JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 23 No. 1 (2015), pp. 29-45

Graph Security Testing

Tomasz Gieniusz[1], Robert Lewoń[1], Michał Małafiejski[1]

[1]*Gdańsk University of Technology, Poland*
Department of Algorithms and System Modeling
Gabriela Narutowicza 11/12, 80-233 Gda´nsk Wrzeszcz
tomasz.gieniusz@gmail.com lewon.robert@gmail.com mima@kaims.pl

Abstract. *In this paper we consider probabilistic approach to the decision problem of security in graphs. In this purpose we define general model (called* property tester*) and criteria for approximating answers for decision problems. We constructed two property testers and one heuristics for the problem of security in graphs.*
Keywords: *secure set, property testing.*

1. Introduction

Set $S \subset V$ is called *secure set* iff $\forall_{X \subset S} |N[X] \cap S| \geq |N(X) \setminus S|$ [1]. That means that every subset of a secure set has at least as many friends (neighbour vertices in S) as enemies (neighbour vertices outside S) and will be defended in case of attack. Problem of determining if given set is secure is $co-\mathcal{NP}$-complete, there is no efficient algorithm solving it [1].

Property testers are algorithms that distinguish inputs with a given property from those that are *far* from satisfying the property [2]. Property testers are allowed to be probabilistic and approximate in exchange for much reduced complexity.

In our work we apply the idea of testing to construct probabilistic and approximate algorithms for graph security problems that run in polynomial time. Our goal was to formalize the testing model and propose various approaches for security

testing and general methods of rating them. We take into account possibilities of practical application, algorithm complexity and quality of the approximation.

2. Property testing

Definition 1 (Property tester [2][3]) *Property testers are algorithms that distinguish inputs with a given property from those that are far from satisfying the property.* Far *means that significant part of the input must be changed so that property can be satisfied.*

2.1. Tester response

Classical deterministic algorithm for decision problem answers the question

Does the given input satisfy that property?

Property tester for decision problem answers the question

Does the given input satisfy that property or is it far from satisfying it?

2.2. Tester requirements

We require our property testers to satisfy two properties:

- Accept secure set with probability > 95%.
- Reject set that is ε-far from being secure with probability > 95%.

If input set is close to being secure we don't put any requirements on the given answer.

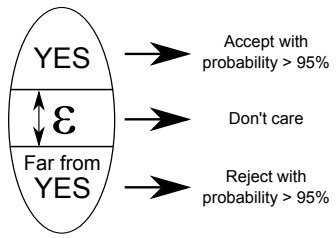

Figure 1. Property tester requirements

2.3. Meaning of ε-far

Meaning of ε-far depends on chosen model and can have impact on:

- Possibilities of practical application.
- Algorithm complexity.
- Fraction of cases we can guarantee the probability of answer correctness.

2.4. Test witness

Witness is a fragment of input or structure generated basing on input. For example number from given input sequence.*Witness testimony* is its property, that can be an evidence of some global input property. For example unordered pair of sequence elements proved it's global unsortedness.

Remark 2 *Testimonies of some witnesses may not entail any informations about global input property. For example ordered pair of sequence elements tells nothing about its global sortedness.*

Witness testimony is *valuable*, if it entails some information about global input property. We call witness *valuable* if its testimony is valuable.

Lemma 3 (Witness lemma) *If a single test catches a valuable witness with a probability* $\geq p$*, than* $s = 3/p$ *iterations of the test catch a witness with probability* $> 95\%$.

Proof. Probability of not catching a valuable witness with s tries does not exceed $P \leq (1-p)^s$. Using $1 - x \leq e^{-x}$, we get $P \leq e^{-ps} = e^{-3} \leq 95\%$. □

3. Formal testing model

3.1. Model definition

Proposed testing model contains of:

Definition of ε-far,
Specification of attributes input must have to be ε-far from satisfying given property.

Definition of witness and it's evidence,
Specification of local and easily verifiable property that can give evidence of given global property.

Witness choose algorithm,
Specification of witness selection method from the set of every possible witnesses.

Witness testimony extraction algorithm,
Specification of method to designate if local property is satisfied for chosen witness. Together with witness choose algorithm it specified probability p_s of catching valuable witness.

3.2. Model rating criteria

For a single problem it is possible to construct many correct testing models. Here is the general model rating schema which fulfilment is required for model usefulness.

Practical and intuitive understanding of ε-far.
Is information that input is ε-far from satisfying given property is valuable for us.

Uniform witness choose.
Does every witness have the same probability to be chosen. If, despite of many valuable witnesses, our algorithm somehow ignores them than we can give guarantee on probability of answer correctness. If it is not uniform distribution than we can not use lemma 3.

Relationship between ε and p_s.
How does probability of choosing valuable witness change depending on how far input is far from satisfying given property.

Choose and check of witness algorithms complexity.
What is the complexity of witness choose and witness testimony check algorithms and is it an advantage over deterministic algorithm.

Tester complexity can be described us: $O(3/p_s \cdot f(I) \cdot g(I))$, where

- $3/p_s$ is number of necessary witness tests to be repeated,
- $f(I)$ is witness choose algorithm complexity,
- $g(I)$ is witness check algorithm complexity.

Fraction of cases that we can give probability guarantee on.
How fraction of cases that we can give probability guarantee on changes depending on ε. Other words, how small ε is required to give this guarantee on at least δ fraction of possible inputs. If for given definition of ε-far significant fraction of input cases is very close to satisfy given property, than we will need very small ε for that cases to be considered as ε-far. That in other hand has impact on algorithm complexity.

3.3. Problem decomposition

Here we present a way to construct proper testing model using problem decomposition.

Definition 4 (General predicate) *Let our decision problem be predicate on set of all possible inputs and call it general.*
$P_G : \mathbb{I} \to \{true, false\}$*, where* $\mathbb{I}$ *is a set of all possible problem inputs.*

Definition 5 (Special predicate) *Let* $D(I) = \{I_1, I_2, \ldots, I_k\}$ *be a function defining an input decomposition and* $P_S : D(I) \to \{true, false\}$ *a predicate on the element of that decomposition (part of the input), such that* $\forall_{I \in \mathbb{I}}\ P_G(I) = true \Leftrightarrow \forall_{I_i \in D(I)}\ P_S(I_i) = true$*. P_S is called special predicate.*

4. Security in graphs

In this section we present basic graph security definitions derived from [4][5][1][6] and reworded for better readability.

Nonempty set $S \subseteq V$ in graph $G = (V, E)$ is called an *alliance*, if $\forall_{v \in S} \ |N[v] \cap S| \geq |N(v) \cap V \setminus S|$. That means that every member of an alliance (vertex in S) has at least as many friends (neighbour vertices in S) as enemies (neighbour vertices outside S) and will be defended in case of attack. In addition if S is also dominating set of G we call it *global alliance*.

Definition 6 *For set $X \subseteq V$ we define predicate* $\mathrm{SEC}(X) \Leftrightarrow |N[X \cap S]| \geq |N(X \cap V \setminus S)|$.

Set S is called *k-alliance*, if $\forall_{X \subseteq S} \ |X| \leq k \Rightarrow \mathrm{SEC}(X)$ and again when S is dominating set of G we call it *global k-alliance*.

For nonempty subset $S = (v_1, v_2, \ldots, v_k) \subseteq V$ in graph $G = (V, E)$, *attack* on S is every sequence of k mutually disjoint sets $A = (A_1, A_2, \ldots, A_k)$, such that $\forall_{1 \leq i \leq k} \ \forall_{a \in A_i} \ a \in N(v) \setminus S$. If $a \in A_i$ we say that *a attacks* v_i. Similarly we define *defence* of S as every sequence of k mutually disjoint sets $D = (D_1, D_2, \ldots, D_k)$, such that $\forall_{1 \leq i \leq k} \ \forall_{d \in D_i} \ a \in N[s_i] \cap S$. If $d \in D_i$ we say that *d defends* v_i. Attack A on S *can be defended*, if there exists defence D such that $\forall_{1 \leq i \leq k} \ |D_i| \geq |A_i|$. Set S is *secure* iff every attack on S can be defended.

Theorem 7 (Secure set as *k*-alliance) *Set S is secure in graph G iff it is global $|S|$-alliance.*

Proof. This theorem was proved in [1]. □

5. Security testing

5.1. Attack based model

The simplest, natural way for security testing is to take into account all possible maximal attacks and require given fraction of them to be defendable.

5.1.1. Model

Definition of ε-far,

S is ε-far from being secure iff ε fraction of attacks cannot be defended. For example set is $1/2$-far from being secure iff half of the possible attacks cannot be defended.

Definition of witness and it's evidence,
Maximal attack on S is a witness, defence possibility is his testimony. Witness is valuable if there is not defence for him.

Witness choose algorithm,
For every enemy we choose his target uniformly and independently at random (his neighbour in S). This way we get a maximal attack, because every enemy is making use of his attack possibility.

Witness testimony extraction algorithm,
To check if attack is defendable we use security matching algorithm described in [4].

5.1.2. Model rating

Practical and intuitive understanding of ε-far,
ε-far is defined that way to be the most intuitive. Enemies usually don't choose their targets at random, so practical application may be narrowed, but it is still valuable information to know that for example half of the attacks cannot be defended.

Uniform witness choose,
Every maximal attack is chosen with the same probability, because every of them can be constructed in exactly one way and there is no such attack, algorithm couldn't construct.

Relationship between ε and p_s,
Relationship is linear, set is ε-far from being secure iff ε fraction of witnesses (attacks) is undefendable, so they are valuable witnesses.

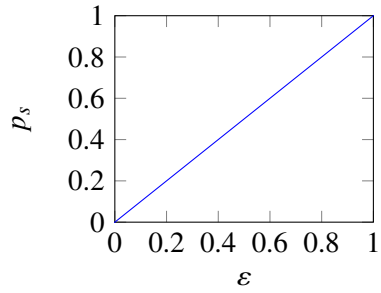

Figure 2. Relationship between ε and p_s in attack based model

Choose and check of witness algorithms complexity,

Witness choose can be done in $O(|V \setminus S||S|)$ - for every enemy we choose his neighbour to be the target of the attack.
Witness check is determined by security matching algorithm [4] and is $O(n^{5/2})$.

Fraction of cases that we can give probability guarantee on,

We have done computer simulations for random graphs with 10 – 20 nodes. Results are shown on the plot below.

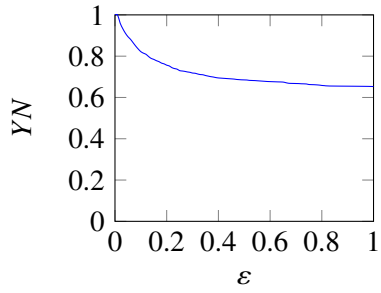

Figure 3. Fraction of cases that we can give probability guarantee on attack based model

That means that even for high ε (low algorithm complexity) random graph is either secure or ε-far from being secure (there are many witnesses) and we can give probability guarantee on algorithm answer.

5.1.3. Algorithm

Algorithm 5.1 Attack based tester

Repeat $3/\varepsilon$ **times**
Chose maximal attack A independently and uniformly at random A.
if A cannot be defended **then**
return S is ε-far from being secure.
end if
return S is secure.
End

5.1.4. Analysis

- Secure set will always be correctly determined.
- For the set that is ε-far from being secure, probability of choosing undefendable attack in one test is no less than ε.
- Using witness lemma 3 $3/\varepsilon$ test repetitions chooses undefendable attack in no less than 95% of the cases.

5.1.5. Conclusions

- Probability of failure to recognize insecure set despite of ε fraction of undefendable attacks is no more than 5%.
- Algorithm complexity is $O(n^{5/2}/\varepsilon)$.

5.1.6. Examples

Troubling example

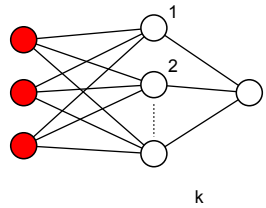

Figure 4. Troubling example for attack based model

In the above graphs family only attacks that are focused on single vertex cannot be defended. There is exactly k of such attacks and in set of all k^3 attacks they are only $1/k^2$ fraction. That means that for any ε we can construct a graph, that is close to being secure and our algorithm cannot guarantee the correctness of the given answer.

Easy example

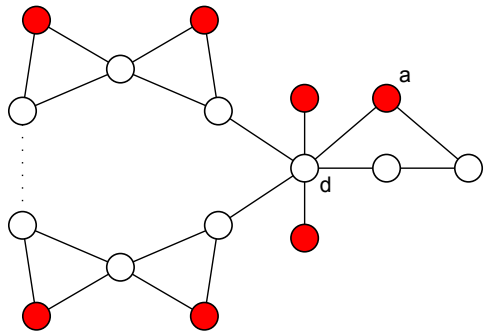

Figure 5. Easy example for attack based model

In the above graphs family exactly half of the attacks cannot be defended. Cycle can be defended independently, every attack in which a chooses d as his target cannot be defended. That means that our algorithm has 50% chance to choose valuable witness in single test and according to witness lemma 3 that gives 95% success rate with only 6 repetitions.

5.2. Subset based model

Next model is derived directly from subset secure set definition [7]. In this case we will take into account all possible subsets of S and require given fraction of them to satisfy [6].

5.2.1. Model

Definition of ε-far,

S is ε-far from being secure iff ε fraction of subsets doesn't satisfy [6].

Definition of witness and it's evidence,
Subset $A \subseteq S$ is a witness and value of SEC(A) is his testimony. Witness is valuable if his testimony equals $false$.

Witness choose algorithm,
For every vertex $v \in S$ choose one possibility uniformly and independently at random:

- v is in witness,
- v is not in witness.

Witness testimony extraction algorithm.
Compute SEC(A) from definition.

5.2.2. Model rating

Lets use 3.2 for such testing model.

Practical and intuitive understanding of ε-far,
This model derives from security definition, so the definition of ε-far is very intuitive. It has practical applications, because it is a valuable information to know that for example half of the subsets are not secure.

Uniform witness choose,
From the set of all possible subsets of S every one is chosen with the same probability, because every of them can be constructed in exactly one way and there is no such subset that algorithm could not construct.

Relationship between ε and p_s,
Relationship is linear, set is ε-far from being secure iff exactly ε fraction of witnesses (subsets) doesn't satisfy [6], so they are valuable.

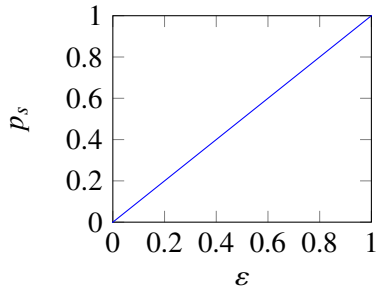

Figure 6. Relationship between ε and p_s for subset based model

Choose and check of witness algorithms complexity,

Witness choose complexity is $O(|S|)$ - for every ally we choose if it is in the witness. Witness check complexity is determined by SEC definition and is $O(n)$.

Fraction of cases that we can give probability guarantee on.

We have done computer simulations for random graphs with 10 – 20 nodes. Results are shown on the plot below.

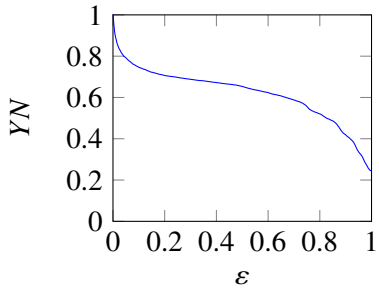

Figure 7. Fraction of cases that we can give probability guarantee on subset based model

That means that even for high ε (low algorithm complexity) random graph is either secure or ε-far from being secure (there are many witnesses) and we can give probability guarantee on algorithm answer.

5.2.3. Algorithm

Algorithm 5.2 Subset based tester

Repeat $3/\varepsilon$ **times**
  Choose subset $X \subseteq S$ uniformly and independently at random.
  if $\sim$ SEC(X) [6] **then**
    return S is ε-far fro being secure.
  end if
  return S is secure.
End

5.2.4. Analysis

- Secure set will always be correctly determined.
- For the set that is ε-far from being secure, probability of choosing subset that doesn't satisfy SEC in one test is no less than ε.
- Using witness lemma 3 $3/\varepsilon$ test repetitions chooses such subset in no less than 95% of the cases.

5.2.5. Conclusions

- Probability of failure to recognize insecure set despite of ε fraction of undefendable attacks is no more than 5%.
- Algorithm complexity is $O(n^2/\varepsilon)$.

5.2.6. Examples

Troubling example

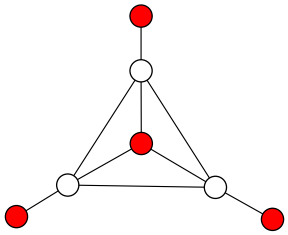

Figure 8. Troubling example for subset based model

In the above graphs family SEC is not satisfied for only the whole S. In set of every $2^{|S|}$ possible subsets of S it is only fraction of $1/2^{|S|}$. That means that for any ε we can construct a graph, that is close to being secure and our algorithm cannot guarantee the correctness of the given answer.

Easy example

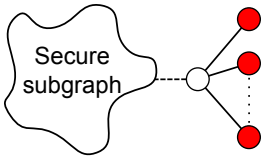

Figure 9. Easy example for subset based model

In the above graphs family SEC is not satisfied for exactly half of the subsets. That means that our algorithm has 50% chance to choose valuable witness in single test and according to witness lemma 3 that gives 95% success rate with only 6 repetitions.

6. Security problem heuristic

In this section we will present heuristic algorithms based on simple idea. Idea is to construct subset that is a valuable witness for security problem in subset based model - subset that does not satisfy SEC. Specific versions of this algorithm differs

in subset construction approach.

We will use following naming convention:

- $S \subseteq V$ is a subset of vertices to be checked for security
- c and d are optimization criteria for any subset $S' \subseteq S$

6.1. Algorithm schema

Algorithm 6.1 Security problem heuristic schema

Split S into n disjoint clusters $S_1, S_2, ..., S_n$
repeat
  For every pair of clusters S_i, S_j, $i \neq j$ calculate $c(S_i \cup S_j)$
  Join two clusters optimizing c, in the case of ambiguity use d criteria
until There are at least two clusters or one of the clusters does not satisfy SEC
if Any of the clusters does not satisfy SEC **then**
  return false
else
  return true
end if

Remark 8 *First step of the algorithm can be randomized (and algorithm can be repeated many times). The simplest version is to take one-element clusters, for* $S = \{s_1, s_2, ..., s_k\}$ *take* $S_i = \{s_i\}$, $i = 1, 2, ..., k$ *as initial clusters.*

In all presented variant of algorithm we take the same c criteria, for every $X \subseteq S$:

$$c(X) = |N[X] \cap S| - |N(X) \setminus S|$$

We will minimize this criteria, negative value means that X does not satisfy SEC, therefore S is not secure. Algorithm objective is to construct valuable witness, so minimization of c is well founded.

6.2. Secondary criteria proposals

Second criteria is used in the case of ambiguity of c criteria. In such situation we propose following approaches:

Avoid massive clusters creation

For any $X \subseteq S$:

$$d_1 = |X| .$$

This will lead to small clusters creation.

Support massive and strong clusters creation

For any $X \subseteq S$:

$$d_2 = |N[X] \cap S| .$$

This will lead to bigger clusters with greater support from the outside creation.

Support neighbourhood-dependent clusters creation

For any $X \subseteq S$:

$$d_3 = |(N[X] \setminus X) \cap S| .$$

This will lead to creation of the clusters that require maximum help from neighbours outside of X.

Remark 9 *In the simplest version secondary criteria can be omitted. We can also use many criteria by choosing secondary criteria randomly at every iteration. In case of further ambiguity after second criteria usage we can introduce additional resolve phases or choose cluster at random.*

References

[1] Dutton, R., *Secure set algorithms and complexity*, In: Proceedings of the Thirty-Seventh Southeastern International Conference on Combinatorics, Graph Theory and Computing, Vol. 180, 2006, pp. 115–121.

[2] Raskhodnikova, S., *Property Testing: Theory and Applications*, Ph.D. thesis, Massachusetts Institute of Technology, 2003.

[3] Goldreich, O., Goldwasser, S., and Ron, D., *Property testing and its connection to learning and approximation*, Journal of the ACM (JACM), Vol. 45, No. 4, 1998, pp. 653–750.

[4] Blukis, T., *Zbiory bezpieczne w grafach*, Master's thesis, Politechnika Gdańka, 2012.

[5] Brigham, R., Dutton, R., and Hedetniemi, S., *Security in graphs*, Discrete applied mathematics, Vol. 155, No. 13, 2007, pp. 1708–1714.

[6] Dutton, R. D., Lee, R., and Brigham, R. C., *Bounds on a graph's security number*, Discrete Appl. Math., Vol. 156, No. 5, March 2008, pp. 695–704.