

Packing triangles is harder than previously thought

THINH D. NGUYEN*

Moscow State University
kosmofarmer@gmail.com

July 9, 2018

Abstract

In this work, we will take one problem, namely PACKING TRIANGLES as an example of combinatorial optimization problems. We show that if one has ever loved reading Prasadlov's books, then one should not try to find efficient algorithm for various restricted cases of this problem.

I. PROPOSITIONS

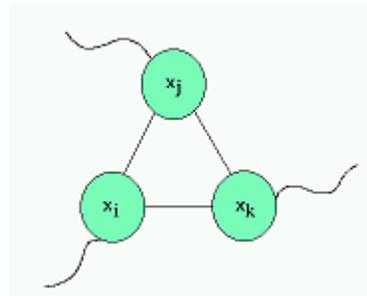
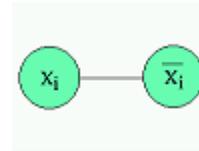
A classical result claims that it is NP-complete to test satisfiability of Boolean formulae. By simple manipulations, we refine this to the following statement.

Proposition 1: SAT remains very hard even when restricted to formulae in which every variable appears exactly three times, two times in *positive* and one time in *negative*.

Proof: The idea is simple. Pick one variable, call it x . If x appears k times in our formula, we sort all of these occurrences in arbitrary order x_1, x_2, \dots, x_k . What we will do is to make these into k different new variables. And of course, we need to make sure all of these need to be assigned the same truth value. So, we do this in a circular way to minimize the number of new clauses added. In particular, by adding more clauses stating x_i implies x_{i+1} , indices are circular ($k + 1 = 1$). And, in logic, a implies b is simply $\neg b \vee a$. **Q.E.D.**

This result and the following one have been all achieved by [1], maybe not to the same restrictedness.

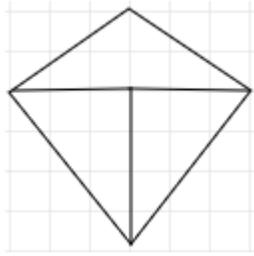
*Perebor



Proposition 2: VERTEX COVER and INDEPENDENT SET are all very hard.

Proof: For each variable x , we create two vertices. And for each clause, we create three vertices like in the figure. The edges are obvious from the construction that is being described. We connect each clause vertex to the corresponding variable vertex.

Thus, we turn the SAT problem into VERTEX COVER and INDEPENDENT SET. We remind that *vertex cover* and *independent set* always come in complement pair. Also, if any clause has only 2



literals, instead of a K_3 graph, we create, for it, a K_2 graph. The details of the *bijective mapping* between satisfying assignment, vertex cover and independent set can be easily obtained with *Prasolov* way of thinking mathematically. **Q.E.D.**

We now have a graph with maximum vertex degree 3. To be strictly 3-regular, we connect each degree-2 vertex to the topmost vertex of the above graph.

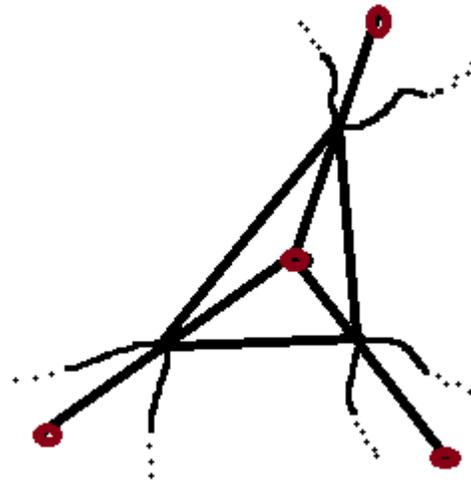
The following statement affirms that the added graphs do not affect our construction.

Proposition 3: The added graphs always require 3 vertices to cover, regardless of the attached (bigger) graph.

Proof: By enumerating all cases, we can claim the correctness of *Proposition 3*. Note that, the edge connecting this graph to the larger graph also needs to be covered, if the vertex where this graph is attached to is not chosen in the solution to VERTEX COVER. **Q.E.D.**

Now, we have a 3-regular graph and continue to transform this into a 4-regular graph. We do this by assign each edge of our 3-regular graph to a new vertex. Each edge in our new 4-regular graph will correspond to one pair of incident edges. We illustrate this in the figure.

To complete our construction, we want to get back to the realm of 3-regularity. To achieve this, we *split* each vertex of the 4-regular graph in to two vertices while respecting the incidence. To illustrate this last manipulation, we give the last illustration. Each triangle must correspond to one of the vertex in the original



graph. We can see here the eternal love for *Prasolov* mathematics moving on post-Soviet.

PROPOSITION:

$$SAT \leq_p \text{PACKING TRIANGLE}$$

And otherwise, if we combine this to a result published in [2, Proposition 2.1], we have $P = NP$. So, we leave the work here for the rest of the mathematics community to do.

II. CONCLUSION

As long as we do the research on a well-known conjecture, we should recall our mathematical nature from *Kvant*, *Prasolov*-style of doing mathematics, similar to mathematics of [3] back to those beautiful days.

REFERENCES

- [1] Michael R. Garey, David S. Johnson, **Computers and Intractability: A Guide to the Theory of NP-Completeness**
- [2] Alberto Caprara, Romeo Rizzi, **Packing triangles in bounded degree graphs**, Information Processing Letters 84 (2002) 175-180
- [3] Phan Dinh Dieu, Le Cong Thanh, Le Tuan Hoa, **Average Polynomial Time Complexity of Some NP-Complete Problems**. Theor. Comput. Sci. 46(3): 219-237 (1986)