Approximating Optimization Problems using EAs on Scale-Free Networks

Ankit Chauhan
Hasso Plattner Institute
Potsdam, Germany 14482
ankit.chauhan@hpi.de

Tobias Friedrich
Hasso Plattner Institute
Potsdam, Germany 14482
tobias.friedrich@hpi.de

Francesco Quinzan
Hasso Plattner Institute
Potsdam, Germany 14482
francesco.quinzan@hpi.de

ABSTRACT

It has been experimentally observed that real-world networks follow certain topological properties, such as small-world, power-law etc. To study these networks, many random graph models, such as Preferential Attachment, have been proposed.

In this paper, we consider the deterministic properties which capture power-law degree distribution and degeneracy. Networks with these properties are known as scale-free networks in the literature. Many interesting problems remain NP-hard on scale-free networks. We study the relationship between scale-free properties and the approximation-ratio of some commonly used evolutionary algorithms.

For the Vertex Cover, we observe experimentally that the (1+1) EA always gives the better result than a greedy local search, even when it runs for only $\mathcal{O}(n \log(n))$ steps. We give the construction of a scale-free network in which a multi-objective algorithm and a greedy algorithm obtain optimal solutions, while the (1+1) EA obtains the worst possible solution with constant probability.

We prove that for the Dominating Set, Vertex Cover, Connected Dominating Set and Independent Set, the (1+1) EA obtains constant-factor approximation in expected run time $\mathcal{O}(n \log(n))$ and $\mathcal{O}(n^4)$ respectively. Whereas, GSEMO gives even better approximation than (1+1) EA in expected run time $\mathcal{O}(n^3)$ for Dominating Set, Vertex Cover and Connected Dominating Set on such networks.

ACM Reference format:
Ankit Chauhan, Tobias Friedrich, and Francesco Quinzan. 2017. Approximating Optimization Problems using EAs on Scale-Free Networks. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071257

1 INTRODUCTION

Evolutionary Algorithms (EAs) are bio-inspired randomized optimization techniques and have been shown to be very successful when applied to combinatorial optimization problems. The success of EAs to solve such problems have attracted lots of attention and there has been extensive research in recent years to understand the behavior of these algorithms. In early research, the main concern was to analyze the run time of EAs like the (1+1) EA for artificial pseudo-boolean functions [7, 12] as well as for some combinatorial optimization problems [10, 18, 19]. Most of the results considered the exact optimization of the function, however many combinatorial optimization problems are NP-hard. This means one cannot hope for an exact solution in polynomial time unless P=NP. Thus, the goal of EAs are to obtain a good approximation of an optimal solution within a certain amount of time instead of finding a exact optimal solution. In previous years, some progress has been made for worst-case approximation analysis of the EAs for the combinatorial problems that can be achieved in a polynomial number of steps in expectation [9, 15, 28]. However, for real-world instances, the solution quality heavily depends on the underlying structural properties. Consider the example of the traveling salesman problem. If the graph instance follows the triangle inequality than the TSP has $\frac{3}{2}$-approximate. On the other hand, TSP for the general graphs cannot be approximate within constant factor unless P=NP [27]. This means that on considering the topological properties of the instances in the algorithms can obtain a good quality solution for those problems which are even NP-hard to approximate. In practice, interesting optimization problems deal with real-world network instances. Therefore, much effort has been made to identify properties of real-world networks to bridge the gap between theory and practice. A wide range of real-world networks, like Internet topologies, the web, power grids, protein-protein interaction graphs, social networks and many other networks high clustering coefficient, small diameter, almost power-law degree distribution and so on [1, 8, 14, 16, 20, 24, 26]. Thus, considering these properties in the analysis of algorithms gives a better understanding about the behavior of the algorithm on real-world networks.

Recently, Cohen et al. [6] consider stability properties of real-world instances and show that simple local search works well for clustering problems on such instances. In this work, we follow the same line of research. Instead of considering stability properties, we consider networks with power-law degree distribution, that is, scale-free networks. A power-law degree distribution means that the number of vertices of degree k in the network is proportional to $k^{-\beta}$, where $\beta > 1$ is the power-law exponent, a constant inherent to the network. Real-world networks, however, usually exhibit slight

GECCO '17, Berlin, Germany
 978-1-4503-4920-8/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071257

Problem	PLB-graphs		General graphs	
	run time	ratio	run time	ratio
MVC	$\mathcal{O}(n \log n)$	$\Theta(1)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$
MDS	$\mathcal{O}(n \log n)$	$\Theta(1)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$
CDS	$\mathcal{O}(n \log n)$	$\Theta(1)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$
MIS	$\mathcal{O}(n^4)$	$\Theta(1)$	$\mathcal{O}(n^4)$	$\frac{\Delta+1}{2}$

Table 1: Comparison of the approximation ratios achieved by (1+1) EA on networks with PLB-U property and power-law exponent $\beta > 2$ and on general graphs. While on general graphs, (1+1) EA takes polynomial many steps in expectation to achieve polynomial-approximation, (1+1) EA achieve constant-approximation on networks with PLB-U property and power-law exponent $\beta > 2$ within expected polynomial many steps.

Problem	PLB-graphs		General graphs	
	run time	ratio	run time	ratio
MVC	$\mathcal{O}(n^3)$	$\Theta(1)$	$\mathcal{O}(n^3 \log n)$	$\mathcal{O}(\log n)$
MDS	$\mathcal{O}(n^3)$	$\Theta(1)$	$\mathcal{O}(n^3 \log n)$	$\mathcal{O}(\log n)$
CDS	$\mathcal{O}(n^3)$	$\Theta(1)$	$\mathcal{O}(n^3 \log n)$	$\mathcal{O}(\log n)$

Table 2: Comparison of the approximation ratios achieved by GSEMO on networks with PLB properties and power-law exponent $\beta > 2$ and on general graphs. While on general graphs, GSEMO takes polynomial many fitness evaluation in expectation to achieves logarithmic-approximation, GSEMO achieve constant-approximation on networks with PLB properties and power-law exponent $\beta > 2$ within polynomial many fitness evaluation in expectation.

deviation from the power-laws. To allow this deviation Brach et al. [3] identify the deterministic properties for real-world networks that capture the degree distribution of many real-world networks. They define buckets containing vertices of degrees $\left[2^i, 2^{i+1}\right)$.If the number of vertices in each bucket is at most as high as for a power-law degree sequence, a network is said to be *power-law bounded*. We denote this property by PLB-U. Brach et al. also define *PLB neighborhoods*. A network has PLB neighborhoods if the degree distribution of degrees of neighbor of each vertices in the network also obey the power-law. In other words, for every vertex of the degree k this property gives an upper-bound on the number of neighbors of the degree at least k. We denote this property by PLB-N. The PLB-N property is closely related to the degeneracy of the graphs. We say a graph is d-degenerate if every subgraph has vertex of degree at most d. Degeneracy of the network is d upper-bounded by the maximum value of PLB-N among all buckets. Brach et al. exploit these properties and prove that algorithms for page rank, maximum clique counting triangle etc. run faster on power-law bounded networks than the known worst case lower bound for the graph. Furthermore, Chauhan et al. [5] show that a simple greedy algorithm obtain a $\Theta(1)$-approximation for Minimum Dominating Set (MDS), Minimum Connected Dominating Set (CDS), Minimum Vertex Cover (MVC), and Maximum independent Set (MIS). On the contrary, a simple greedy algorithm achieves $\Omega(\ln(\Delta))$-approximation for the MDS, CDS and Δ-approximation for MIS on general graphs where Δ is the maximum degree of the graph [25, 27]. The PLB-U property alone captures a much broader class of graph hence Chauhan et al. define PLB-L, which gives lower-bound for the number of vertices in each bucket to restrict networks to real-world networks. Chauhan et al. show that even if the network have all three properties, PLB-(U,L,N) computing MDS, MVC and MIS is APX-Hard. A formal definition of all three properties can be found in section 2.

We look at different instances of networks with PLB-(U,L,N) properties with power-law exponent $\beta > 2$, and experimentally observed that the (1+1) EA always finds a better solution for MVC than the degree-based greedy algorithm on the different instances with these properties. This observation raises a serious question: does the (1+1) EA always give a better solution than greedy on the networks with PLB-(U,L,N)?

We theoretically analyze EAs for well-known NP-hard problems: MDS, MVC, MIS and CDS. The MVC has been studied extensively [21, 22], Friedrich et al. [9] presented a instance of bipartite graph in which (1+1) EA can obtain arbitrary bad approximation in expected polynomial time. To overcome this, Friedrich et al. provide a multi-objective approach for MVC and prove that the GSEMO can obtain an optimal solution in the expected polynomial time for that instance. Also, by using GSEMO they prove the approximation of $\mathcal{O}(\log n)$ for a more general problem, minimum set cover. Bäck and Khuri [2] give a single objective EA for MIS and claim its superiority by experimental observations. Peng [23] analyzes the (1+1) EA for the maximum independent set, and proves that the (1+1) EA obtains a $\frac{\Delta+1}{2}$-approximation within expected runtime of $\mathcal{O}(n^4)$. We analyze the (1+1) EA for all four above-mentioned problems. We prove that for MDS, MVC and CDS the (1+1) EA gives $\mathcal{O}(1)$-approximation within the expected runtime $\mathcal{O}(n \log n)$. Contrary to the experiments results, we give a worst case example of network with PLB-(U,L,N) properties where (1+1) EA can obtain the worst possible solution with $\mathcal{O}(1)$-probability and the greedy algorithm obtains the optimal solution. Since the (1+1) EA can obtain such a bad solution, we analyze the multi-objective EAs on networks with PLB-(U,L,N). We study the GSEMO for the MDS, MVC and CDS on the networks with PLB-(U,L,N). We prove that the GSEMO gives a better approximation than the (1+1) EA in expected runtime of $\mathcal{O}(n^3)$. Also, as a byproduct of the analysis of GSEMO, we present an improvement over the result of Chauhan et al. [5] for the approximation on MDS and CDS. A summary of these results is given in Table 1 and Table 2.

Algorithm 1 $(\mu+1)$ EA

choose population $P_\mu \subseteq \{0,1\}$ u.a.r. of size μ
while convergence criterion not met **do**
  choose *parent* $x \subseteq P_\mu$ u.a.r.
  create *offspring* y by flipping bits of x w.p. $\frac{1}{n}$
  discard weakest element in $P_\mu \cup \{y\}$
return fittest element in P_μ

Algorithm 2 GSEMO

choose $x \in \{0,1\}^n$ uniformly at random
add x to Pareto front P
while convergence criterion not met **do**
  Choose $x \in P$ uniformly at random
  create y by flipping bits of x with probability $\frac{1}{n}$
  if y is not dominated by any point in P **then**
    add y to P
    delete all solutions in P dominated by y.
return P

2 PRELIMINARIES

In this section, we introduce the algorithms considered in this paper, and define the basic definitions that are used later in the proofs. We consider undirected graphs $G = (V, E)$ without loops, where V is the set of vertices and E is the set of edges in the graph with $n := |V|$. Throughout the paper we use $\deg(v)$ to denote the degree of the vertex $v \in V$, Δ for maximum degree of the graph and OPT for the optimal solution set. For a $S \subseteq V$ we then define $vol(S) = \sum_{i \in S} \deg(i)$.

2.1 Algorithms

The $(\mu+1)$ EA is a randomized algorithm inspired by the process of natural selection (cf. Algorithm 1). After an initial *population* of size μ is chosen *uniformly at random (u.a.r)*, the $(\mu+1)$ EA chooses a *parent* x from P_μ u.a.r. An offspring y is then generated, by flipping all bits of x independently with probability $1/n$. The fitness is then computed for all elements of $P_\mu \cup \{x\}$, and the weakest one is discarded. The (1+1) EA is an instance of the $(\mu+1)$ EA, with $\mu = 1$. The (1+1) EA is one of the simplest instances of a single-objective evolutionary algorithm.

The GSEMO algorithm is a multi-objective evolutionary strategy (cf. Algorithm 2). As in the case of the (1+1) EA, this heuristic chooses an initial solution u.a.r. from the objective space, and stores it in the Pareto front. An element x is then chosen u.a.r. from the Pareto front P, and a new solution y is then computed from x by flipping each bit independently *with probability (w.p.)* $1/n$. If y is not strongly dominated by any other solution in the Pareto front, the y is saved as a new solution, and all elements which are strongly dominated by y are discarded.

Local search algorithms are iterative improvement algorithms (cf. Algorithm 3). Again, the first step is to choose a solution u.a.r. Then, a second solution is chosen u.a.r. in a neighborhood of x. Again, the two solutions are compared and the best one is stored in memory. The last algorithm we take into account for the analysis is a deterministic one (cf. Algorithm 4). This algorithm is specifically designed to find a minimum vertex cover of an input graph. It iteratively adds nodes, which have highest degree to the cover, until all edges are covered. For the randomized algorithms the run time is always counted in terms of the number of fitness evaluations, and for the deterministic processes the run time is given in the number of steps.

Algorithm 3 Local Search Algorithm

choose $x \in \{0,1\}^n$ u.a.r.
while (termination condition not satisfied) **do**
  choose y in a neighbourhood of x u.a.r.
  if $f(y) > f(x)$ **then**
    $x \leftarrow y$
return x

Algorithm 4 Greedy Algorithm for vertex cover on $G = (V, E)$

$S \leftarrow \emptyset$
while not all edges are covered **do**
  add vertex $s \in V \setminus S$ with highest degree to S
return S

2.2 Technical definitions

Particularly important for the analysis is the power-law distribution hypothesis. We formally frame this concept, by giving some related definitions (cf. Brach et. al. [3] and Chauhan et. al. [5]). Many of these concepts have been informally introduced in the introduction.

Definition 2.1. A graph G is power law upper-bounded (PLB-U) for some parameters $1 < \beta = \mathcal{O}(1)$ and $t \geq 0$, and universal constant $c_1 > 0$ if for every integer $d \geq 0$, the number of vertices v, such that $\deg(v) \in \left[2^d, 2^{d+1}\right)$ is at most

$$c_1 n (t+1)^{\beta-1} \sum_{i=2^d}^{2^{d+1}-1} (i+t)^{-\beta}.$$

Definition 2.2. A graph G is power law lower-bounded (PLB-L) for some parameters $1 < \beta = \mathcal{O}(1)$, $t \geq 0$ and universal constant $c_2 > 0$ if for every integer $\lfloor \log d_{min} \rfloor \leq d \leq \lfloor \log \Delta \rfloor$, the number of vertices v, such that $\deg(v) \in \left[2^d, 2^{d+1}\right)$ is at least

$$c_2 n (t+1)^{\beta-1} \sum_{i=2^d}^{2^{d+1}-1} (i+t)^{-\beta}.$$

We also consider the definition of PLB neighbourhood, given below. Again, we follow the work of Brach et. al. [3].

Definition 2.3. A graph G has PLB neighborhoods (PLB-N) if for every vertex v of degree k, the number of neighbors of v of degree at least k is at most

$c_3 \max\left(\log n, (t+1)^{\beta-2} k \sum_{i=k}^{n-1} i(i+t)^{-\beta}\right)$ where $c_3 > 0$ is constant and $1 < \beta = \mathcal{O}(1)$, $t \geq 0$ are parameters.

We conclude with a definition that describes how the solution of an optimization problem is approximated.

Definition 2.4. An algorithm is an *α-approximation* for problem P if it produces a solution set S with $\alpha \geq \frac{|S|}{|\mathrm{OPT}|}$ if P is a minimization problem and with $\alpha \geq \frac{|\mathrm{OPT}|}{|S|}$ if P is a maximization problem.

Useful for the analysis is the following lemma. The lemma gives the upper-bound on the $\sum_{i \in S} h(\deg(S))|S|$ by using the highest possible volume in the PLB-U graph. The reader may find a proof in Chauhan et. al. [5].

Lemma 2.5 (Potential Volume Lemma). *Let G be a graph without loops and with the PLB-U property for some $\beta > 2$, some constant $c_1 > 0$ and some constant $t \geq 0$. Let S be a solution set for which we can define a function $g\colon \mathsf{R}^+ \to \mathsf{R}$ as continuously differentiable and $h(x) := g(x) + C$ for some constant C such that*

(1) *g non-decreasing,*
(2) *$g(2x) \leq c \cdot g(x)$ for all $x \geq 2$ and some constant $c > 0$,*
(3) *$g'(x) \leq \frac{g(x)}{x}$,*

then it holds that $\frac{\sum_{x \in S} h(\deg(x))}{|S|}$ is at most

$$c\left(1 + \frac{\beta-1}{\beta-2}\frac{1}{1-\left(\frac{t+2}{t+1}\right)^{1-\beta}}\right) g\left(\left(c_1 \tfrac{\beta-1}{\beta-2}\tfrac{n}{M} \cdot 2^{\beta-1} \cdot (t+1)^{\beta-1}\right)^{\frac{1}{\beta-2}}\right) + C,$$

where $M(n) \geq 1$ is chosen such that $\sum_{x \in S} \deg(x) \geq M$.

3 THE MINIMUM DOMINATING SET

In this section we consider two problems commonly found in combinatorial optimization. Given a graph $G = (V, E)$, we consider a *minimum dominating set* (MDS) problem, this consists of finding a subset $S \subseteq V$ of minimum size such that for each $v \in V$ either v or a node adjacent to v is in S. Similarly, the *minimum vertex cover* (MVC) problem consists of finding a subset $S \subseteq V$ of minimum size such that each edge $e \in E$ is incident to at least one node from S.

3.1 Single-objective optimization

In order to implement the (1+1) EA to obtain a solution for the MDS and MVC, we first need to give the encoding of solution and the definition of fitness functions. For each $S \subseteq V$, we can use $(x_1, x_2, \cdots, x_n) \in \{0,1\}^n$ to represent it, where n is the number of vertices in the graph $G(V, E)$. We set bit $x_i = 1$ iff. the vertex v_i is in S, otherwise $x_i = 0$. The fitness function is defined as

$$F(x) = n^2 u(x) + |x|_1$$

where $u(x)$ is the number of uncovered nodes, and $|x|_1$ gives the number of 1s in the current solution. The solution space can be partitioned into two sets according to the value of fitness function, as shown below.

$$S_1 = \{x \in \{0,1\}^n : \ x \text{ is infeasible and } F(x) \geq n\}$$

$$S_2 = \{x \in \{0,1\}^n : \ x \text{ is feasible and } F(x) < n\}$$

Because of the weight on $u(x)$, the (1+1) EA first tries to find a feasible solution than it will minimize accordingly. We can use the same fitness function for the vertex cover, with the difference that $u(x)$ returns the number of uncovered edges, instead of uncovered vertices. The following theorem holds.

Theorem 3.1. *For the PLB-U graphs with parameters $\beta > 2, t \geq 0$ consider the quantities*

$$a_{\beta,t} := 1 + \frac{\beta-1}{\beta-2}\frac{1}{1-\left(\frac{t+2}{t+1}\right)^{1-\beta}}$$

$$b_{c_1,\beta,t} := \left(c_1 \tfrac{\beta-1}{\beta-2} \cdot 2^{\beta} \cdot (t+1)^{\beta-1}\right)^{\frac{1}{\beta-2}} + 1$$

Then the (1+1) EA *find a $\left(2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t}\right)$-approx dominating set in expected $\mathcal{O}(n \log n)$.*

Proof. (Phase 1: finding feasible solution) We first observe that the value of $F(x)$ will never increase during run time. Suppose that the current solution x belongs to the S_1 portion of the objective space, let ℓ be the number of vertices that incident on at least one of the uncovered vertices. These ℓ vertices are such that if all are flipped, then the $u(x)$ goes to 0. Now let S_1^i contains all search points with i accepted bits from these ℓ bits. The probability that (1+1) EA mutates the solution such that it changes S_1^i to S_1^{i+1} is given by

$$p_i = \frac{l-i}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq \frac{l-i}{ne}$$

Hence, the expected waiting time until a solution with all ℓ bits is in solution is upper bounded by $\sum_{i=1}^{l} \frac{1}{i} = \mathcal{O}(n \log \ell)$. Also, since $1 \leq l \leq n$ hence expected time in which (1+1) EA produce a feasible solution for dominating set is upper bounded by $\mathcal{O}(n \log n)$.
(Phase 2: removing redundant vertices) Let there be p many vertices which are redundant. By the same argument as in (Phase 1) they can be done $\mathcal{O}(n \log n)$ fitness evaluations. Let A be the solution produced by the (1+1) EA after (Phase 2). This solution is an $\alpha-$approximation of OPT. To give the bound on the approximation ratio α we use Lemma 2.5. Consider a worst case instance where (1+1) EA obtains exactly α-approximation solution. This means,

$$\frac{|A|}{|\mathrm{OPT}|} = \alpha$$

Since the solutions produced after (Phase 2) of the (1+1) EA have no redundant nodes, this means that $|A| \leq n - |\mathrm{OPT}|$.

$$\frac{n - |\mathrm{OPT}|}{|\mathrm{OPT}|} \geq \alpha$$

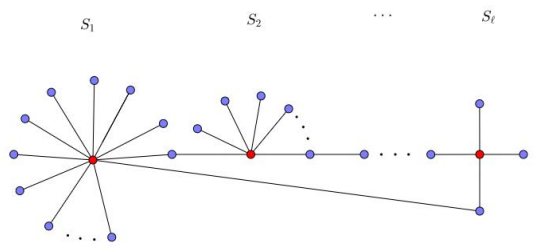

Figure 1: Worst Case graph G for the (1+1) EA with $\Delta = n^{\frac{1}{\beta-1}} + 1$ and $|V| = N = \Theta(n)$. V_1 is the set of red vertices and V_2 is the set of blue vertices. On this graph a greedy search beats the (1+1) EA.

This is the case when every neighbor of the node in the optimal solution is taken as the solution by the (1+1) EA

$$\alpha \leq \frac{n - |\mathrm{OPT}|}{|\mathrm{OPT}|} \leq \frac{\sum_{i \in \mathrm{OPT}} \deg(i)}{|\mathrm{OPT}|} \tag{1}$$

As $\sum_{x \in \mathrm{OPT}} \deg(x) \geq \frac{n}{2}$ than by taking $S = \mathrm{OPT}$ we have that $h(\deg(x)) = \deg(x)$, $g(x) = h(x)$, and that $M = \frac{n}{2}$. Since $g(x)$ follows the all three properties of Lemma 2.5 we get, $\alpha \leq 2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t}$

□

We conclude the theorem 3.1 by giving a worst example for the (1+1) EA. We give a graph G as described below (cf. Figure 1).

(1) for all $j \in \{2, \cdots, \log_2(\Delta - 1)\}$ construct $\left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil$ many star graphs where degree of the center vertex is 2^j.
(2) Let ℓ be the total no. of star graphs, then pick an arbitrary vertex v_i from the star i and add an edge between v_i and center vertex of star $(i+1) \pmod{\ell}$. This step ensures that the graph is connected.

We use the lemma below to prove that the graph G has PLB-(U,L,N) properties (cf. Chauhan et. al. [5]).

Lemma 3.2. *Let $1 \leq a \leq b/2$, for a, b natural numbers, and let $c > 0$ be a constant. Then*

$$a^{-c} \leq \tfrac{c}{1-2^{-c}} \sum_{i=a}^{b-1} i^{-c-1}.$$

From Lemma 3.2 the following theorem follows.

Theorem 3.3. *G is a N vertices graph have PLB-(U,L,N) properties with parameter β, $t = 0$, $c_1 = \frac{2}{1-2^{-\beta+1}}$, $c_2 = (4^{1-\beta} - 5^{1-\beta})p$ and $c_3 = 1$ where $N = \Theta(n)$ and $p = \left(2 \cdot 4^{1-\beta}\left(\frac{4}{\beta-2} + \frac{1}{\beta-1}\right)\right)^{-1}$.*

A proof of this theorem can be found in full version [4]. We use this result to prove the following statement.

Theorem 3.4. *For the PLB-(U,L,N) graph G (1+1) EA with probability $\mathcal{O}(1)$ obtains worst possible dominating set of size $n - |\mathrm{OPT}|$ within the expected polynomial number of steps. In particular, the expected time to produce an dominating set better than $n - |\mathrm{OPT}|$ is exponential.*

The idea of the proof is to first divide the vertices of G in two sets, $V_1 = \{v \in V \mid \deg(v) > 3\}$, $V_2 = \{v \in V \mid \deg(v) \leq 2\}$. We then distinguish two phases. First, we investigate the probability that the (1+1) EA takes V_2 as the dominating set with at least one vertex of $|V_1|$ missing in the solution. Then, we give the lower bound on the probability that all the vertices of the $|V_1|$ will be removed to obtain a local optima. A formal proof can be found in the full version [4]. The corollary below follows from the theorem 3.1.

Corollary 3.5. *For the PLB-U graphs with parameters $\beta > 2$, $t \geq 0$ 1+1-EA produces $\left(2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t}\right)$-approx vertex cover in expected $\mathcal{O}(n \log n)$ time.*

Since a dominating set in the graph G is also a vertex cover, all hereby presented result generalize to the MVC.

3.2 Multi-objective optimization

We consider GSEMO, which is the multi-objective counterpart of the (1+1) EA. The fitness function is defined as

$$F(x) = (u(x), |x|_1)$$

where $u(x)$ is the number of uncovered vertices, and $|x|_1$ the number of 1s in input string. The following lemma holds.

Lemma 3.6. *Given a graph G having a minimum dominating set of size* OPT *and with n_k as the number of non dominated vertices after taking k nodes in the solution, then at each step there exists a node v such that after taking node v in solution there holds $n_k \leq n\left(1 - \frac{1}{|\mathrm{OPT}|}\right)^k$.*

Proof of Lemma 3.6 can be found in the full version [4]. We use this result to prove the following result.

Theorem 3.7. *The expected time until GSEMO has obtained $\ln 2 + \ln\left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) + 1$-approximation for dominating set on PLB-U graphs with $\beta > 2$, $t \geq 0$ is $\mathcal{O}(n^3)$.*

A formal proof can be found in the full version [4].

Corollary 3.8. *The expected time of until GSEMO has obtained the optimal solution on the graph G is $\mathcal{O}(n^3)$.*

Proof. The corollary is a consequence of theorem 3.7 and the fact that the greedy output V_1 as solution of the graph G which is also a optimal solution. □

Corollary 3.9. *The expected time until GSEMO has obtained $\ln 2 + \ln\left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) + 1$-approximation for minimum vertex cover on PLB-U graphs with $\beta > 2$, $t \geq 0$ is $\mathcal{O}(n^3)$.*

Again, the hereby presented results can be easily generalised to the MVC.

4 THE CONNECTED DOMINATING SET

We consider the following problem. Given a Graph $G = (V, E)$, a connected dominating set is a minimum connected subset $S \subseteq V$, such that for each $v \in V$ either v or a neighbor of v is in S. The *connected dominating set* problem (CDS) consists of finding a minimal connected dominating set.

4.1 Single-objective optimization

We study the optimization process for the CDS with the (1+1) EA. The fitness function is defined as

$$F(x) = n^2(u(x) + (p(x) - 1)) + |x|_1,$$

where $u(x)$ is the number of uncovered vertices, and $p(x)$ is the number of connected components in the complete subgraph induced by the chosen solution. Again, the goal is to minimize $F(x)$. It can be observed that the x is a connected dominating set if and only if $u(x) = 0$ and $p(x) = 1$. The algorithm tends to reach a state s.t. $p(x) - 1 + u(x) = 0$, and then it removes unnecessary nodes. Again, the objective space can be divided into two sets.

$$S_1 = \{x \in \{0,1\}^n : \ x \text{ is infeasible and } F(x) \geq n - 2\}$$
$$S_2 = \{x \in \{0,1\}^n : \ x \text{ is feasible and } F(x) < n - 2\}$$

THEOREM 4.1. (1+1) EA *obtains* $\left(2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t}\right)$*-approximation connected dominating set in expected* $\mathcal{O}(n \log n)$ *run time.*

The argument to prove this theorem is identical to the one given for Theorem 3.1.

4.2 Multi-objective optimization

To implement the GSEMO we view the fitness $F(x)$ described above as two objective functions $F(x) = (F_1(x), F_2(x))$, with $F_1(x) = |x|_1$ and $F_2 = u(x) + p(x)$. Note that there holds $\min_x\{F_2(x)\} = 1$. From this point of view, once a covering has been found, the algorithm tries to join the dominating set found at each iteration. In fact, GSEMO tries to add a vertex which will either dominate the maximum number of vertices or connect the maximum number of vertices already in solution. The following Lemma 4.2 holds.

LEMMA 4.2. *Let G be a connected graph,* OPT *be the optimal connected dominating set of the solution and f_i the value of $u(x)+p(x)$ after taking i-many vertices as a solution. Then there exist a vertex such that*

$$\begin{aligned} f_i &\leq f_{i-1} - \frac{f_{i-1}}{|\text{OPT}|} + 1 \\ &\leq f_0 \left(1 - \frac{1}{|\text{OPT}|}\right)^i + \sum_{j=1}^{i-1} \left(1 - \frac{1}{|\text{OPT}|}\right)^j \end{aligned}$$

A proof of this lemma is given by Guha et. al. [11]. The following theorem holds.

THEOREM 4.3. *The expected time until GSEMO has obtained* $\ln 2 + \ln\left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) + 1$*-approximation for minimum connected dominating set on PLB-U graphs with $\beta > 2$, $t \geq 0$ is $\mathcal{O}(n^3)$.*

A proof of this theorem can be found in the full version [4].

5 MAXIMUM INDEPENDENT SET

For a graph $G = (V, E)$, the *maximum independent set* (MIS) is a subset $S \subseteq V$ of maximum size, such that no two different vertices $u, v \in S$ are adjacent.

5.1 Single-objective optimization

To implement the (1+1) EA we use the fitness function used by Bäck and Khuri [2], which is defined as

$$F(x) = |x|_1 - n^2 \sum_{i=1}^{n} x_i \sum_{j=1}^{n} x_j e_{ij}$$

where e_{ij} is 1 if there is an edge (v_i, v_j). In this case, the objective of the (1+1) EA is to maximize the $F(x)$. The first part of the fitness function gives the number of vertices in the solution. It can be observe that the second part $n \sum_{i=1}^{n} x_i \sum_{j=1}^{n} x_j e_{ij} = 0$ if and only if the vertices from the first part give an independent set. First the (1+1) EA adds vertices in the solution, in order to obtain a consistent solution. Then, it will try to remove points to maximize the fitness. To perform the analysis, we show that (1+1) EA can simulate the 3-flip neighborhood algorithm for maximum independent set. Khanna et al. [13] proved that 3-flip neighborhood produces $\frac{\Delta+1}{2}$-approx solution for the independent set. We first prove that the 3-flip neighborhood is a constant-approx for the PLB-Graphs.

LEMMA 5.1. *For the PLB-U graphs with parameters $\beta > 2$, $t \geq 0$ a 3-flip neighborhood local search produces a $\left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right)$-approx independent set, where $\left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) \geq 1$.*

A proof of this lemma can be found in the full version [4]. We will use again the fitness partitioning method to prove the theorem below. By using the Lemma 5.1, we partition the objective space in three parts.

$$\begin{aligned} S_1 &= \{x \in \{0,1\}^n : \ x \text{ is infeasible and } F(x) < 0\} \\ S_2 &= \{x \in \{0,1\}^n : \ x \text{ is feasible and} \\ &\qquad 0 \geq F(x) < \left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) + 1\} \\ S_3 &= \{x \in \{0,1\}^n : \ x \text{ is feasible solution and} \\ &\qquad F(x) \geq \left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) + 1\} \end{aligned}$$

The following theorem holds.

THEOREM 5.2. *For the PLB-U graphs with parameters $\beta > 2, t \geq 0$* (1+1) EA *produces $\left(a_{\beta,t} \cdot b_{c_1,\beta,t}\right) + 1$-approx independent set in expected $\mathcal{O}(n^4)$ time.*

PROOF. Let there be ℓ many nodes in the solutions, whose deletion will make the solution from S_1 to S_2. By the same argument as in theorem 3.1, a solution x can be changed into a feasible one in $\mathcal{O}(n \log n)$ fitness evaluations.
If we assume that the current solution is in S_2, then there

exists a 3-bit flip operation. The probability that the (1+1) EA performs a 3-bit flip operation is bounded by $\frac{1}{n^3}\left(1-\frac{1}{n}\right)^{n-3} \geq \frac{1}{en^3}$. It can be observe that when the solution $x \in S_2$ then $1 \leq F(x) \leq n-1$. Thus, there are at most n $3-bit$ flips. It takes at most $\mathcal{O}(n^4)$ on expectation to perform all valid 3-bit flips. Hence it takes $\mathcal{O}(n^4)$ to bring the solution from S_2 to S_3. □

6 EXPERIMENTS

We experimentally compare the performance of the (1+1) EA and greedy algorithm on scale-free networks. We consider both artificially generated graphs and one downloaded from the Stanford Network Analysis Project (SNAP). We consider the *Minimum Vertex Cover* problem. As previously described, for a given graph $G = (V, E)$, with n nodes, a solution is stored in memory as a pseudo-boolean array, its length is the number of vertices of the graph. In all cases, the quality of the solution is evaluated against the following function

$$f(x) = n^2 u(x) + |x_i|_1 \tag{2}$$

with $u(x)$ returning the number of uncovered edges. In all cases we perform non-linear regression to infer asymptotic trend. The fitting is performed using the nonlinear least-squares Marquardt-Levenberg algorithm implemented by the "lm" command of R 3.2.2 GUI 1.66 Mavericks build (6996). We perform a t-Student Test on the each model, to determine whether it outperforms "random noise" as a predictor. We look at the corresponding p-value, and consider the model valid only for p-value < 0.05. We accept variables such that the probabilities $p_{|t|}$ of obtaining a corresponding value outside the confidence interval are $p_{|t|} < 0.1^{-10}$. Thus all variables have a very high level of significance. All tests are preformed on Ubuntu 14.04.4 LTS, and implemented as Unix command line executable.

In the first set of experiments we compare the (1+1) EA with the greedy algorithm on artificially generated scale-free graphs. For a given number of nodes n, we generate a random graph with power-law distribution exponent $\alpha = 2.5$. We then let the (1+1)-EA run for $\Theta(n \log(n))$ steps. In all cases, the (1+1) EA is able to find a covering. We consider 10^2 runs and we look at the sample mean of the solution size. We then compare it with the fitness of the solution found deterministically with the greedy algorithm. The results are displayed in Figure 2. We clearly see that, for increasing number of nodes, the (1+1) EA outperforms the greedy search. In fact, in the case of the (1+1) EA the fitting curve is $\Theta(\sqrt{n})$, while in the case of greedy the best solution has linear size $\Theta(n)$ in the number of nodes. Given the positive results for the (1+1) EA, we consider the more generic $(\mu+1)$ EA and test it with a real social network. We consider the General Relativity and Quantum Cosmology collaboration network, from the e-print arXiv (cf. Leskovec et. al. [17]). It covers scientific collaborations between authors submitted to General Relativity and Quantum Cosmology category. If an author i co-authored a paper with author j, the graph contains a undirected edge from i to j. It consists of 5×10^3 nodes and 14×10^5 edges. For a given population size μ, we let the $(\mu+1)$ EA run for $\Theta(n \log(n))$ fitness evaluations. Since a large population requires more fitness evaluations at each step, then there is a trade-off between population size and number of steps. In Figure 3 we display the fitness of the best solution found, for a given population size. We display both the size of the covering and the number of uncovered edges. We see that for increasing μ, more and more edges are left uncovered, and the covering size decreases. Thus, with fitness defined as in Equation 2 the $(\mu+1)$ EA does not yields a significant advantage in comparison with the (1+1) EA. However, a lighter weight on the $u(x)$ may give a different optimal μ.

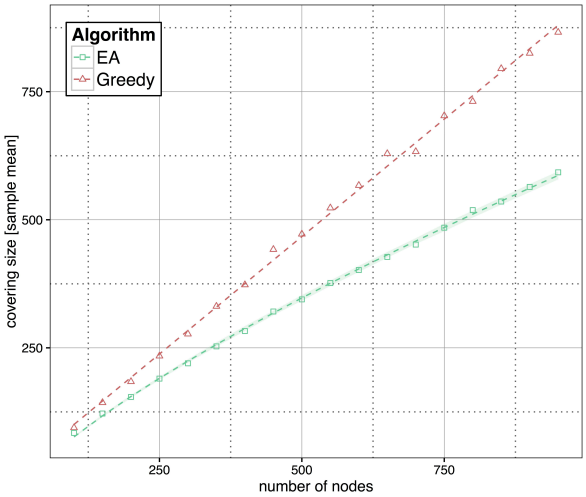

Figure 2: Run time for the (1+1) EA and greedy algorithm on artificially generated scale-free graphs, to find an approximation of the minimum vertex cover. We let the (1+1) EA run for $\Theta(n \log(n))$ steps, and take the sample mean of 10^2 runs. The greedy algorithm stops as soon as covering is generated. The fitting curves are obtained via non-linear regression, and the shading is proportional to the sample standard deviation. We see that the (1+1) EA beats the greedy algorithm on all graphs.

7 CONCLUSION

In this paper, we looked at the approximation ratio and run time analysis of both single- and multi-objective EAs, for well known NP on the graphs with deterministic PLB properties, and the power-law exponent $\beta > 2$. In sections 3 and 4 we analyze the (1+1) EA and GSEMO for the maximum dominating set, maximum vertex cover and connected dominating set problems. We show that the (1+1) EA and GSEMO obtain constant-approximation within polynomial run time. In section 5 we analyze the (1+1) EA for the maximum independent set problem and show that it obtains constant approximation ratio within expected polynomial steps.

In section 6 we observe experimentally that the (1+1) EA always produces better results than the greedy algorithm for

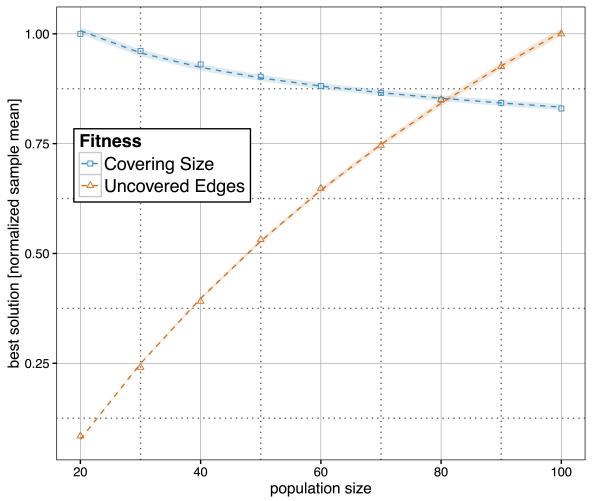

Figure 3: Run time for the $(\mu+1)$ EA on a real-world network. We consider a scale-free network from the Stanford Network Analysis Project (SNAP), which has 5×10^3 nodes and 14×10^5 edges. We let $(\mu+1)$ EA run for $\Theta(n\log(n))$ fitness evaluations. The dashed curves are obtained via non-linear regression, and the shading is proportional to the sample standard deviation. We see that the covering size decreases, and that the number of uncovered edges increases.

the minimum vertex cover problem. To this end, we give a worst case instance with the PLB properties, where greedy algorithm obtain an optimal solution, but the (1+1) EA gives worst possible solution with constant probability. We conclude that the EAs for the above-mentioned problems on the graphs with PLB properties and $\beta > 2$, obtain better approximation than the known worst-case approximation. This implies that topological properties of real-world instances play an important role in the performance of EAs. On the other hand, the worst-case example indicates that just PLB properties are not enough to always obtain a better results than the greedy algorithm. Therefore, other properties of real-world networks may affect the EAs run time.

We plan to further explore the interplay between topological properties and the run time analysis of single- and multi-objective algorithms in the future.

REFERENCES

[1] L. Adamic, O. Buyukkokten, and E. Adar. A social network caught in the web. *First Monday*, 8, 2003.
[2] T. Bäck and S. Khuri. An evolutionary heuristic for the maximum independent set problem. In *Proc. of 1st WCCI*, pages 531–535, 1994.
[3] P. Brach, M. Cygan, J. Lacki, and P. Sankowski. Algorithmic complexity of power law networks. In *Proc. of 27th SODA*, pages 1306–1325, 2016.
[4] A. Chauhan, T. Friedrich, and F. Quinzan. Approximating Optimization Problems using EAs on Scale-Free Networks. *ArXiv e-prints*, 1704.03664, 2017.
[5] A. Chauhan, T. Friedrich, and R. Rothenberger. Greed is good for deterministic scale-free networks. In *Proc. of 36th FSTTCS*, pages 33:1–33:15, 2016.
[6] V. Cohen-Addad and C. Schwiegelshohn. One Size Fits All : Effectiveness of Local Search on Structured Data. *ArXiv e-prints*, 1701.08423, 2017.
[7] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51 – 81, 2002.
[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proc. of SIGCOMM*, pages 251–262, 1999.
[9] T. Friedrich, N. Hebbinghaus, F. Neumann, J. He, and C. Witt. Approximating covering problems by randomized search heuristics using multi-objective models. In *Proc. of GECCO*, pages 797–804, 2007.
[10] O. Giel and I. Wegener. Evolutionary algorithms and the maximum matching problem. In *Proc. of 20th STACS*, 2003.
[11] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, 1998.
[12] T. Jansen and I. Wegener. Evolutionary algorithms - how to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Transactions on Evolutionary Computation*, 5:589–599, 2001.
[13] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28:164–191, 1998.
[14] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 31:1481–1493, 1999.
[15] X. Lai, Y. Zhou, J. He, and J. Zhang. Performance analysis of evolutionary algorithms for the minimum label spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 18:860–872, 2014.
[16] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proc. of 11th KDD*, pages 177–187, 2005.
[17] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1, 2007.
[18] F. Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181:1620 – 1629, 2007.
[19] F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science*, 378:32–40, 2007.
[20] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
[21] P. S. Oliveto, J. He, and X. Yao. Evolutionary algorithms and the vertex cover problem. In *Proc. of CEC*, pages 1870–1877, 2007.
[22] P. S. Oliveto, J. He, and X. Yao. Analysis of population-based evolutionary algorithms for the vertex cover problem. In *Proc. of CEC*, pages 1563–1570, 2008.
[23] X. Peng. Performance analysis of (1+1) EA on the maximum independent set problem. In *Proc. of 1st ICCCS*, pages 448–456, 2015.
[24] A. G. Phadke and J. S. Thorp. *Computer Relaying for Power Systems*. John Wiley & Sons, Ltd, 2009.
[25] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K. Ko. A greedy approximation for minimum connected dominating sets. *Theoretical Computer Science*, 329:325–330, 2004.
[26] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
[27] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.
[28] C. Witt. *Worst-Case and Average-Case Approximations by Simple Randomized Search Heuristics*, pages 44–56. 2005.