Approximating Optimization Problems using EAs on Scale-Free Networks

Ankit Chauhan* Tobias Friedrich* Francesco Quinzan*

June 15, 2017

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 the (1+1) EA takes $\Omega(n^2)$ steps to obtain a solution as good as the greedy algorithm 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.

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 [6, 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, 29, 15, 28]. However, for real-world instances, the solution quality heavily depends on the underlying

*Algorithm Engineering Group, Hasso Plattner Institute Potsdam, Germany, `firstname.lastname@hpi.de`

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 $\mathsf{P}{=}\mathsf{NP}$ [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 [7, 14, 16, 1, 24, 20, 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.[5] 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 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 $[2^i, 2^{i+1})$.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 atleast 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.[4] 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[27, 25]. 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

problems	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 withPLB properties 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 properties and power-law exponent $\beta > 2$ within expected polynomial many steps.

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

Table 2: Comparison of the approximation ratios achieved by GSEMO on networks withPLB 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.

a better solution than the greedy on the networks with PLB-(U,L,N) in asymptotically lesser time?

We theoretically analyze EAs for well-known NP-hard problems: MDS, MVC, MIS and CDS. The MVC has been studied extensively [22, 21], 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. Back et al. in [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. [4] for the approximation on

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

MDS and CDS. A summary of these results is given in Table 1 and Table 2.

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.

Algorithm 3 Local Search Algorithm

choose $x \in \{0,1\}^n$ u.a.r.
while (termination condition not satisfied) **do**
    choose y in a neighborhood 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

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.

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. [4]). Many of these concepts have been informally introduced in the introduction.

Definition 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. *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 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 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. [4].

Lemma 1 (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

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

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 the 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 1. *For the PLB graphs with parameters $\beta > 2$, $t \geq 0$ consider the quantities*

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

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

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

Proof. The run-time of $\mathcal{O}(n \log n)$ follows from the Theorem 1 of [8] and from the fact that a vertex cover is also a dominating set. Let A be the solution produced by the (1+1) EA. This solution is an $\alpha-$approximation of OPT. To give the bound on the approximation ratio α we use Lemma 1. 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 $|A| \leq n - |\mathrm{OPT}|$.

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

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 1 we get, $\alpha \leq 2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t}$ □

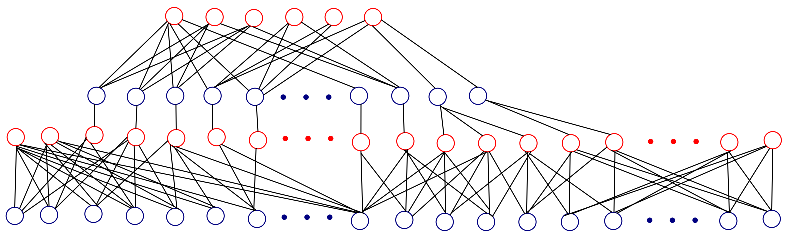

Figure 1: Worst Case graph G_{vc} for the (1+1) EA with $\Delta = \lceil n^{\frac{1}{\beta-1}} \rceil$, $\delta = 2^2$ and $|V| = N = \Theta(n)$. $|A_1|$ is the set of red vertices and A_2 is the set of blue vertices. On this graph a greedy search beats the (1+1) EA.

We now discuss two worst-case graphs with PLB-(U,L,N) properties. The first example relates to the MVC problem, and it shows that on an instance with PLB-(U,L,N) properties the (1+1) EA outputs the worst possible solution in $\Theta(n \log n)$ many fitness evaluations. In the second example, we show that on a PLB-(U,L,N), graph greedy outperforms the (1+1) EA.

Example 1: We give a construction of the graph $G_{vc} = (V, E)$(cf. Figure 1), and show that it has PLB-(U,L,N) properties. Consider $\delta = 2^p$, where δ is the smallest degree and is a constant. Also, $\Delta = \lceil n^{\frac{1}{\beta-1}} \rceil$ is a maximum degree of the graph G_{vc}. Let b_i be the PLB buckets where $b_{\log \Delta}$ is the bucket which contains the vertices with degree Δ. The procedure of constructing the G_{vc} is given as follows:

1. This steps consists of filling constant many high degree vertices. For all $j \in \{\log_2(\Delta) - p, \cdots, \log_2(\Delta)\}$ fill b_j with $\left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil$ many vertices, with p a constant such that the number of vertices in these buckets is at least $\delta - 1$. This gives the first layer V_1 of the vertices. Let $\{v_1, \cdots, v_k\}$ be the vertices created during this step, and $deg_c(v)$ the current degree of the vertex v. Also, let $deg(v)$ be the resultant degree needed for the vertex v.

2. Arrange the vertices according to their resultant degree in decreasing order. Add new vertices $\{u_1, \cdots u_{deg(v_1)}\}$ say second layer of vertices V_2 and then add an edge between each u_i and v_1. Now for each u_i add an edge to the next v_j such that degree of each u_i is exactly $\delta - 1$. If there is an vertex v_i whose degree does not satisfy the degree to be in bucket for which it was created then create another $\{u_{deg(v_1)+1} \cdots u_{deg(v_i)-deg_c(v_i)}\}$. For each iteration j repeat the above step for each v_i until all vertices $v_1, \cdots v_k$ have there resultant degree needed. If there is some vertex u_b whose degree is not $\delta - 1$ and adding an edge to some v_j is not possible than create $\delta - deg_c(u_b)$ many new vertices $v^{u_b} = \{v_1^{u_b}, \cdots, v_{\delta - deg_c(u_b)}^{u_b}\}$ and add an edge to u_b.

3. Now for each vertex u_i of degree $\delta - 1$, add a vertex v_{k+i} and add an edge (u_i, v_{k+i}). Let this is the third vertex layer V_3 which currently contain the vertices v_{k+i} for each u_i and the set v^{u_b} for each u_b in the second layer. Now for each bucket b_j where $j \in \{2, \cdots, \log_2(\Delta) - p\}$. Fill the buckets with $\left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil$ many vertices by using currently present vertices and by adding more vertices. Add the edge as described in the first step creating fourth layer of vertices V_4. Now for each v_i^4 in V_4 add an edge to the next v_j such that degree of each v_i^4 is exactly δ. If there is a some vertex with degree less than δ in V_4 than choose an vertex v_i in bucket j

such that $j \in \{4, \cdots, \log_2(\Delta) - p\}$ and add an edge to v. It is to be noted that there can be at most constant many vertices in V_4 whose degree is not δ after sequential addition. Thus addition some more edges to the vertices in above bucket will not change the number of vertices in each bucket as the degree blow up is less than half for any of the above vertices.

We use the lemma below to prove that the graph G has PLB-(U,L,N) properties (cf. Chauhan et. al. [4]). This lemma helps us to give the bound for the number of vertices in each bucket of PLB graph G.

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

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

Theorem 2. *G_{vc} is a N vertices graph have PLB-(U,L,N) properties with parameter β, $t = 0$, $c_1 = \frac{3}{(2\delta)^{\beta-1}}, c_2 = \frac{p}{\delta}$ and $c_3 = 1$ where $N = \Theta(n)$ and $0 < p < 1$.*

Proof. First we give the bound in the number of vertices N in graph G. From the graph construction algorithm we know that

$$N = |V_1| + |V_2| + |V_3| + |V_4|$$

We give the upper bound on the number of vertices in each set by using the simple integration method,

$$\begin{aligned} |V_1| &= \sum_{j=\log(\Delta)-p}^{\log\Delta} \left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil \\ &\leq \sum_{j=\log(\Delta)-p}^{\log\Delta} (n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} + 1) \\ &\in \mathcal{O}(1) \end{aligned}$$

$$\begin{aligned} |V_2| &\leq \sum_{j=\log(\Delta)-p}^{\log\Delta} \left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i * i^{-\beta} \right\rceil \\ &\leq \sum_{j=\log(\Delta)-p}^{\log\Delta} (n \sum_{i=2^j}^{2^{j+1}-1} i^{1-\beta} + 1) \\ &\leq n \sum_{j=\log(\Delta)-p}^{\log\Delta} \sum_{i=2^j}^{2^{j+1}-1} i^{1-\beta} + n^{\frac{2}{\beta-1}} \\ &\leq n \int_{n^{\frac{1}{p(\beta-1)}}}^{\Delta} x^{1-\beta} dx + n^{\frac{1}{\beta-1}} \\ &\in \Theta(n^{\frac{1}{\beta-1}}) \end{aligned}$$

$$\begin{aligned} |V_3| &\leq \sum_{j=2}^{log(\Delta)-p} \left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil + \Theta(n^{\frac{1}{\beta-1}}) \\ &\in \mathcal{O}(n + n^{\frac{1}{\beta-1}}) \\ &\in \Theta(n) \end{aligned}$$

$$
\begin{aligned}
|V_4| &\leq \sum_{j=2}^{log(\Delta)-p} \left[n \sum_{i=2^j}^{2^{j+1}-1} i * i^{-\beta} \right] + \delta * \mathcal{O}(n^{\frac{1}{\beta-1}}) \\
&\in \Theta(n)
\end{aligned}
$$

Thus, $N \in \Theta(n)$. Now we prove that the graph G_{vc} have PLB-(U,L,N) properties. PLB-(U,L): From the step 1,2 and 3 it can be observed that the number of vertices b_j with degree in $[2^j, 2^{j+1})$ where $j \in \{2, \cdots, \log_2(\Delta - 1)\}$ is

$$
\begin{aligned}
n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_j \leq n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} + 1 \\
n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_j \leq 2n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta}
\end{aligned}
\tag{2}
$$

Since $|V_3| = C * n$, where C is some constant hence $n \geq pN$ where $p \leq 1$

$$
pN \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_j \leq 2N \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta}
$$

Now for j=1

$$
\begin{aligned}
|V_4| < b_1 &\leq \sum_{j=2}^{\log_2(\Delta)} \left[n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right] + \mathcal{O}(n^{\frac{1}{\beta-1}}) \\
\frac{|V_3|}{\delta} < b_1 &\leq 3N \sum_{i=2^{\log(\delta)+1}}^{\Delta} i^{-\beta} \\
\frac{pN}{\delta} \cdot \sum_{i=\delta}^{2\delta-1} i^{-\beta} \leq b_1 &\leq \frac{3N}{\beta-1} (2\delta)^{1-\beta} \\
\frac{pN}{\delta} \sum_{i=\delta}^{2\delta-1} i^{-\beta} \leq b_1 &\leq 3N \cdot \frac{1}{(2\delta)^{\beta-1}} \sum_{i=\delta}^{2\delta-1} i^{-\beta}
\end{aligned}
\tag{3}
$$

Inequality 3 follows from lemma 2 G also have PLB-N property as the vertices of degree more than δ are adjacent to the degree δ vertices. Thus graph is a PLB-(U,L,N) with $c_1 = \frac{3}{(2\delta)^{\beta-1}}, c_2 = \frac{p}{\delta}$ and $c_3 = 1$. □

Theorem 3. *For the PLB-(U,L,N) graph G_{vc}, the* (1+1) EA *with probability $\mathcal{O}(1)$ output worst possible vertex cover in expected $\Theta(N \log N)$ time whereas takes $\Omega(N^\delta)$ to converge.*

Proof. We can divide the vertices of G_{vc} in two sets, $A_1 = \{V_1 \cup V_3\}$, $A_2 = \{V_2 \cup V_4\}$. It can be easily observable that the OPT $= A_1$ as $|A_1| < |A_2|$. Let $|A_1| = \epsilon N$ and $|A_2| = (1-\epsilon)N$ In the *first phase* we investigate the probability that the (1+1) EA takes A_2 as vertex cover with at least one vertex of $|A_1|$ missing in the solution. In the *second phase* we give the lower bound on the probability that all the vertices of the $|A_1|$ will be removed to obtain a local optimum. We conclude that the (1+1) EA reaches a configuration with constant probability, from which it escapes in expected $\Omega(n^\delta)$ many steps.

First phase: This phase runs for $\frac{4e}{(1-\epsilon)} N \ln N$ steps. We prove that the (1+1) EA outputs a vertex cover within this phase which contains A_2 with constant probability. We consider a *simple event*, by which only single bit flips of A_2 occur. The probability of the simple event is at least $\frac{1-\epsilon}{e}$, meaning that its expected waiting time is $\frac{e}{1-\epsilon}$. Now if the (1+1) EA runs for k steps, than by the Chernoff bound the probability that there are at least $\frac{k(1-\epsilon)}{2e}$ A_2 bit flips is greater than $1 - e^{-\frac{k(1-\epsilon)}{8e}}$. Since $k = \frac{4e}{(1-\epsilon)} N \ln N$, then the probability that in this phase $2N \ln N$ many simple A_2 events occur is at least $\left(1 - \frac{1}{N}\right)$. Let N' be the current number of uncovered edges in the graph. All simple A_2-steps that add a vertex of A_2 are accepted and the total distance decrease of these step is N', since choosing all vertices from A_2 is clearly a valid vertex cover. Thus the number of uncovered edges after a simple event adds the vertex to the solution is at most $N'(1 - \frac{1}{(1-\epsilon)N})$. Since $N' \leq N$, then the number of uncovered edges after $2N \ln N$ steps is $N(1 - \frac{1}{(1-\epsilon)N})^{2N \ln N} \leq \frac{1}{N}$, which is strictly less than $\frac{1}{2}$. Combining this observation with Markov's inequality, a vertex cover with A_2 is produced with probability at least $\frac{1}{2}\left(1 - \frac{1}{N}\right)$.
We now analyze the probability that the (1+1) EA does not include at least one vertex of A_1 with constant degree in the solution, after $\frac{4e}{(1-\epsilon)} N \ln N$ steps. Consider the case when A_2 is in the solution before A_1. By Chernoff bound at least $\frac{|A_1|}{3} = \frac{\epsilon N}{3}$ many vertices of A_1 are not in the initial solution with probability $1 - e^{\Omega(\epsilon N)}$. Thus the probability that the vertices from $|A_1|$ which are not in the initial solution is not chosen after $\frac{4e}{(1-\epsilon)} N \ln N$ steps is at least

$$1 - \left(1 - \left(1 - \frac{1}{N}\right)^{\frac{4e}{(1-\epsilon)} N \ln N}\right)^{\frac{\epsilon N}{3}} > 1 - e^{-\frac{\epsilon(1-\epsilon)}{12e}}$$

Combining this inequality with the probability that the (1+1) EA choose A_2 in the solution we get the total probability that the (1+1) EA chooses A_2 without at least one node from A_1 is lower bounded by $\frac{1-e^{-\frac{\epsilon(1-\epsilon)}{12e}}}{2}\left(1 - \frac{1}{N}\right)$.

Second phase: We give that with constant probability all the accepted vertices of A_1 will be removed in expected $\mathcal{O}(N \log N)$ steps. We consider the case when all nodes of A_2 and all nodes of A_1 but one is in the solution. We consider two possibilities. Either the number of vertices of A_1 in solution increases and as many vertices of A_2 decreases we call *bad event*. This is the case when all the missing node in the neighbor of $v \in A_2$ are included in the solution and at least as many vertices of A_2 are removed from the solutionj. The other possibility we consider is when the number of vertices of A_1 decrease or stay same after an mutation step, which we call *good event*. Following our terminology, if a bad event occurs, then the solution drifts away from the local optimum. On the other side, as more and more good events occur the covering approaches the solution A_2. The probability of a good event is at least $\frac{|A_1|-k}{en} \geq \frac{1}{eN}$ where k is the missing vertices of A_1 from the current solution. The probability of the bad event is upper bounded from $N^{-2+\frac{1}{\beta-1}} + N^{-1}$. We calculate the probability of a good event before a bad event, which is lower bounded by $\frac{1/eN}{2/eN + N^{-2+\frac{1}{\beta-1}}} = \frac{N^{-2+\frac{1}{\beta-1}}}{N^{-2+\frac{1}{\beta-1}} + 2e}$. Thus the probability that all the nodes of V_1 is remove from the solution is,

$$\prod_{k=1}^{\epsilon n} \left(\frac{N^{-2+\frac{1}{\beta-1}}}{N^{-2+\frac{1}{\beta-1}} + e}\right) \in \Omega(1)$$

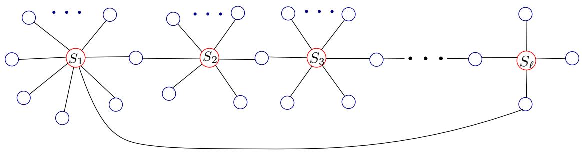

Figure 2: Worst Case graph G_W for the (1+1) EA with $\Delta = \lceil n^{\frac{1}{\beta-1}} \rceil + 1$ and $|V| = N = \Theta(n)$. V_1 is the set of red vertices and V_2 is the set of blue vertices. On this graph a greedy search beats the (1+1) EA.

Now to converge the solution towards global optima, at least δ many vertices of A_1 are needed to be added to remove at least one more than that many vertices of A_2 from the common neighborhood of added vertices, the probability of this event is at most $n^{-\delta-1+\frac{1}{\beta-1}}$. Thus the expected waiting time to converge the solution is $\Omega(n^{-\delta-1+\frac{1}{\beta-1}}) \in \Omega(n^{\delta})$, from which the thesis follows. $\square$

Example 2 We give a graph G_W as described below (cf. Figure 2).

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.

Theorem 4. *G_W 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 $p = \left(2 \cdot 4^{1-\beta}\left(\frac{4}{\beta-2} + \frac{1}{\beta-1}\right)\right)^{-1}$.*

Proof. First we give the bound in the number of vertices N in graph G, From the step 1 of graph construction algorithm,

$$\begin{aligned} N &= \sum_{j=2}^{\log_2(\Delta-1)} (2^j+1) \left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil \\ &\le n \sum_{j=2}^{\log_2(\Delta-1)} \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta}(i+1) + \sum_{j=2}^{\log_2(\Delta-1)} (2^j+1) \\ &\le 2n \int_4^{\Delta} x^{-\beta}(x+1)dx \\ &\le 2 \cdot 4^{1-\beta}\left(\frac{4}{\beta-2} + \frac{1}{\beta-1}\right) n \end{aligned} \tag{4}$$

From inequality 4, $N = \Theta(n)$

Now we prove that the graph G_W have PLB-(U,L,N) properties.

PLB-(U,L): From the step 1 and 2, it can be observed that the number of vertices b_j

with degree in $[2^j, 2^{j+1})$ where $j \in \{2, \cdots, \log_2(\Delta - 1)\}$ is

$$\begin{aligned} n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_j \leq n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} + 1 \\ n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_j \leq 2n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \end{aligned} \tag{5}$$

From inequality 4, $n \geq pN$

$$pN \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_j \leq 2N \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta}$$

Now for j=1

$$\begin{aligned} \sum_{j=2}^{\log_2(\Delta-1)} n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \leq b_1 = \sum_{j=2}^{\log_2(\Delta-1)} \left\lceil n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \right\rceil \\ pN \sum_{i=4}^{\Delta} i^{-\beta} \leq b_1 \leq 2N \sum_{i=4}^{\Delta} i^{-\beta} \\ pN \sum_{j=4}^{\Delta} j^{-\beta} \cdot \sum_{i=2}^{3} i^{-\beta} \leq b_1 \leq \frac{2N}{\beta - 1} 2^{1-\beta} \\ \frac{(4^{1-\beta} - 5^{1-\beta})}{\beta - 1} pN \sum_{i=2}^{3} i^{-\beta} \leq b_1 \leq 2N \cdot \frac{1}{1 - 2^{-\beta+1}} \sum_{i=2}^{3} i^{-\beta} \end{aligned} \tag{6}$$

Inequality 6 follows from lemma 2 and since $\sum_{i=2}^{3} i^{-\beta} < 1$. Now it remain to prove for the number of vertices of degree 1 for vertices of degree 1 we just need to prove PLB-L as $N = \Theta(n)$ hence they already follow $PLB - U$.

$$\begin{aligned} b_0 &\geq \sum_{j=2}^{\log_2(\Delta-1)} (2^j - 1) n \sum_{i=2^j}^{2^{j+1}-1} i^{-\beta} \\ &\geq 3n \frac{(4^{1-\beta} - 5^{1-\beta})}{\beta - 1} \end{aligned}$$

G also have PLB-N property as the vertices of degree more than 2 are adjacent to the degree 1 and degree 2 vertices. Thus graph is a PLB-(U,L,N) with $c_1 = \frac{2}{1-2^{-\beta+1}}, c_2 = \frac{(4^{1-\beta}-5^{1-\beta})}{\beta-1} p$ and $c_3 = 1$.

$\square$

Theorem 5. [1] *For the PLB-(U,L,N) graph G_W with $\beta > 2.5$, the* (1+1) EA *with probability $\mathcal{O}(1)$ produces worst possible dominating set in $\Theta(N \log N)$ time whereas takes $\Omega(N^2)$ time to produce a dominating set better than that.*

[1] Corrected version the theorem 3.4 from paper appeared in GECCO'17

Proof. We can divide the vertices of G_W in two sets, $V_1 = \{v \in V | \deg(v) > 3\}$, $V_2 = \{v \in V | \deg(v) \leq 2\}$. It can be easily observable that the OPT $= V_1$. In the *first phase* we investigate the probability that (1+1) EA takes V_2 as dominating set with at least one vertex of $|V_1|$ missing in the solution. In the *second phase* we give the lower bound on the probability that all the vertices of the $|V_1|$ will be removed to obtain a local optimum. We conclude that the (1+1) EA reaches a configuration with constant probability, from which it escapes in expected $\Omega(n^2)$ many steps.

First phase: This phase runs for $\frac{4e}{(1-\epsilon)}N \ln N$ steps. We prove that the (1+1) EA outputs a dominating set with in this phase which contains V_2 with constant probability. We consider a *simple event*, by which only single bit flips of V_2 occur. Probability of the simple event is at least $\frac{1-\epsilon}{e}$, meaning that its expected waiting time is $\frac{e}{1-\epsilon}$. Now if the (1+1) EA runs for k steps, than by the Chernoff bound the probability that there are at least $\frac{k(1-\epsilon)}{2e}$ V_2 bit flips is greater than $1 - e^{-\frac{k(1-\epsilon)}{8e}}$. Since $k = \frac{4e}{(1-\epsilon)}N \ln N$, then the probability that in this phase $2N \ln N$ many simple events occur is at least $\left(1 - \frac{1}{N}\right)$.
Let N' be the current number of non-dominated vertices in the graph. All simple V_2-steps that add a vertex of V_2 are accepted and the total distance decrease of these step is N', since choosing all vertices from V_2 is clearly a valid dominating set. Thus the number of non dominated vertices after a simple event adds the vertex to the solution is at most $N'(1 - \frac{1}{(1-\epsilon)N})$. Since $N' \leq N$, then the number of non dominated vertices after $2N \ln N$ steps is $N(1 - \frac{1}{(1-\epsilon)N})^{2N \ln N} \leq \frac{1}{N}$, which is strictly less than $\frac{1}{2}$. Combining this observation with Markov's inequality, a dominating set with V_2 is produced with probability at least $\frac{1}{2}\left(1 - \frac{1}{N}\right)$.
We now analyze the probability that the (1+1) EA does not include constant many nodes of V_1 with constant degree in the solution, after $\frac{4e}{(1-\epsilon)}N \ln N$ steps. Consider the case when V_2 is in the solution before V_1. By Chernoff bound at least $\frac{|V_1|}{3} = \frac{\epsilon N}{3}$ many vertices of V_1 are not in the initial solution with probability $1 - e^{\Omega(\epsilon N)}$. Thus the probability that the vertices from $|V_1|$ which are not in the initial solution is not chosen after $\frac{4e}{(1-\epsilon)}N \ln N$ steps is at least

$$1 - \left(1 - \left(1 - \frac{1}{N}\right)^{\frac{4e}{(1-\epsilon)}N \ln N}\right)^{\frac{\epsilon N}{3}} > 1 - e^{-\frac{\epsilon(1-\epsilon)}{12e}}$$

Combining this inequality with the probability that (1+1) EA choose V_2 in the solution we get the total probability that the (1+1) EA chooses V_2 without at least one node from V_1 is lower bounded by $\frac{1-e^{-\frac{\epsilon(1-\epsilon)}{12e}}}{2}\left(1 - \frac{1}{N}\right) \cdot \Theta(1)$.

Second phase: We give that with constant probability all the accepted vertices of V_1 will be removed in expected $\mathcal{O}(N \log N)$ steps. We consider the case when all nodes of V_2 and all nodes of V_1 but one is in the solution. We consider two possibilities. A missing nodes of V_1 is chosen and added to the solution removing at least one more V_2 vertices, which we call *bad event.* The other possibility we consider is when the number of vertices of V_1 decrease or stay same after an mutation step, which we call *good event.* Following our terminology, if a bad event occurs, then the solution drifts away from the local optimum. On the other side, as more and more good events occur the covering approaches the solution V_2. The probability of a good event is at least $\frac{|V_1|-k}{en} \geq \frac{1}{eN}$ where k is the missing vertices of V_1 from the current solution. We claim that probability of the bad event is upper bounded from $N^{-2} + N^{-1}$. We calculate the probability of a good event before a bad event, which is lower bounded by $\frac{1/eN}{2/eN+N^{-2}} = \frac{N}{2N+e}$. Thus the probability that all the nodes of V_1 is remove from the solution is,

$$\prod_{k=1}^{\epsilon N}\Big(\frac{N}{N+2e}\Big) \geq e^{-2e} \in \Omega(1)$$

Thus the expected waiting time for removing V_1 vertices is $\mathcal{O}(N \log N)$. We give an upper bound on the probability of a *bad event* as follows.

Claim 6. *The probability of the bad event when at least one more vertex of the A_2 are removed from the solution then the number of vertices of A_1 are included in the solution is at most*

$$\Theta(N^{-2})\Big(1 + \Theta(N^{\frac{1}{(\beta-1)}-1} + \frac{\mathcal{O}(1)}{(\ln N)^2} N^{\frac{3}{2(\beta-1)}-1}\Big) \leq \Theta(N^{-2}).$$

Proof of claim 6. For the bad event to be happening, the accepted moves are when at least one bit corresponding to the vertex v in V_1 is set to 1 and at least two bit corresponding to the neighbor vertices of v is set to 0. Thus if k vertices of V_1 are accepted in solution than at least $k+1$ vertices of V_2 are needed to be removed from the solution. First we consider the case with 3 and 4 accepted bit flips in the solution. Since there are ϵN many V_1 vertices, then the total number of accepted three bits flips are $\sum_{v \in V_1} \binom{deg(v)}{2}$. It follows that $\sum_{v \in V_1} \binom{deg(v)}{2} \leq N \sum_{i=4}^{\Delta} i^{2-\beta} \in \Theta(N)$. This gives an upper bound of $\Theta(N^{-2})$ on the favorable 3-bit flips. Similarly, we can give the upper bound of $\Theta(N^{\frac{1}{\beta-1}-3})$ for the favorable 4-bits flips. We now analyze the case of at least 5 favorable bit-flips occur. This is the case when more than one vertices of V_1 are accepted in the solution. Let S be the set of accepted steps in the (1+1) EA, where at least $i \geq 2$ vertices of V_1 is accepted in the solution. We denote with $E(i,j,\ell)$ the event s.t. one vertex $v \in V_1$ is accepted in the solution and j many adjacent vertices are removed from the solution. The corresponding probability is

$$Pr[E(i,j,l)] = \binom{\epsilon N}{i} \frac{1}{N^i} \sum_{j=2}^{deg(v)} \binom{deg(v)}{j} \prod_{u \in S \setminus v} \left(\sum_{\ell_u=1}^{deg(u)} \binom{deg(u)}{\ell_u} \right)$$

where l_u is the number of vertices of V_2 is removed from the solution which are adjacent to the vertex $u \in V_2 \setminus v$. Using the integral approximation of the sum there holds

$$
\begin{aligned}
Pr[E(i,j,l)] &= \binom{\epsilon N}{i}\frac{1}{N^i}\sum_{j=2}^{deg(v)}\binom{deg(v)}{j}\frac{1}{N^j}\prod_{u\in S\setminus v}\left(\sum_{\ell_u=1}^{deg(u)}\binom{deg(u)}{\ell_u}\frac{1}{N^{\ell_u}}\right)\\
&\leq \epsilon^i\sum_{j=2}^{deg(v)} deg(v)^j\frac{1}{N^j}\prod_{u\in S\setminus v}\left(\sum_{\ell_u=1}^{deg(u)} deg(u)^{\ell_u}\frac{1}{N^{\ell_u}}\right)\\
&\leq \epsilon^i\int_2^{cN^{\frac{1}{\beta-1}}}\left(cN^{\frac{x}{\beta-1}-x}\right)dx\prod_{u\in S\setminus v}\int_1^{deg(u)}\left(\frac{deg(u)}{N}\right)^y dy\\
&\leq \epsilon^i\frac{(\beta-1)\,N^{\frac{2}{\beta-1}-2}}{(\beta-2)\ln(N)}\prod_{u\in S\setminus v}\frac{(b-1)\,N^{\frac{1}{2b-2}-1}}{(b-2)\ln(N)} && (7)\\
&\leq \epsilon^i\frac{(\beta-1)\,N^{\frac{2}{\beta-1}-2}}{(\beta-2)\ln N}\Big(\frac{(b-1)\,N^{\frac{\ell}{2b-2}-\ell}}{(b-2)\ln N}\Big)\\
&\leq \frac{\mathcal{O}(1)}{(\ln N)^2}N^{\frac{\ell+2}{2(\beta-1)}-(\ell+2)}\\
&\leq \frac{\mathcal{O}(1)}{(\ln N)^2}N^{\frac{3}{2(\beta-1)}-3}. && (8)
\end{aligned}
$$

Inequality 7 is valid because there is only one vertex with maximum degree $n^{\frac{1}{\beta-1}}+1 \leq cN^{\frac{1}{\beta-1}}$, and because the second highest degree of graph G_W is $n^{\frac{1}{2(\beta-1)}} \leq cN^{\frac{1}{2(\beta-1)}}$. Inequality 8 is valid because there holds $\ell \geq 1$, and because the function $N^{\frac{\ell+2}{2(\beta-1)}-(\ell+2)}$ is a decreasing function with the increasing value of ℓ. We can estimate the probability of a bad event occurring as

$$
\begin{aligned}
&Pr[3-\mathit{accepted\ bits}] + Pr[4-\mathit{accepted\ bits}] + Pr[E(i,j,l)]\\
&\qquad\leq \Theta(N^{-2}) + \Theta(N^{\frac{1}{\beta-1}-3}) + \frac{\mathcal{O}(1)}{(\ln N)^2}N^{\frac{3}{2(\beta-1)}-3}\\
&\leq \Theta(N^{-2})\left(1+\Theta(N^{\frac{1}{(\beta-1)}-1}+\frac{\mathcal{O}(1)}{(\ln N)^2}N^{\frac{3}{2(\beta-1)}-1}\right).
\end{aligned}
$$

□

From the claim above, ti follows that the probability of the (1+1) EA hitting the locally optimal solution V_2 is at least $\frac{1-e^{-\frac{\epsilon(1-\epsilon)}{12e}}}{2e^e}\left(1-\frac{1}{N}\right)\cdot\Theta(1)\in\Omega(1)$. Now if the current solution has to be updated in a way s.t. the (1+1) EA adds all the nodes of V_1 removing at least one more than as many as nodes of V_2 from the solution, than the probability of this event is at most $\Theta(N^{-2})\Big(1+\Theta(N^{\frac{1}{(\beta-1)}-1}+\frac{\mathcal{O}(1)}{(\ln N)^2}N^{\frac{3}{2(\beta-1)}-1}\Big)$. Thus the expected waiting time such that the (1+1) EA gives produces solution S where $|S| < N - |\mathrm{OPT}|$) produced is

$$\Omega(1)N^2 = N^{\Omega(2)},$$

. □

The corollary below follows from the theorem 1.

Corollary 1. *For the PLB graphs with parameters $\beta > 2$, $t \geq 0$ 1+1-EA produces $(2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t})$-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. *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* $n_k \leq n\left(1 - \frac{1}{|\text{OPT}|}\right)^k$.

Proof. We prove this by induction for base case take, $i = 1$ and since the number of non dominated is n than there exist a $v \in V$ such that it dominates atleast $\frac{n_i}{|\text{OPT}|}$ many vertices in the graph otherwise OPT is not the optimal solution. Then the number of non dominated vertices are,

$$n_1 \leq n - \frac{n}{|\text{OPT}|} \leq n \cdot \left(1 - \frac{1}{|\text{OPT}|}\right)$$

Let for n_i the statement is true i.e, $n_i \leq n\left(1 - \frac{1}{|\text{OPT}|}\right)^i$

Now again there will be at least one vertex $v_{i+1} \in V$ such that it dominate atleast $\frac{n_i}{|\text{OPT}|}$ otherwise optimum have more than $|\text{OPT}|$ vertices. Therefore the number of vertices to be dominated after v_{i+1} in solution is

$$\begin{aligned} n_{i+1} &\leq n_i - \frac{n_i}{|\text{OPT}|} \\ &\leq n_i\left(1 - \frac{1}{|\text{OPT}|}\right) \\ &\leq n\left(1 - \frac{1}{|\text{OPT}|}\right)^{i+1} \end{aligned}$$

□

Theorem 7. *The expected time until GSEMO has obtained* $\ln 2 + \ln(a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1$*-approximation for dominating set on PLB-U graphs with $\beta > 2$, $t \geq 0$ is $\mathcal{O}(n^3)$.*

Proof. We first analyze the expected time until GSEMO has produced the solution 0^n for the first time. This solution is a Pareto optimal and therefore will stay in the solution once added to the population. Also, the population size can be at most $n + 1$ as it will contain at most one solution with exactly i 1-bits. The solution 0^n is the feasible solution hence no infeasible will be added after that. To analyze this, we consider in each iteration the individual P with the minimum number of 1-bits, let $b = |P|_1$ be the number of $1-$bits. Probability to choose P is atleast $\frac{1}{n+1}$ Since the b can not increase during the run hence the probability of producing a solution P' where $|P'|_1 = b - 1$ is atleast $\frac{1}{n+1}\frac{b}{en}$. Therefore the expected waiting time to include the solution 0^n is upper bounded by $\sum_{b=1}^{n}\left(\frac{b}{en^2}\right)^{-1} = \mathcal{O}(n^2 \log n)$.

Now that 0^n bit string with fitness vector included in the population. By Lemma 3 there exist a vertex such that adding that vertex to the 0^n will reduce number of non dominated vertices to at most $n \cdot \left(1 - \frac{1}{|\mathrm{OPT}|}\right)$. Since the probability of this vertex is at least $n_1 = \frac{1}{en}$ and probability of choosing the individual 0^n is at least $\frac{1}{(n+1)}$. Hence, in expected $\mathcal{O}(n^2)$ time a individual with fitness value $(n_1, 1)$ can be added to the solution.

If $n_1 \geq 1$, then there still exist a vertex which will make $u(x) \leq n\left(1 - \frac{1}{|\mathrm{OPT}|}\right)^2$. Thus an individual $(n_2, 2)$ can be added to the solution in expected $\mathcal{O}(n^2)$ after adding $(n_1, 1)$. Similarly individual (n_k, k) can be added to the population after $(n_{k-1}, k-1)$ is included in population in expected $\mathcal{O}(n^2)$.

Since we know that $n_i \leq n \cdot \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i$ Now we give bound on the value of k, Since to be dominating set $n_i \leq n_{i-1}$ and $n_k = 0$, we have $n_{k-|\mathrm{OPT}| \geq |\mathrm{OPT}|}$. Set $i = k - |\mathrm{OPT}|$ and $n \leq \sum_{x \in \mathrm{OPT}} \deg(x) + 1$

$$\sum_{x \in \mathrm{OPT}} (\deg(x) + 1) \cdot \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i \geq |\mathrm{OPT}|$$

$$\frac{\sum_{x \in \mathrm{OPT}} (\deg(x) + 1)}{|\mathrm{OPT}|} \cdot \frac{1}{e^{i/|\mathrm{OPT}|}} \geq 1 \tag{9}$$

(10)

Now again by taking $S = \mathrm{OPT}$, $h(\deg(x)) = \deg(x) + 1$, $g(x) = h(x) - 1$ and $M = \frac{n}{2}$ as in theorem 1. we can rewrite equation 9,

$$2 \cdot (a_{\beta,t} \cdot b_{c_1,\beta,t}) \cdot \frac{1}{e^{i/|\mathrm{OPT}|}} \geq 1$$

since $k = i + |\mathrm{OPT}|$

$$(\ln 2 + \ln (a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1)|\mathrm{OPT}| \geq k$$

So, a dominating set with k many vertices will be included in population in $\mathcal{O}(n^2 k) = \mathcal{O}(n^3)$ after $(n, 0)$ included in the population. □

Corollary 2. *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 7 and the fact that the greedy output V_1 as solution of the graph G which is also a optimal solution. □

We can restate the result for the general graphs also which is the consequence of Lemma 3 and can be proven by same argument,

Corollary 3. *The expected time until GSEMO has obtained* $\ln \frac{n}{\mathrm{OPT}} + 1$*-approximation for minimum dominating set on PLB-U graphs with* $\beta > 2$*,* $t \geq 0$ *is* $\mathcal{O}(n^3)$*.*

Corollary 4. *The expected time until GSEMO has obtained* $\ln 2 + \ln(a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1$*-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 generalized 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 sub-graph 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 8. (1+1) EA *obtains* $(2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t})$*-approximation connected dominating set in expected* $\mathcal{O}(n \log n)$ *run time.*

The argument behind this theorem is identical to the one given for Theorem 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 holds.

Lemma 4. *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

$$f_i \le f_{i-1} - \frac{f_{i-1}}{|\mathrm{OPT}|} + 1$$
$$\le f_0 \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i + \sum_{j=1}^{i-1} \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^j$$

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

Theorem 9. *The expected time until GSEMO has obtained* $\ln 2 + \ln(a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1$*-approximation for minimum connected dominating set on PLB-U graphs with* $\beta > 2$*,* $t \ge 0$ *is* $\mathcal{O}(n^3)$*.*

Proof. This theorem can be proved by using the same argument of the theorem 7. Since there are also at most $n+1$ individuals in the population hence the expected time to include the $(n, 0)$ is $\mathcal{O}(n^2 \log n)$. Now if there are k many vertices are sufficient for the connected dominating set then the expected time for GSEMO is $\mathcal{O}(kn^2)$. We just need to give bound on k, From Lemma 4 we know that for every i there exist a vertex such that,

$$f_i \le f_0 \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i + \sum_{j=1}^{i-1} \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^j$$
$$\le f_0 \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i + |\mathrm{OPT}| \cdot \left(1 - \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i\right)$$
$$\le (f_0 - |\mathrm{OPT}|) \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i + |\mathrm{OPT}|$$

Since $f_i = f_{i-1} - 1$ and $f_k = 1$ hence $f_{k-2|\mathrm{OPT}|} \ge 2|\mathrm{OPT}| + 1$, set $i = k - |\mathrm{OPT}|$

$$(n - |\mathrm{OPT}|) \left(1 - \frac{1}{|\mathrm{OPT}|}\right)^i + |\mathrm{OPT}| \ge 2\mathrm{OPT}$$
$$(n - |\mathrm{OPT}|) \cdot \frac{1}{e^{i/|\mathrm{OPT}|}} \ge \mathrm{OPT}$$
$$|\mathrm{OPT}| \ln \left(\frac{n - |\mathrm{OPT}|}{|\mathrm{OPT}|}\right) \ge i$$
$$|\mathrm{OPT}| \ln \left(\frac{\sum_{x \in \mathrm{OPT}} (\deg(x) + 1)}{|\mathrm{OPT}|} - 1\right) \ge i$$
$$|\mathrm{OPT}| \ln(a_{\beta,t} \cdot b_{c_1,\beta,t} - 1) + \ln 2 \ge i$$

As $k = i + 2|\mathrm{OPT}|$,

$$k \le |\mathrm{OPT}|(\ln(a_{\beta,t} \cdot b_{c_1,\beta,t} - 1) + \ln 2 + 2)$$

Thus GSEMO will produce $|\mathrm{OPT}|(\ln(a_{\beta,t} \cdot b_{c_1,\beta,t} - 1) + \ln 2 + 2)$-approximation connected dominating set in expected $\mathcal{O}(n^3)$ time. □

We can restate the result for the general graphs also which is the consequence of Lemma 4 and can be proven by same argument,

Corollary 5. *The expected time until GSEMO has obtained* $\ln \frac{n}{|\mathrm{OPT}|} + 3$*-approximation for minimum connected dominating set on PLB-U graphs with* $\beta > 2$*,* $t \ge 0$ *is* $\mathcal{O}(n^3 \ln n)$*.*

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 Back et al. [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 $PLBU$-Graphs.

Lemma 5. *For the PLB-U graphs with parameters $\beta > 2$, $t \geq 0$ a 3-flip neighborhood local search produces a $(a_{\beta,t} \cdot b_{c_1,\beta,t})$-approx independent set, where $(a_{\beta,t} \cdot b_{c_1,\beta,t}) \geq 1$.*

Proof. We prove this lemma by using the same argument used by Khanna et al. [13]. Let S is the solution produced by 3-flip neighborhood. Let OPT be the optimal solution and $I = S \cap \mathrm{OPT}$. Since we can not add a vertex outside S to get the bigger independent set it means that every vertex in S have atleast one incoming edge from $V \backslash S$. Also, since we can not two vertices outside S to get bigger independent set it means that at most $|S \backslash I|$ vertices in $|\mathrm{OPT} \backslash I|$ can have exactly one vertices coming into S. Thus, $|\mathrm{OPT}| - |I|$ vertices in $\mathrm{OPT} \backslash I$ must have at least two edges incoming to S.

This implies that the minimum number of edges between between S and $\mathrm{OPT} \backslash I$ is $|S| - |I| + 2(|\mathrm{OPT}| - |S|)$. Also the maximum number edges between S and $\mathrm{OPT} \backslash X$ is bounded by the $vol(S \backslash I)$. Hence,

$$|S| - |I| + 2(|\mathrm{OPT}| - |S|) \leq vol(S \backslash I) \leq vol(S) \tag{11}$$

Now the equation 11 can be rewritten as,

$$\frac{|S| - |I| + 2(|| - |S|)}{|S|} \leq \frac{vol(S)}{|S|} \tag{12}$$

Now again using Lemma 1 $h(\deg(x)) = \deg(x) + 1$, $M = \frac{n}{2}$, $g(x) = \deg(x)$ we can rewrite the equation 12 as,

$$\frac{|S| - |I| + 2(|\mathrm{OPT}| - |S|)}{|S|} \leq \frac{vol(S)}{|S|}$$

$$1 - \frac{|I|}{|S|} + 2\frac{|\mathrm{OPT}|}{|S|} - 1 \leq (2 \cdot a_{\beta,t} \cdot b_{c_1,\beta,t})$$

$$(a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1 \geq \frac{|\mathrm{OPT}|}{|S|}$$

Hence the 3-flip neighborhood algorithm gives $(a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1$-approximation solution. □

By using the Lemma 5, we partition the objective space in three parts.

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

The following theorem holds.

Theorem 10. *For the PLB-U graphs with parameters $\beta > 2, t \geq 0$ (1+1) EA produces $(a_{\beta,t} \cdot b_{c_1,\beta,t}) + 1$-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 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 real-world instances 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

$$f(x) = n^2 u(x) + |x_i|_1 \tag{13}$$

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. 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. We consider both the Chung-Lu and the Hyperbolic model. For a given number of nodes n, we generate a random graph with power-law distribution exponent $\beta = 2.5$. 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

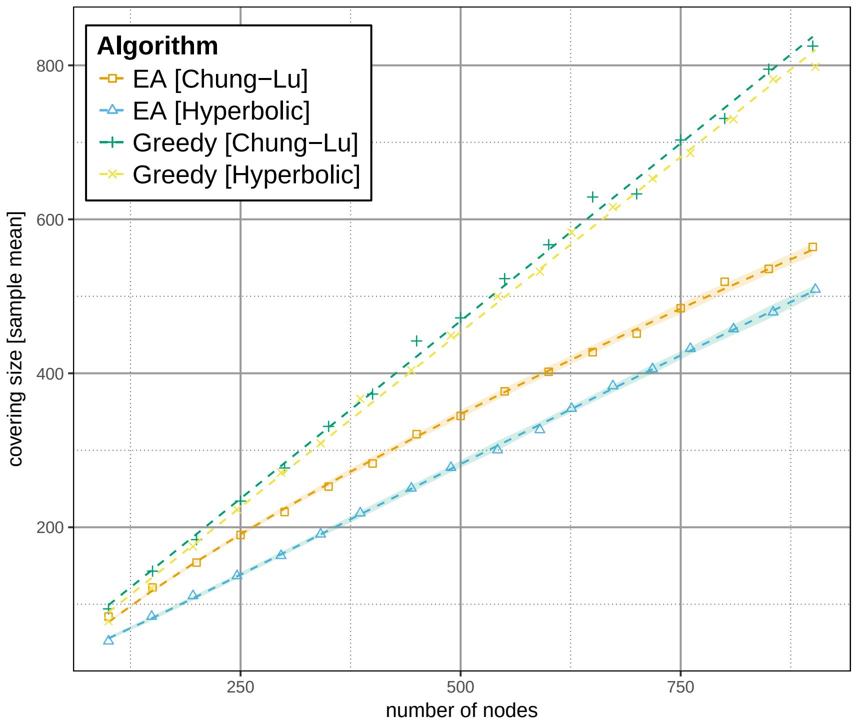

Figure 3: Run time for the (1+1) EA and greedy algorithm on artificially generated scale-free graphs with power law exponent $\beta = 2.5$, to find an approximation of the minimum vertex cover. We let the (1+1) EA run for $\Theta(n \log(n))$ steps, and take the sample mean of 10^2 runs. The greedy algorithm 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, both for the Hyperbolic model and the Chung˙Lu model.

deterministically with the greedy algorithm. The results are displayed in Figure 3. 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 $\mathcal{O}(\sqrt{n})$, 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 5 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 13 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. We conclude by experimentally showing that on the worst-case graph outlined in Section 3, the (1+1) EA gives worst performance

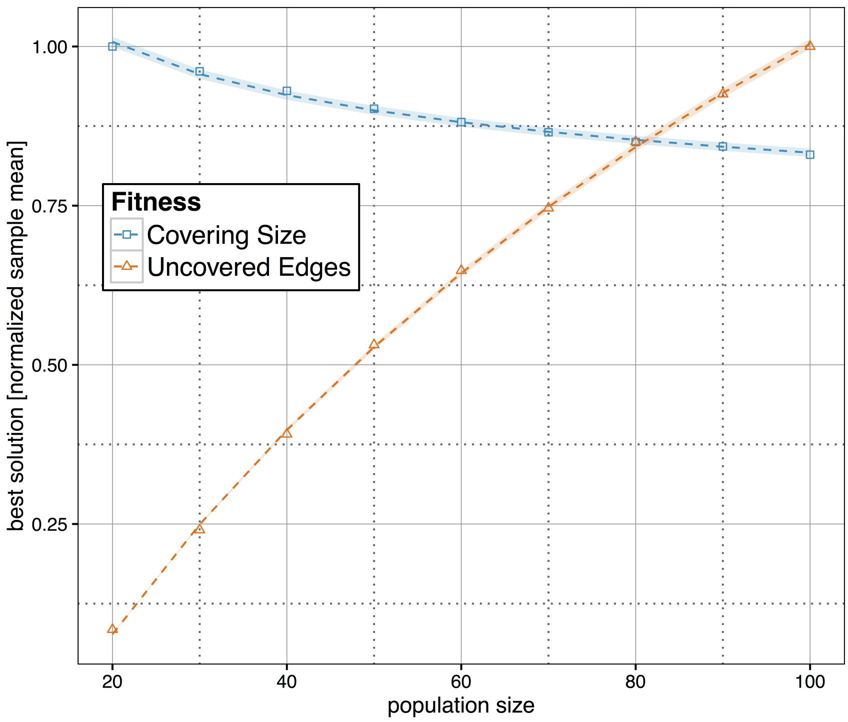

Figure 4: Run time for the $(\mu + 1)$ EA on a real-world network. We consider a scale-free network from the Stanford Network Analysis Project (SNAP), which has 5×10^3 nodes and 14×10^5 edges. We let $(\mu + 1)$ EA run for $\Theta(n \log(n))$ fitness evaluations. The dashed curves are obtained via non-linear 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.

than Greedy. Note that on any such worst-case graph the minimum vertex cover problem as outlined above is equivalent th the minimum dominating set problem. As in the case of the first experiment, for a given number of nodes n, we generate a graph with power-law distribution exponent $\beta = 2.5$. We then let the (1+1) EA run until the optimal solution is generated. We consider 10^2 runs and we look at the sample mean of the solution size. In Figure **??** we display the fitness of the best solution found, for a given number of nodes. The curve outlined for greedy follows the bound given in Theorem 5. We see that the experimental results follows the theoretical predictions: For increasing problem size greedy outperforms the (1+1) EA in terms of number of steps needed to obtain the optimal solution.

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 NPon 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.

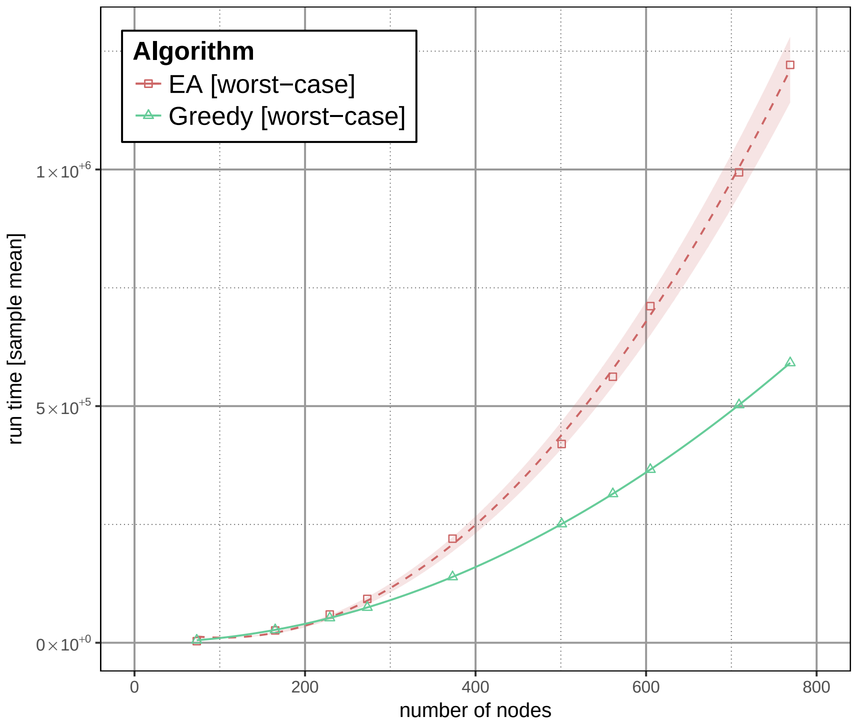

Figure 5: Run time for the (1+1) EA and on the worst-case example with power law exponent $\beta = 2.5$, to find an approximation of the minimum vertex cover (cf. Section 3). We let the (1+1) EA run until the global optimum is found, and take the sample mean of 10^2 runs. The run time of greedy is given by the upper bound predicted theoretically (cf. Theorem 5). The fitting curve is obtained via non-linear regression, and the shading is proportional to the sample standard deviation. We see that greedy beats the (1+1) EAon these graphs.

In section 6 we observe experimentally that the (1+1) EA always produces better results than the greedy algorithm for the minimum vertex cover problem. We show that the (1+1) EA gives better approximation ratio than greedy on the Chung-Lu and Hyperbolic model. We observe that on the Hyperbolic model the (1+1) EA reaches better approximation ratio than in the Chung-Lu case. 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. Łacki, and P. Sankowski. Algorithmic complexity of power law networks. In *Proc. of 27th SODA*, pages 1306–1325, 2016.

[4] 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.

[5] V. Cohen-Addad and C. Schwiegelshohn. One Size Fits All : Effectiveness of Local Search on Structured Data. *ArXiv e-prints*, 1701.08423, 2017.

[6] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51 – 81, 2002.

[7] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proc. of SIGCOMM*, pages 251–262, 1999.

[8] N. Fountoulakis and A. Huber. Quasirandom rumour spreading on the complete graph is as fast as randomized rumour spreading. *SIAM J. Disc. Math.*, 23:1964–1991, 2009.

[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.

[29] Y. Yu, X. Yao, and Z.-H. Zhou. On the approximation ability of evolutionary optimization with application to minimum set cover. *Artificial Intelligence*, 180:20 – 33, 2012.