# Absolute $o(\log m)$ Error in Approximating Random Set Covering: An Average Case Analysis

Orestis Telelis Vassilis Zissimopoulos
{telelis, vassilis}@di.uoa.gr
Phone: +30 210 727 5100, Fax: +30 210 727 5114

Division of Theoretical Informatics
Department of Informatics and Telecommunications
University of Athens
Panepistimiopolis Ilissia, Athens 15784 , Greece

**Abstract**

This work concerns average case analysis of simple solutions for random set covering ($SC$) instances. Simple solutions are constructed via an $O(nm)$ algorithm. At first an analytical upper bound on the expected solution size is provided. The bound in combination with previous results yields an absolute asymptotic approximation result of $o(\log m)$ order. An upper bound on the variance of simple solution values is calculated. Sensitivity analysis performed on simple solutions for random $SC$ instances shows that they are highly robust, in the sense of maintaining their feasibility against augmentation of the input data with additional random constraints.

**Keywords:** random set covering, algorithms, statistical analysis

## 1 Introduction

Given a *ground set* $X$, $|X| = m$, and a family $\mathcal{F}$ of subsets of $X$, $|\mathcal{F}| = n$, the uniform cost *Set Covering Problem* ($SC$) involves finding a minimum cardinality set $S \subseteq \mathcal{F}$, such that: $\bigcup_{A \in S} A = X$. The $SC$ is well known to be NP-hard. The problem is not approximable within a constant factor. Recently Feige [2] has shown that, $(1 - o(1)) \log m$ is the lowest achievable approximation ratio, unless NP has slightly superpolynomial time algorithms. Many approximation algorithms have appeared [8] for the $SC$ and several special cases of it.

Perhaps the most widely celebrated algorithm for solving the $SC$ is the $O(\log m)$ approximation *greedy algorithm*, of $O(nm^2)$ complexity, independently studied by Lóvasz [6] and Johnson [5]. The greedy algorithm's performance has proved to be satisfactory in comparison to several other well elaborated heuristics [7]. The greedy algorithm achieves an $O(\log m)$ approximation for both, uniform cost and weighted $SC$ problems [1]. The tightest known so far analysis of the greedy algorithm was developed in [9].

Random instances of $SC$ have largely been used throughout the literature as testbed for experimentations and evaluation of novel heuristics. Previous theoretical work on random instances was carried out in [11], where several asymptotic results of strong and weak convergence were presented, in a spirit of exposing the advantages incurred by the distribution of the input data.

In this work we present an algorithm for random $SC$, and study the expected solution values and their variance, thus concluding the analysis of [11]. Our analysis contributes to the under-

standing of the random $SC$ hardness and the advantages incurred by the input data distribution, through experimentally testable theoretical results. Elaboration of previous results from [11] leads to approximability properties. We also apply sensitivity analysis of simple coverings produced by the described algorithm, with respect to input data perturbation via augmentation with additional random constraints.

Section 2 comments on the random instance model and discusses a simple $O(nm)$ algorithm for the random $SC$. Analysis is developed in section 3. Some experiments illustrating our theoretical results are presented in section 4, and we finally conclude in section 5.

## 2 The random model and the simple algorithm

A random $SC$ instance will be from now on denoted with $(m, p)$, where $m$ is the cardinality of the ground set $X = \{x_1, x_2, \ldots, x_m\}$, $|X| = m$. The quantity $p$ suggests probability measure in $(0, 1)$. A family $\mathcal{F}$ of $n$ subsets of $X$ is assumed, $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$, $|\mathcal{F}| = n$, $A_j \subseteq X$. The parameter $n$ does not appear in the instance prescription, because it is determined upon $m$ by certain conditions explained below, that ensure non-triviality of an instance. The construction of a random $SC$ instance is done by letting a ground element $x_i$ belong to a subset $A_j$ with probability $p$ (that is through a Bernoulli trial):

$$Pr[x_i \in A_j] = p,\ x_i \in X,\ A_j \in \mathcal{F}$$

Thus the instance is constructed by performing $mn$ such independent *Bernoulli Trials*. In deriving asymptotic results on the instance size, a discrimination between two models is made in [12] concerning the generation of random instances of increasing size: the *independent* model, where an entirely new instance is generated, and the *incremental* model where each instance of greater dimensions is generated by an extension of an instance of smaller dimensions. In the sequel we consider only the independent model.

At first a condition is needed, that establishes feasibility of random instances. Indeed, not all random instances are feasible. The following theorem fortunately holds for all the *interesting* instances, occurring in real world applications:

**Theorem 1** *([11] Theorem 2.2) If the following condition is satisfied, then the corresponding random* SC *instances are feasible almost everywhere (with probability 1):*

$$\lim_{m \to \infty} \frac{n}{\log m} = \infty$$

In [11] a second condition is assumed to hold, namely that there exists $\alpha > 0$, such that $n \leqslant m^{\alpha}$: the number of subsets is polynomialy bounded by the number of ground elements. This condition holds in most real world $SC$ instances, and ensures that the instance is non trivial with high probability: e.g. an exponential in $m$ number of random subsets created through independent Bernoulli trials, subsumes a high probability of existence of a subset $A$, $A = X$, which makes the instance trivial. We will assume that these conditions are satisfied, and in the sequel we will consider $m$ to be the leading parameter of our analysis, assuming that $n$ behaves appropriately.

The simple algorithm is described as algorithm 1. The parameter $k_0$ depends on $p$ and $m = |X|$, and its value will be determined later. The algorithm picks $k_0$ arbitrary subsets from $\mathcal{F}$ and updates $X$ with respect to their union. Any ground element not covered by these $k_0$ subsets, is subsequently covered by an appropriately selected subset from $\mathcal{F}$. The algorithm's complexity is $O(nm)$ due to the $O(nm)$ $X$ update by the union of the first $k_0$ subsets.

**Algorithm 1** The simple algorithm

```
simple(X, F, p)
1.   S ← ∅
2.   k_0 ← k_0(|X|, p)

3.   store in S k_0 arbitrary distinct subsets from F
4.   F ← F − S
5.   X ← X − ⋃_{A∈S} A

6.   while X ≠ ∅
7.       pick x ∈ X and A ∈ F : x ∈ A
8.       S ← S ∪ A
9.       F ← F − {A}
10.      X ← X − {x}
11.  return S
```

The simple algorithm is obviously deterministic. In what follows, we apply an average case analysis of the algorithm's behavior on the class of random $SC$ instances. We shall prove that, on average, the algorithm provides solutions with $o(\log m)$ absolute deviation from the optimum, exhibiting $O(1)$ variance. Furthermore, simple solutions remain feasible under augmentation of the instance with $\omega(m)$ random constraints.

# 3 Analysis

In order to derive an upper bound on the expected size of the solution produced by the simple algorithm, some definitions are introduced. Assuming that $k_0 \leqslant m$, the simple algorithm picks at most $m$ subsets from $\mathcal{F}$. We divide the progress of the algorithm in *steps*. At each step a subset is selected. Each of the first $k_0$ steps corresponds to selection of an arbitrary subset, whereas each of the subsequent steps (within the *while* loop) corresponds to selection of a subset covering a specific ground element. Let $k \in \{1, \ldots, m\}$ denote the $k$-th selected subset (on the $k$-th step). Although the algorithm may select less than $m$ subsets, $k$ is extended to the range $1 \ldots m$, assuming that if a feasible solution has been constructed before the $k$-th step, then no subset is selected during this step and the subsequent ones. Some definitions follow:

- $\{S_m\}$ is a sequence of random variables, $S_m$ being the simple solution size, i.e. the number of subsets used by the simple algorithm for covering a ground set of $m$ elements.
- $\{opt_m\}$ is a sequence of random variables, with $opt_m$ being the optimum solution value of a random $SC$ instance $(m, p)$.
- Let $\{C_k\}$, $k \in \{1, \ldots, m\}$ be a sequence of random variables with $C_k$ being the number of ground elements newly covered by the $k$-th subset entering the solution set. For $k > S_m$ we define $C_k = 0$.
- Let $\{U_k\}$, $k \in \{1, \ldots, m\}$ be a sequence of random variables with $U_k$ denoting the number of elements remaining uncovered after entrance of the $k$-th subset in the solution set. For $k > S_m$ we define $U_k = 0$.
- The indicator random variables $a_{ij}$, for $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$, denote whether the ground element $x_i \in X$ is covered by the subset $A_j \in \mathcal{F}$. Obviously for a random $SC$ instance $(m, p)$ $\Pr[a_{ij} = 1] = p$.

By the description of the simple algorithm one can observe that:

$$U_k \leqslant U_{k-1},\ U_k + \sum_{l=1}^{k} C_l = m$$

The following lemma determines such a value for $k_0$ that $E[C_k] \geqslant 1$ for all $k \leqslant k_0$:

**Lemma 1** *Consider a random* SC *instance* $(m, p)$*. Then:*

$$E[U_{k_0}] < \frac{1}{p},\ \ k_0 = \lfloor 1 - \frac{\log(pm)}{\log(1-p)} \rfloor$$

**Proof.** From the description of the simple algorithm, just before the $k$-th step, $U_{k-1}$ ground elements remain uncovered, so that the following random sum of i.i.d. Bernoulli variables expresses the number of ground elements to be covered at the $k$-th step:

$$C_k = \sum_{r=1}^{U_{k-1}} a_{i_r j_k}$$

For $U_k$ we have $U_k = U_{k-1} - C_k$. Observe that $U_{k-1}$ is pretty independent of $a_{i_r j_k}$ (see [3], chapter 12), so that because of the binomial distribution of the ground elements to subsets:

$$E[C_k] = pE[U_{k-1}],\ E[U_k] = (1-p)E[U_{k-1}] \tag{1}$$

Assuming $U_0 = m$, $E[U_0] = m$, one easily derives the closed forms:

$$E[U_k] = (1-p)^k m \tag{2}$$

$$E[C_k] = p(1-p)^{k-1} m \tag{3}$$

We identify an upper bound $k_0$ on the step index $k$, so that $E[C_k] \geqslant 1$ for all $k \leqslant k_0$. By equation (3):

$$k_0 = \lfloor 1 - \frac{\log(pm)}{\log(1-p)} \rfloor \tag{4}$$

Now observe that $k_0 > -\log(pm)/\log(1-p)$. Thus we get an upper bound for $E[U_{k_0}]$, by substituting $-\log(pm)/\log(1-p)$ for $k$ in equation (2) and this completes the proof.□

### 3.1 Approximation

At this point we prove an analytical tight upper bound on the expected cardinality of solutions constructed by the simple algorithm for random instances:

**Theorem 2** *Given a random* SC *instance* $(m, p)$*,* $|X| = m$*,* $|\mathcal{F}| = n$*, the simple algorithm returns a solution* $S$ *of cardinality* $S_m$*,* $E[S_m]$ *being bounded by:*

$$-\frac{\log(pm)}{\log(1-p)} < E[S_m] < \frac{1}{p} + 1 - \frac{\log(pm)}{\log(1-p)}$$

**Proof.** Observe that by the simple algorithm's description, $S_m = k_0 + U_{k_0} \geqslant k_0$. Thus:

$$E[S_m] = k_0 + E[U_{k_0}] \tag{5}$$

Substituting in equation (5) the values obtained from lemma 1 the result follows.□

The central result of our work makes use of the following theorem:

**Theorem 3** *([11], Theorem 3.1) The sequence of random variables* $\{opt_m\}$ *satisfies:*

$$\lim_{m\to\infty} \frac{opt_m}{\log m} = -\frac{1}{\log(1-p)} \text{ almost everywhere (a.e.)}$$

By combining theorems 2 and 3 it is shown:

**Theorem 4** *Random* SC *is on average approximated within a term of* $o(\log m)$ *from the optimum almost everywhere.*

**Proof.** The proof follows by theorem 2:

$$\lim_{m\to\infty} \frac{E[S_m]}{\log m} = \frac{-1}{\log(1-p)}$$

By theorem 3: $\lim_{m\to\infty} \frac{E[S_m] - opt_m}{\log m} = 0$ a.e., thus $E[S_m] = opt_m + o(\log m)$ a.e.□

The result of theorem 4 would be of diminished value if solutions built by the simple algorithm exhibited large variance. This does not hold however. The following is proved:

**Lemma 2** $V[U_k] = E[U_k]$, *for* $k \leqslant k_0$.

**Proof.** It is possible to calculate the variance of $U_k$, because $U_{k-1}$ is independent of $a_{i_r j_k}$ (see [3], chapter 12):

$$U_k = \sum_{r=1}^{U_{k-1}} (1 - a_{i_r j_k}) \Rightarrow$$

$$V[U_k] = V[(1 - a_{i_1 j_k})]E[U_{k-1}] + \Big(E[(1 - a_{i_1 j_k})]\Big)^2 V[U_{k-1}] \Rightarrow$$

$$V[U_k] = p(1-p)E[U_{k-1}] + (1-p)^2 V[U_{k-1}]$$

This is a recurrent relation, with a fortunate terminating condition: $U_1$ is the number of ground elements remaining uncovered after the first step (selection of the first subset), and is binomially distributed. Thus $V[U_1] = p(1-p)m$. Lemma 1 and, in particular, equation (2) gives $E[U_k]$ for $k \leqslant k_0$. The following closed form is thus obtained:

$$V[U_k] = p\sum_{l=1}^{k-1} (1-p)^{2(k-l)-1} E[U_l] + (1-p)^{2(k-1)} V[U_1]$$

Manipulation of this equation after substitution of $E[U_l]$ and $V[U_1]$ yields $V[U_k] = (1-p)^k m = E[U_k]$.□

**Theorem 5** *For the class of random* SC *instances* $(m, p)$ *the simple algorithm yields solution values with* $O(1)$ *variance.*

**Proof.** The variance of produced solutions is first calculated:

$$V[S_m] = E[S_m^2] - (E[S_m])^2 \tag{6}$$

Because $S_m = k_0 + U_{k_0}$ equation (6) becomes:

$$V[S_m] = E[(k_0 + U_{k_0})^2] - \Big(k_0 + E[U_{k_0}]\Big)^2 \Rightarrow$$

$$V[S_m] = E[U_{k_0}^2] - \Big(E[U_{k_0}]\Big)^2 \Rightarrow V[S_m] = V[U_{k_0}] \tag{7}$$

By lemmas 2 and 1 $V[S_m] = E[U_{k_0}] < 1/p$.□

### 3.2 Sensitivity Analysis

In this section we perform sensitivity analysis on simple solutions with respect to increments of the instance's data. That is, examination of the possibility that an existent solution remains feasible despite the instance's augmentation with random constraints. The situation of the instance's input data being altered after a solution has been calculated is of interest in several contexts such as in evaluation of reliability bounds [10]. Here we prove a theorem analogous to the result of [4], where $O(\log m)$ element insertions in a $SC$ instance do not alter the existent solution's approximation properties with respect to the optimum value of the novel instance.

The constraints of the $SC$ correspond to the coverage of all ground elements. Augmentation with new constraints essentially corresponds to introduction of new ground elements, and their insertion to each of the subsets of $\mathcal{F}$ with the same probability $p$.

**Theorem 6** *A simple solution $S$ for a random* SC *instance $(m, p)$, $|X| = m$, remains feasible under $\omega(m)$ element insertions on average.*

**Proof.** Assume augmentation of $X$ with still one ground element, and appropriate insertion of the element in some subsets of $\mathcal{F}$ under independent Bernoulli trials of probability $p$. Let $S \subseteq \mathcal{F}$ be a solution produced by the simple algorithm, with $|S| = S_m = s$. Furthermore, $T_m$ is a random variable that counts the number of new ground element insertions before the existent solution becomes infeasible. If $x$ is the newly inserted element:

$$\Pr[x \notin (\cup_{A \in S} A) | S_m = s] = (1-p)^s \Rightarrow$$

$$E[T_m | S_m = s] = (1-p)^{-s}$$

In order to estimate $E[T_m]$, since $1 \leqslant S_m \leqslant m$, we have:

$$E[T_m] = \sum_{s=1}^{m} \Big( E[T_m | S_m = s] \Pr[S_m = s] \Big) \Rightarrow$$

$$E[T_m] = \sum_{s=1}^{m} \Big( (1-p)^{-s} \Pr[S_m = s] \Big) = \sum_{s=1}^{m} \frac{m \Pr[S_m = s]}{(1-p)^s m} \Rightarrow$$

$$E[T_m] = m \Big[ \sum_{s=1}^{k_0} \frac{\Pr[S_m = s]}{(1-p)^s m} + \sum_{s=k_0+1}^{m} \frac{\Pr[S_m = s]}{(1-p)^s m} \Big] \tag{8}$$

By lemma 1, for $s \geqslant k_0 + 1$ we have:

$$(1-p)^s m \leqslant (1-p)^{k_0+1} m = (1-p) E[U_{k_0}] \Rightarrow (1-p)^s m < \frac{1-p}{p}$$

On the other hand, for $s \leqslant k_0$, $(1-p)^s m = E[U_k] \leqslant m$. So, expression 8 becomes:

$$E[T_m] > m \Big[ \sum_{s=1}^{k_0} \frac{\Pr[S_m = s]}{m} + \frac{p}{1-p} \sum_{s=k_0+1}^{m} \Pr[S_m = s] \Big] \Rightarrow$$

$$E[T_m] > c \frac{p}{1-p} m, \;\; c = \sum_{s=k_0+1}^{m} \Pr[S_m = s] \neq 0$$

Thus $E[T_m]$ has a strict lower bounding order of $\omega(m)$.□

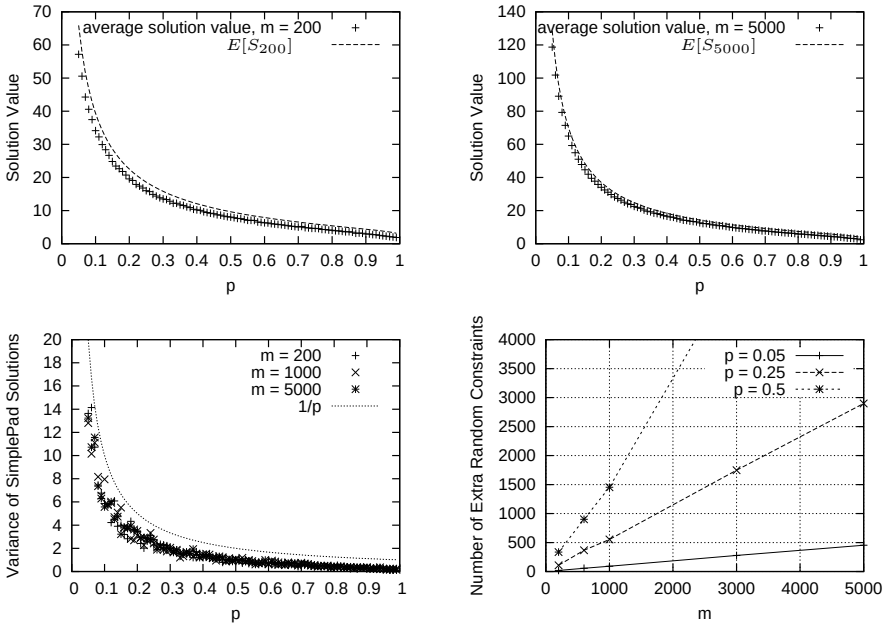


Figure 1: Averages for simple solution values, variance, and average number of extra feasibility preserving constraints.

# 4 Experimental Illustration

We present some experiments that are in perfect concordance with the theoretical results. Experiments were conducted for instances with $m \in \{200, 600, 1000, 3000, 5000\}$ and $0.05 \leqslant p \leqslant 1.0$, with a step of $10^{-2}$. For each parameter combination 1000 random instances were generated. We report on average solution values and on the average number of extra feasibility preserving random constraints added to each instance. The unbiased variance estimator of solutions produced by the simple algorithm is also depicted for the aforementioned classes of instances, and is shown to be bounded by $1/p$.

The upper two diagrams of fig. 1 depict average of solution values for instances $(200, p)$ and $(5000, p)$, and the corresponding curves obtained by theorem 2. The curves upper bound the averages produced by the experiments as was expected. The unbiased variance estimator of experimental simple solution values is shown to be bounded by $1/p$ on the lower left diagram, whereas the linear dependence on $m$ of average number of feasibility preserving random constraints is depicted on the lower right graph.

# 5 Conclusions

In this work we have developed the first (to the best of our knowledge) average case absolute error bound in approximating the class of random set covering instances. The value of this bound was further strengthened by the favorably low variance of solution values produced by the simple algorithm. Simple solutions were also shown to be extremely robust with respect to perturbation of input data incurred by augmentation of additional random constraints.

It appears that on average, the simple algorithm produces solutions with a small deviation from the optimum. It is a matter of future work to show that the smarter greedy algorithm performs even better (it does so in practice) by lessening or possibly eliminating the deviation. Our results provide strong intuition that random $SC$ instances may constitute a broad class of

instances for which the greedy algorithm proves to be optimum on average. It was empirically shown in [7] that random instances are the field of weak performance of several intelligent heuristics, outperformed by the greedy algorithm. Although random instances seldom occur in practice, they present a challenge for complex heuristics, whereas they are well approximated by the greedy algorithm and, as this work intrigues, by the simple algorithm.

It seems that the analysis for the random set covering can extend to handle average case analysis of maximum k-covers, where the target is to maximize the number of covered ground elements when a solution consists of precisely $k$ subsets.

## References


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

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

[3] W. Feller. *An Introduction to Probability Theory and its Applications, Third Edition, Volume I.* Wiley International, 1967.

[4] G. Gambosi, M. Protasi, and M. Talamo. Note: Preserving approximation in the min-weighted set cover problem. *Discrete Applied Mathematics*, 73:13–22, 1997.

[5] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and Systems Sciences*, 9:256–278, 1974.

[6] L. Lóvasz. On the ratio of optimal integer and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[7] E. Marchiori and A. Steenbeek. An iterated heuristic algorithm for the set covering problem. In *Proceedings of the 2nd Workshop on Algorithm Engineering, WAE'98*, pages 1–12, August 1998.

[8] V. T. Paschos. A survey of approximately optimal solutions to some covering and packing problems. *ACM Computing Surveys*, 29(2), June 1997.

[9] P. Slavík. A tight analysis of the greedy algorithm for set cover. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 435–439, 1996.

[10] S. Tsitmidelis, M. Koutras, and V. Zissimopoulos. Evaluation of reliability bounds by set covering models. *Statistics and Probability Letters*, 61(2):163–175, 2003.

[11] C. Vercellis. A probabilistic analysis of the set covering problem. *Annals of Operations Research*, 1:255–271, 1984.

[12] B. W. Weide. *Statistical methods in algorithm design and analysis.* PhD thesis, Carnegie Mellon University, 1978.