao**[2] and **Zhi-Hua Zhou**[1]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] Center of Excellence for Research in Computational Intelligence and Applications,
School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK
yuy@nju.edu.cn, x.yao@cs.bham.ac.uk, zhouzh@nju.edu.cn

Abstract

Evolutionary algorithms (EAs) are a large family of heuristic optimization algorithms inspired by natural phenomena, and are often used in practice to obtain satisficing instead of optimal solutions. In this work, we investigate a largely underexplored issue: the approximation performance of EAs in terms of how close the obtained solution is to an optimal solution. We study an EA framework named simple EA with isolated population (SEIP) that can be implemented as a single- or multi-objective EA. We present general approximation results of SEIP, and specifically on the minimum set cover problem, we find that SEIP achieves the currently best-achievable approximation ratio. Moreover, on an instance class of the k-set cover problem, we disclose how SEIP can overcome the difficulty that limits the greedy algorithm.

1 Introduction

Evolutionary algorithms (EAs) [Bäck, 1996] have been successfully applied to many fields and can achieve extraordinary performance in addressing some real-world hard problems, particularly NP-hard problems [Koza *et al.*, 2003; Hornby *et al.*, 2006; Benkhelifa *et al.*, 2009]. To gain an understanding of the behavior of EAs, many theoretical studies have focused on the running time required to achieve exact optimal solutions [He and Yao, 2001; Yu and Zhou, 2008; Neumann and Witt, 2010; Auger and Doerr, 2011]. In practice, EAs are most commonly used to obtain *satisficing* solutions, yet theoretical studies of their approximation ability have only emerged in the recent decade.

He and Yao (2003) first studied conditions under which the *wide-gap far distance* and the *narrow-gap long distance* problems are hard to approximate using EAs. Giel and Wegener (2003) investigated a (1+1)-EA for the *maximum matching* problem and found that the time taken to find exact optimal solutions is exponential but is only $O(n^{2\lceil 1/\epsilon \rceil})$ for $(1+\epsilon)$-approximate solutions, which demonstrates the value of EAs as approximation algorithms. Subsequently, further results on the approximation ability of EAs were reported. For the (1+1)-EA, the simplest type of EA, two directions of results have been obtained. On one hand, it was found that the (1+1)-EA has an arbitrarily poor approximation ratio for the *minimum vertex cover* as well as the *minimum set cover* problem [Friedrich *et al.*, 2010; Oliveto *et al.*, 2009]. Meanwhile, it was also found that (1+1)-EA can be a good approximation algorithm for a subclass of the *partition* problem [Witt, 2005] and some subclasses of the minimum vertex cover problem [Oliveto *et al.*, 2009], and can be a useful post-optimizer [Friedrich *et al.*, 2009].

Recent advances in multi-objective EAs have shed light on the power of EAs as approximation optimizers. For a single-objective problem, the *multi-objective reformulation* introduces an auxiliary objective function, for which a multi-objective EA is then used as the optimizer. Scharnow *et al.* (2002) suggested that multi-objective reformulation could be superior to the original single objective formulation. This was confirmed for various problems [Neumann and Wegener, 2005; 2007; Friedrich *et al.*, 2010; Neumann *et al.*, 2008] by showing that while a single-objective EA gets stuck, multi-objective reformulation solves the problems efficiently. Regarding approximations, it has been shown that multi-objective EAs are effective for some NP-hard problems, including the minimum set cover problem[Friedrich *et al.*, 2010] and the *minimum multicuts* problem [Neumann and Reichel, 2008].

In the present study, we investigate the approximation performance of EAs by introducing a framework *simple evolutionary algorithm with isolated population* (SEIP), which can be implemented as a single- or multi-objective EA. We obtain general approximation results of SEIP; and then on the minimum set cover problem (MSCP), we prove SEIP achieves the currently best-achievable result, i.e., for the unbounded MSCP, SEIP efficiently obtains an H_k-approximation ratio (where H_k is the harmonic number of the cardinality of the largest set), the asymptotic lower bound [Feige, 1998], and for the minimum k-set cover problem, SEIP efficiently yields an $(H_k - \frac{k-1}{8k^9})$-approximation ratio, the currently best-achievable quality [Hassin and Levin, 2005]. Moreover, on a subclass of the problem, we demonstrate how SEIP can overcome the difficulty that limits the greedy algorithm.

In the remainder of the paper, Section 2 introduces SEIP, Section 3 presents the main results, and Section 4 concludes.

*This paper is an extended abstract of the AI Journal publication [Yu *et al.*, 2012]. Full proofs can be found in the journal paper.

2 SEIP

We use bold small letters such as $\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ to represent vectors. Denote $[m]$ as the set $\{1, 2, \ldots, m\}$ and 2^S as the power set of S, which consists of all subsets of S. Denote $H_n = \sum_{i=1}^{n} \frac{1}{i}$ for the nth harmonic number. Note that $H_n \sim \ln n$ since $\ln n \leq H_n \leq \ln n + 1$. We consider minimization problems as follows.

Definition 1 (Minimization problem)
Given an evaluation function f and a set of feasibility constraints $\mathcal{C}$, find a solution $\boldsymbol{x} \in \{0,1\}^n$ that minimizes $f(\boldsymbol{x})$ while satisfying constraints in $\mathcal{C}$. A problem instance can be specified by its parameters $(n, f, \mathcal{C})$.

A solution is called *feasible* if it satisfies all the constraints in $\mathcal{C}$; otherwise, it is called an *infeasible* solution, or a *partial* solution in this paper. Here, we assume that the evaluation function f is defined on the full input space, that is, it evaluates all possible solutions regardless of their feasibility.

The SEIP framework is depicted in Algorithm 1. It uses an *isolation function* μ to isolate solutions. For some integer q, the function μ maps a solution to a subset of $[q]$. If and only if two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are mapped to subsets with the same cardinality, that is, $|\mu(\boldsymbol{x}_1)| = |\mu(\boldsymbol{x}_2)|$, the two solutions compete with each other. In that case, we say the two solutions are in the same *isolation*, and there are at most $q+1$ isolations since the subsets of $[q]$ have $q+1$ different cardinalities.

Algorithm 1 (SEIP)
Given a minimization problem $(n, f, \mathcal{C})$, an isolation function μ encodes each solution as a binary vector of length m. SEIP minimization of f with respect to constraint $\mathcal{C}$ consists of the following steps.

```
1:  P ← {x^∅ = (0, 0, ..., 0)}
2:  while stop ≠ true do
3:     Choose x ∈ P uniformly at random
4:     Mutate x to generate x'
5:     if ∀x ∈ P : superior(x, x') = false then
6:        Q ← {x ∈ P | superior(x', x) = true}
7:        P ← P ∪ {x'} − Q
8:     end if
9:  end while
10: return the best feasible solution in P
```

where the `superior` *function of two solutions determines whether one solution is superior to the other. This is defined as follows:* $\mathtt{superior}(\boldsymbol{x}, \boldsymbol{y})$ *is* `true` *if both of the following rules are satisfied:*
1) $|\mu(\boldsymbol{x})| = |\mu(\boldsymbol{y})|$
2) $f(\boldsymbol{x}) < f(\boldsymbol{y})$, or $f(\boldsymbol{x}) = f(\boldsymbol{y})$ but $|\boldsymbol{x}| < |\boldsymbol{y}|$
and $\mathtt{superior}(\boldsymbol{x}, \boldsymbol{y})$ *is* `false` *otherwise.*

The mutation step commonly employs two operators:
One-bit mutation: Flip one randomly selected bit position of $\boldsymbol{x}$ to generate $\boldsymbol{x}'$.
Bit-wise mutation: Flip each bit of $\boldsymbol{x}$ with probability $\frac{1}{m}$ to generate $\boldsymbol{x}'$.
Usually, one-bit mutation is considered for local searches, while bit-wise mutation is suitable for global searches as it has a positive probability for producing any solution. We denote SEIP with one-bit mutation as LSEIP ("L" for "local") and with bit-wise mutation as GSEIP ("G" for "global"). The `stop` criterion is not described in the definition of SEIP, since EAs are usually used as anytime algorithms. In our analysis we will determine the approximation performance as well as the corresponding computational complexity.

By configuring the isolation function, SEIP can be implemented as a single- or multi-objective EA. When the isolation function puts all solutions in one isolation, all solutions competes with each other and thus SEIP will equal the single objective (1+1)-EA. The isolation function is also be configured to simulate the dominance relationship of multi-objective EAs such as SEMO/GSEMO [Laumanns *et al.*, 2002] and DEMO [Neumann and Reichel, 2008]. If we are dealing with k-objective optimization with discrete objective values, a simple approach is to use one of the objective functions, say f_1, as the fitness function and use the combination of the values of the remaining $k-1$ objective functions (say $f_2, \ldots, f_k$) as the isolation functions. In such way, all the non-dominant solutions of a multi-objective EA are also kept in the SEIP population. Hence, SEIP can be viewed as a generalization of multi-objective EAs in terms of the solutions retained.

3 Approximation Performance of SEIP

3.1 General Approximation Results

Given a minimization problem, we denote $\boldsymbol{x}^{OPT}$ as an optimal solution of the problem, and $OPT = f(\boldsymbol{x}^{OPT})$. For a feasible solution $\boldsymbol{x}$, we regard the ratio $\frac{f(\boldsymbol{x})}{OPT}$ as its *approximation ratio*. If the approximation ratio of a feasible solution is upper-bounded by some value r, that is, $1 \leq \frac{f(\boldsymbol{x})}{OPT} \leq r$, the solution is called an *r-approximate solution*. An algorithm that guarantees to find an r-approximate solution for an arbitrary problem instance in polynomial time is an *r-approximation algorithm*. We analyze the approximation ratio obtained by SEIP.

For minimization problems, we consider *linearly additive* isolation functions. μ is a linearly additive isolation function if, for some integer q,

$$\begin{cases} \mu(\boldsymbol{x}) = [q], & \forall \text{ feasible } \boldsymbol{x}, \\ \mu(\boldsymbol{x} \cup \boldsymbol{y}) = \mu(\boldsymbol{x}) \cup \mu(\boldsymbol{y}), & \forall \text{ solutions } \boldsymbol{x} \text{ and } \boldsymbol{y}. \end{cases}$$

The quality of a feasible solution is measured in terms of the approximation ratio. To measure the quality of a partial (infeasible) solution, we define a *partial reference function* and *partial ratio* as follows.

Definition 2 (Partial reference function)
Given a set $[q]$ and a value v, a function $\mathcal{L}_{[q],v} : 2^{[q]} \to \mathbb{R}$ is a partial reference function if $\mathcal{L}_{[q],v}([q]) = v$, and $\mathcal{L}_{[q],v}(R_1) = \mathcal{L}_{[q],v}(R_2)$ for all $R_1, R_2 \subseteq [q]$ such that $|R_1| = |R_2|$.

For a minimization problem with optimal cost OPT and an isolation function μ mapping feasible solutions to the set $[q]$, we denote a partial reference function with respect to $[q]$ and OPT as $\mathcal{L}_{[q],OPT}$. When the problem and the isolation function are clear, we omit the subscripts and simply denote the partial reference function as $\mathcal{L}$.

We then can define the *partial ratio* of a solution x as $\text{p-ratio}(\boldsymbol{x}) = \frac{f(\boldsymbol{x})}{\mathcal{L}(\mu(\boldsymbol{x}))}$ and the *conditional partial ratio* of

$\boldsymbol{y}$ conditioned on $\boldsymbol{x}$ as $\text{p-ratio}(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{x})}{\mathcal{L}(\mu(\boldsymbol{y})|\mu(\boldsymbol{x}))}$, under the given minimization problem $(n, f, \mathcal{C})$, the isolation function μ, and the partial reference function $\mathcal{L}$, where $f(\boldsymbol{y} \mid \boldsymbol{x}) = f(\boldsymbol{x} \cup \boldsymbol{y}) - f(\boldsymbol{x})$ and $\mathcal{L}(\mu(\boldsymbol{y}) \mid \mu(\boldsymbol{x})) = \mathcal{L}(\mu(\boldsymbol{y}) \cup \mu(\boldsymbol{x})) - \mathcal{L}(\mu(\boldsymbol{x}))$. The partial ratio is an extension of the approximation ratio, noticing that the partial ratio for a feasible solution equals its approximation ratio. We have two properties of the partial ratio: one is that it is non-increasing in SEIP, and the other is its decomposability. As the consequence, we can show the following theorem.

Theorem 1
Given a minimization problem $(n, f, \mathcal{C})$ and an isolation function μ mapping to subsets of $[q]$, assume that every solution is encoded in an m-length binary vector. For some constant $r \geq 1$ with respect to a corresponding partial reference function $\mathcal{L}$, if

1. *$\text{p-ratio}(\boldsymbol{x}^{\emptyset}) \leq r$,*
2. *for every partial solution $\boldsymbol{x}$ such that $\text{p-ratio}(\boldsymbol{x}) \leq r$, SEIP takes $\boldsymbol{x}$ as the parent solution and generates an offspring partial solution $\boldsymbol{y}$ such that $\mu(\boldsymbol{x}) \subset \mu(\boldsymbol{x} \cup \boldsymbol{y})$ and $\text{p-ratio}(\boldsymbol{y} \mid \boldsymbol{x}) \leq r$ in polynomial time in q and m,*

then SEIP finds an r-approximate solution in polynomial time in q and m.

Theorem 1 reveals how SEIP can work to achieve an approximate solution. Starting from the empty set, SEIP uses its mutation operator to generate solutions in all isolations, and finally generates a feasible solution. During the process, SEIP repeatedly optimizes the partial ratio in each isolation by finding partial solutions with better conditional partial ratios. Since the feasible solution can be viewed as a composition of a sequence of partial solutions from the empty set, the approximation ratio is related to the conditional partial ratio of each partial solution. Moreover, for GSEIP, we can further have a tighter approximation ratio by Theorem 2.

Definition 3 (Non-negligible path)
Given a minimization problem $(n, f, \mathcal{C})$ and an isolation function μ mapping to subsets of $[q]$, assume that every solution is encoded in an m-length binary vector. A set of solutions N is a non-negligible path *with* ratios *$\{r_i\}_{i=0}^{q-1}$ and* gap *c if $\boldsymbol{x}^{\emptyset} \in N$ and, for every solution $\boldsymbol{x} \in N$, there exists a solution $\boldsymbol{x}' = (\boldsymbol{x} \cup \boldsymbol{y}^{+} - \boldsymbol{y}^{-}) \in N$, where the pair of solutions $(\boldsymbol{y}^{+}, \boldsymbol{y}^{-})$ satisfies*

1. *$1 \leq |\boldsymbol{y}^{+}| + |\boldsymbol{y}^{-}| \leq c$,*
2. *$f(\boldsymbol{y}^{+} - \boldsymbol{y}^{-} \mid \boldsymbol{x}) \leq r_{|\mu(\boldsymbol{x})|} \cdot OPT$,*
3. *if $|\mu(\boldsymbol{x})| < q$, $|\mu(\boldsymbol{x} \cup \boldsymbol{y}^{+} - \boldsymbol{y}^{-})| > |\mu(\boldsymbol{x})|$.*

Theorem 2
Given a minimization problem $(n, f, \mathcal{C})$ and an isolation function μ mapping to subsets of $[q]$, assume that every solution is encoded in an m-length binary vector. If there exists a non-negligible path with ratios $\{r_i\}_{i=0}^{q-1}$ and gap c, then GSEIP finds an $(\sum_{i=0}^{q-1} r_i)$-approximate solution in expected time $O(q^2 m^c)$.

Using Theorem 2 to prove the approximation ratio of GSEIP for a specific problem, we need to find a non-negligible path and then calculate the conditional evaluation function for each jump on the path. One way of finding a non-negligible path for a problem is to follow an existing algorithm for the problem. This will lead to a proof that the EA can achieve the same approximation ratio by simulating the existing algorithm. Similar ideas have been used to confirm that EAs can simulate dynamic programming [Doerr *et al.*, 2009]. In addition, note that the concrete form of the partial reference function is not required in the theorem.

3.2 Approximation Results on MSCP

In the definition of the minimization problem, solutions are represented in binary vectors. When the aim of a minimization problem is to find a subset from a universal set, we can equivalently use a binary vector to represent a subset, where each element of the vector indicates the membership of a corresponding element of the universe set. We denote $\boldsymbol{x}^{\emptyset} = (0, \ldots, 0)$ as the vector corresponding to the empty set.

We investigate the minimum set cover problem (MSCP), which is an NP-hard problem.

Definition 4 (Minimum set cover problem (MSCP))
Given a set of n elements $U = [n]$ and a collection $C = \{S_1, S_2, \ldots, S_m\}$ of m nonempty subsets of U, where each S_i is associated with a positive cost $w(S_i)$, find a subset $X^* \subseteq C$ such that $\sum_{S \in X^*} w(S)$ is minimized with respect to $\bigcup_{S \in X^*} S = U$.

Definition 5 (Minimum k-set cover problem)
An MSCP $(n, \boldsymbol{w}, C, U)$ is a k-set cover problem if, for some constant k, it holds that $|S| \leq k$ for all $S \in C$, denoted as $(n, \boldsymbol{w}, C, U, k)$.

To apply SEIP to the MSCP, we use the fitness function $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$, which has the objective of minimizing the total weight. For a solution $\boldsymbol{x}$, we denote $R(\boldsymbol{x}) = \cup_{S \in \boldsymbol{x}(C)} S$, that is, $R(\boldsymbol{x})$ is the set of elements covered by $\boldsymbol{x}$. We use the isolation function $\mu(\boldsymbol{x}) = R(\boldsymbol{x})$, which makes two solutions compete only when they cover the same number of elements owing to the effect of the isolation function. We could regard a partial reference function $\mathcal{L}$ of $\boldsymbol{x}$ to be the minimum price that optimal solutions pay for covering the same number of elements covered by $\boldsymbol{x}$, although it is not necessary to calculate the partial reference function.

For the (unbounded) MSCP, we show that SEIP achieves an H_k-approximation ratio in Theorem 3, where k is the cardinality of the largest set.

Theorem 3
Given an MSCP $(n, \boldsymbol{w}, C, U)$ where $|C| = m$, LSEIP and GSEIP find an H_k-approximate solution in expected time $O(mn^2)$, where k is the size of the largest set in C.

The theorem is proved via the construction of a non-negligible path following the greedy algorithm [Chvátal, 1979]. It has been proven that a multi-objective EA named SEMO [Friedrich *et al.*, 2010] achieves the same approximation ratio, which is known as the asymptotic lower bound for the problem. The theorem confirms that SEIP can simulate multi-objective EAs in terms of the solution retained, so that SEIP is able to achieve the approximation ratio obtained by multi-objective EAs.

For the minimum k-set cover problem, we show that GSEIP achieves the same $(H_k - \frac{k-1}{8k^9})$-approximation ratio as the GAWW algorithm [Hassin and Levin, 2005] in Theorem 4, which is the current best-achievable result.

Theorem 4
Given a minimum k-set cover problem $(n, \boldsymbol{w}, C, U, k)$, where $|C| = m$, we denote $R(\boldsymbol{x}) = \cup_{S \in \boldsymbol{x}} S$. GSEIP using $\mu(\boldsymbol{x}) = R(\boldsymbol{x})$ finds an $(H_k - \frac{k-1}{8k^9})$-approximate solution in expected time $O(m^{k+1}n^2)$.

The theorem is proved via the construction of a non-negligible path following the GAWW algorithm. This result reveals that when the problem is bounded, which is very likely in real-world situations, GSEIP can yield a better approximation ratio than in the unbounded situation. Since the greedy algorithm cannot achieve an approximation ratio lower than H_n, the result also implies that GSEIP has essential non-greedy behaviors for approximations.

For a comparison of the greedy algorithm, LSEIP and GSEIP, we study on a subclass of the minimum k-set cover problem, denoted as problem I. Problem I is a minimum k-set cover problem $(n, \boldsymbol{w}, C, U)$ constructed as follows. Note that $n = kL$ for some integer L. The optimal solution consists of L non-overlapping sets $\{S_i^*\}_{i=1}^L$, each of which contains k elements in U. Imagine that elements in each S_i^* are ordered and let S_{ij} contain only the jth element of set S_i^*. The collection of sets C consists of all sets of S_i^* and S_{ij}. Assign each S_i^* weight $1+\epsilon$ for some $\epsilon > 0$, and assign each S_{ij} weight $1/j$. Figure 1 illustrates Problem I.

Proposition 1
Given an arbitrary value $\xi > 0$, the approximation ratio of the greedy algorithm for problem I is lower-bounded by $H_k - \xi$.

Proposition 2
With probability of at least $1 - \frac{1}{k+1}$, LSEIP finds a $\left(H_k - \frac{k}{n}(H_k - 1)\right)$-approximate solution for problem I in $O\left((1+\frac{1}{k})n^3\right)$ expected steps.

Proposition 3
GSEIP finds the optimal solution for problem I in $O(\frac{1}{k} \cdot (1+\frac{1}{k})^{k+1} \cdot n^{k+3})$ expected steps.

For GSEIP, we can also derive Proposition 4 that discloses the time-dependent approximation performance of GSEIP, which illustrates an interesting property: compared with the greedy algorithm, whose performance cannot be improved given extra time, SEIP always seeks better approximate solutions. Users may allocate more running time in a trade-off for better solutions.

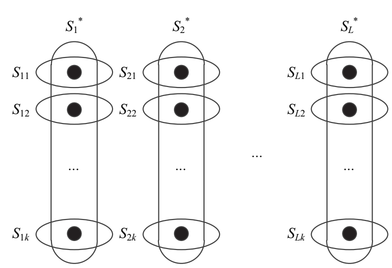

Figure 1: Subclass I of the minimum k-set cover problem.

Proposition 4
GSEIP finds an $\left(H_k - c\frac{k}{n}(H_k - 1)\right)$-approximate solution for problem I in $O(mn^2 + cn^2m^{k+1}/k)$ expected steps, for $c = 0, 1, \ldots, \frac{n}{k}$.

4 Conclusion

We studied the approximation performance of an EA framework, SEIP, introducing an isolation function to manage competition among solutions, which can be configured as a single- or multi-objective EA. We analyzed the approximation performance of SEIP using the partial ratio and obtained a general characterization of the SEIP approximation behavior. Our analysis discloses that SEIP can achieve a guaranteed approximate solution: it tries to optimize the partial ratio of solutions in every isolation by finding good partial solutions, then these partial solutions form a feasible solution with a good approximation ratio. We then took the MSCP as a benchmark problem. Our analysis shows that for the unbounded MSCP, SEIP achieves an H_k-approximation ratio (where H_k is the harmonic number of the cardinality of the largest set), the asymptotic lower bound. For the minimum k-set cover problem, it achieves an $(H_k - \frac{k-1}{8k^9})$-approximation ratio, the current best-achievable result [Hassin and Levin, 2005], which is beyond the ability of the greedy algorithm. Moreover, for a subclass of the MSCP, we show how SEIP with either one-bit or bit-wise mutation can overcome the difficulty that limits the greedy algorithm.

We have shown that SEIP can simulate the greedy algorithm. Since the greedy algorithm is a general scheme for approximations and has been analyzed for many problems, SEIP analysis can be extended to cases for which the greedy algorithm has been applied, and therefore we can easily show that SEIP is a $\frac{1}{k}$-approximation algorithm for k-extensible systems [Mestre, 2006], including b-matching, maximum profit scheduling and maximum asymmetric TSP problems. Moreover, to prove the approximation ratio of SEIP for the minimum k-set cover problem, we used SEIP to simulate the GAWW algorithm, which implies that SEIP also has extra behaviors that provide opportunities to exceed the greedy algorithm. Limitations of SEIP can also be found from Theorem 2. When SEIP is required to flip a number of bits at a time, which depends on n, to achieve a good solution, we will find that the one-bit mutation is limited since it only flips one bit at a time, and the bit-wise mutation is also limited as it requires exponential time.

Our theoretical analysis suggests that EAs can achieve solutions with guaranteed performance. We believe that guaranteed and better approximation performance can be achieved by better EA design in the future.

Acknowledgements

We thank Dr. Per Kristian Lehre and Dr. Yitong Yin for helpful discussions. This work was partly supported by the National Fundamental Research Program of China (2010CB327903), the National Science Foundation of China (61073097, 61021062), EPSRC (UK) (EP/I010297/1), and an EU FP7-PEOPLE-2009-IRSES project under Nature Inspired Computation and its Applications (NICaiA) (247619).

References

[Auger and Doerr, 2011] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics - Foundations and Recent Developments*. World Scientific, Singapore, 2011.

[Bäck, 1996] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.

[Benkhelifa *et al.*, 2009] E. Benkhelifa, G. Dragffy, A.G. Pipe, and M. Nibouche. Design innovation for real world applications, using evolutionary algorithms. In *Proceedings of 2009 IEEE Congress on Evolutionary Computation*, pages 918 –924, Trondheim, Norway, 2009.

[Chvátal, 1979] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[Doerr *et al.*, 2009] B. Doerr, A. V. Eremeev, C. Horoba, F. Neumann, and M. Theile. Evolutionary algorithms and dynamic programming. In *Proceedings of GECCO'09*, pages 771–778, Montreal, Canada, 2009.

[Feige, 1998] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[Friedrich *et al.*, 2009] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt. Analyses of simple hybrid algorithms for the vertex cover problem. *Evolutionary Computation*, 17(1):3–19, 2009.

[Friedrich *et al.*, 2010] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt. Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation*, 18(4):617–633, 2010.

[Giel and Wegener, 2003] O. Giel and I. Wegener. Evolutionary algorithms and the maximum matching problem. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 415–426, Berlin, Germany, 2003.

[Hassin and Levin, 2005] R. Hassin and A. Levin. A better-than-greedy approximation algorithm for the minimum set cover problem. *SIAM Journal on Computing*, 35(1):189–200, 2005.

[He and Yao, 2001] J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artifical Intelligence*, 127(1):57–85, 2001.

[He and Yao, 2003] J. He and X. Yao. An analysis of evolutionary algorithms for finding approximation solutions to hard optimisation problems. In *Proceedings of 2003 IEEE Congress on Evolutionary Computation*, pages 2004–2010, Canberra, Australia, 2003.

[Hornby *et al.*, 2006] G. S. Hornby, A. Globus, D. S. Linden, and J. D. Lohn. Automated antenna design with evolutionary algorithms. In *Proceedings of 2006 American Institute of Aeronautics and Astronautics Conference on Space*, pages 19–21, San Jose, CA, 2006.

[Koza *et al.*, 2003] J. R. Koza, M. A. Keane, and M. J. Streeter. What's AI done for me lately? Genetic programming's human-competitive results. *IEEE Intelligent Systems*, 18(3):25–31, 2003.

[Laumanns *et al.*, 2002] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Proceedings of PPSN'02*, pages 44–53, London, UK, 2002.

[Mestre, 2006] J. Mestre. Greedy in approximation algorithms. In *Proceedings of the 14th Annual European Symposium on Algorithms*, pages 528–539, Zurich, Switzerland, 2006.

[Neumann and Reichel, 2008] F. Neumann and J. Reichel. Approximating minimum multicuts by evolutionary multi-objective algorithms. In *Proceedings of PPSN'08*, pages 72–81, Dortmund, Germany, 2008.

[Neumann and Wegener, 2005] F. Neumann and I. Wegener. Minimum spanning trees made easier via multi-objective optimization. In *Proceedings of GECCO'05*, pages 763–769, Washington DC, 2005.

[Neumann and Wegener, 2007] F. Neumann and I. Wegener. Can single-objective optimization profit from multiobjective optimization? In J. Knowles, D. Corne, and K. Deb, editors, *Multiobjective Problem Solving from Nature - From Concepts to Applications*, pages 115–130. Springer, Berlin, Germany, 2007.

[Neumann and Witt, 2010] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization - Algorithms and Their Computational Complexity*. Springer-Verlag, Berlin, Germany, 2010.

[Neumann *et al.*, 2008] F. Neumann, J. Reichel, and M. Skutella. Computing minimum cuts by randomized search heuristics. In *Proceedings of GECCO'08*, pages 779–786, Atlanta, GA, 2008.

[Oliveto *et al.*, 2009] P. Oliveto, J. He, and X. Yao. Analysis of the (1+1)-EA for finding approximate solutions to vertex cover problems. *IEEE Transactions on Evolutionary Computation*, 13(5):1006–1029, 2009.

[Scharnow *et al.*, 2002] J. Scharnow, K. Tinnefeld, and I. Wegener. Fitness landscapes based on sorting and shortest paths problems. In *Proceedings of PPSN'02*, pages 54–63, Granada, Spain, 2002.

[Witt, 2005] C. Witt. Worst-case and average-case approximations by simple randomized search heuristics. In *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science*, pages 44–56, Stuttgart, Germany, 2005.

[Yu and Zhou, 2008] Y. Yu and Z.-H. Zhou. A new approach to estimating the expected first hitting time of evolutionary algorithms. *Artificial Intelligence*, 172(15):1809–1832, 2008.