

Optimal Approximation Algorithms for Reoptimization of Constraint Satisfaction Problems

Victor Alex Mikhailyuk

V.M. Glushkov Institute of Cybernetics, National Academy of Sciences of Ukraine, Kyiv, Ukraine

Email: mikhailyukvictor2@gmail.com

Received November 23, 2012; revised December 25, 2012; accepted January 8, 2013

ABSTRACT

The purpose of reoptimization using approximation methods—application of knowledge about the solution of the initial instance I , provided to achieve a better quality of approximation (approximation ratio) of an algorithm for determining optimal or close to it solutions of some “minor” changes of instance I . To solve the problem *Ins-Max-EkCSP-P* (reoptimization of *Max-EkCSP-P* with the addition of one constraint) with approximation resistant predicate P exists a polynomial threshold (optimal) $\psi(q(P))$ -approximation algorithm, where $\psi(q(P)) = 2 - \frac{1}{q(P)} = 2 - d(P) = 2 - 2^{-k} |P^{-1}(1)|$

($d(P)$ - the threshold “random” approximation ratio of P). When the unique games conjecture (UGC) is hold there exists a polynomial threshold (optimal) $\psi(\alpha_z)$ -approximation algorithm (where $\psi(\alpha_z) = 2 - \frac{1}{\alpha_z}$ and α_z - the integrality gap of semidefinite relaxation of *Max-EkCSP-P* problem Z) to solve the problem *Ins-Max-EkCSP-P*.

Keywords: C -Approximation Algorithm; Reoptimization; Approximation Resistant Predicates; Integrality Gap; Unique Games Conjecture (UGC)

1. Introduction

In the constraint satisfaction problems (or CSP problems), there are many variables and a set of constraints (defined by predicates), each of which depends on a number of variables, and the goal is to find such assigning values to variables that satisfy the maximum number of constraints.

More formally, CSP problem Q is defined by a set of predicates over a finite domain $[q] = \{1, 2, \dots, q\}$. Each instance of problem Q consists of a set of variables V and a set of constraints on it. The goal is to find the assignment to variables that satisfies the maximum number of constraints. In general, the predicates can be replaced with actual payoff functions, and the goal is to maximize the total payment. A large number of fundamental optimization problems, such as Max Cut and Max k -Sat, there are examples of CSP problems.

Most of the CSP problems are NP-hard and so to solve them exactly in a reasonable time is hardly possible. Therefore we considered an effective approximation algorithm for solving such problems. For the maximization problem saying that the algorithm is the C -approximation algorithm, if for any instance gives a solution with objec-

tive function value no less than $\frac{1}{C} \cdot OPT (1 < C)$, where

OPT —the global optimum. In this C is called the approximation ratio. Such a definition can be given to minimization problems.

For the problem Q an upper bound of approximation ratio is established, if there exists a polynomial C -approximation algorithm for solving Q . For the problem Q the lower bound of approximation ratio c is established, if for any $\varepsilon > 0$ there is no polynomial approximation algorithm for Q where the approximation ratio $c - \varepsilon$ (or strictly less than c) is achieved. If $C = c$, then, for the problem Q the threshold of approximation ratio is established (is equal to $C = c$). The corresponding algorithm is called the threshold or optimal (and the approximation ratio-optimal).

A fundamental question for a given NP-hard problem is to determine for which values you can rely on efficient (polynomial) C -approximation algorithm. This is a large research area in theoretical computer science with its positive and negative results.

The problem of establishing lower bounds for the approximation ratio (like any problem of obtaining lower

bounds for the complexity) is a very difficult task. For this problem there is the name of inapproximability or the hardness of approximation. Great influence on the development of methods for obtaining lower bounds the famous PCP theorem [1] and discrete Fourier analysis to test the properties of problems (property testing) are provided [2,3].

Beginning with Goemans and Williamson [4,5] for Max Cut, semidefinite programming (SDP) has become the main tool in the construction of approximation algorithms for the CSP problems. For many of the problems are built SDP relaxations and apply appropriate procedures for the probabilistic rounding the solutions were obtained.

As already noted, the problem of inapproximability has been solved successfully for many of the problems due to PCP theorem. In particular, Hastad [6-8] showed that Max-E3-Lin-2 and Max 3-Sat are NP-hard to approximate with ratios $2 - \epsilon$ and $8/7 - \epsilon$ respectively. This means that a simple random algorithm for assigning is the best (optimal) for these problems, if $P \neq NP$ or that ratios 2 and 8/7 are the threshold. In [9] is showed (also involving PCP theorem) that the set covering problem has a threshold approximation ratio equal to $\ln n$.

In studying the problem of inapproximability for general constraints satisfaction problems (in particular with the predicates of arity 2), such progress was not achieved. The most promising approach to obtaining strong results (thresholds of approximation ratios)—the so called *Unique Games Conjecture (UGC)*, introduced by S. Khot [11-14]. *Unique games conjecture (UGC)* is one of the most important open problems in modern theoretical computer science because of the large number of strong results in inapproximability that follow from the UGC. For example, $2 - \epsilon$ —the hardness of approximating Vertex Cover [12], Max Independent Set [15], Multi Cut [16].

Recently, a close relationship between the concepts of the approximation ratio, the inapproximability ratio and integrality gap of simple SDP relaxation (defined as the maximum ratio of the SDP solution to the real optimum) has been established. From the truth of UGC, it follows that the simple SDP relaxation gives the optimal approximation ratio for CSP. For the first time link between the SDP rounding schemes for relaxation and results in inapproximability based on the UGC, was noticed in [13] for Boolean CSP of two variables. In general, in [17-19] proposed the rounding schemes by which the optimal approximation ratio for each CSP problem, assuming the true UGC, is achieved.

The concept of reoptimization [20-26] is as follows. Let Q -some NP-hard (perhaps NP-complete) problem, I -the initial problem instance of Q , the optimal solution of which is known. We propose a new instance I' of the problem Q , received some “minor” changes of instance I .

The question arises: how can we effectively utilize the knowledge of the optimal solution of I for the calculation of exact or approximate solution of the instance I' ? The purpose of reoptimization using approximation methods - application of knowledge about the solution of the initial instance I , provided either to achieve a better quality of approximation (approximation ratio), or a more efficient (in time) algorithm for determining optimal or close to it solutions, or execution of the first and second points.

Such results for the reoptimization of discrete optimization problems are known. When an elementary disjunction is inserted reoptimization of Max Weighted Sat (weighted satisfiability problem for maximum) approximated with the ratio of 0.81, while Max Weighted Sat - approximable with ratio 0.77 [25]. When inserting a vertex in the graph reoptimization of Min Vertex Cover (minimum vertex cover of a graph) can be approximated with the ratio of 1.5, Min Vertex Cover-with the ratio of 2 [25]. When inserting a vertex (terminal or not) reoptimization of Min Steiner Tree (minimum Steiner tree) can be approximated with the ratio of 1.5, Min Steiner Tree-approximated by the ratio $1 + \ln(3)/2 \approx 1.55$ [21]. When you insert or delete an item from a set, the set covering problem is approximable with ratio $\left(2 - \frac{1}{\ln m + 1}\right)$, where

m —the number of elements. A similar result occurs when you insert or delete an arbitrary number of $1 < p < m$ elements of the set [26]. It should be noted a series of papers on the problem of reoptimization of traveling salesman problem (TSP-Travelling Salesman Problem) [20-22,24]. For example, the problem of Minimum Metric TSP (Min TSP—the traveling salesman problem with the minimum metric distances) approximable with ratio 1.5, its reoptimization when inserting a new unit—with ratio of 1.34, reoptimization of this problem when you change the distance—with ratio of 1.4 [25]. For the general traveling salesman problem (Min TSP) are unknown estimates of approximation ratio as for herself, and for different versions of reoptimization.

The main results of this paper are as follows. We investigated the reoptimization versions of constraint satisfaction problems with predicates of arity k (*Ins-Max-EkCSP-P*) by adding of any constraint. To solve the problem *Ins-Max-EkCSP-P* (reoptimization *Max-EkCSP-P*) with approximation resistant predicates there exists a polynomial threshold (optimal) $\psi(q(P))$ —approximation algorithm, where

$$\psi(q(P)) = 2 - \frac{1}{q(P)} = 2 - d(P) = 2 - 2^{-k} |P^{-1}(1)|$$

($d(P)$ —the threshold “random” approximation ratio).

When the unique games conjecture (UGC) is hold there exists a polynomial threshold (optimal) $\psi(\alpha_z)$ —appro-

ximation algorithm (where $\psi(\alpha_z) = 2 - \frac{1}{\alpha_z}$ and α_z — the integrality gap of semidefinite relaxation of *Max-EkCSP-P* problem Z) to solve the problem *Ins-Max-EkCSP-P*.

2. Preliminaries

Present the necessary notations and definitions [6,28,31]. Under the predicate P of arity k we mean the map $P: \{-1,1\}^k \rightarrow \{0,1\}$. For notational convenience, the input data with a value of -1 is interpreted as “true”, value of 1 as “false”. If the predicate P accepts an input value y then $P(y) = 1$, else $P(y) = 0$. Thus, the set of values accepted by the predicate P , is denoted as $P^{-1}(1)$. Logic AND, OR and XOR with two variables is denoted as $x \wedge y, x \vee y$ and $x \oplus y$, respectively. For integer k denote predicates $kOR, kAND$ and $kXOR$ as a logical OR, AND and XOR of the k variables, respectively. If $kXOR(x_1, \dots, x_k) = 1$, then (x_1, \dots, x_k) has odd parity, else even parity. Literal—a Boolean variable or its negation.

Definition 1. Suppose there is a predicate $P: \{-1,1\}^k \rightarrow \{0,1\}$. An instance of the problem *Max-CSP-P* consists of m weighted constraints, each of them is a k -tuple of literals $(z_{i_1}, \dots, z_{i_k})$ drawn from the set $\{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$. All variables in the tuple are different. Constraint is satisfied if and only if P accepts a tuple. The solution is the assignment of truth values to $\{x_1, \dots, x_n\}$. Value of the solution is $\sum_{i=1}^m w_i P(z_{i_1}, \dots, z_{i_k})$,

where w_i is (not negative) weight of i -th constraint. The goal is to maximize this value. When P depends on no more than k literal *Max-CSP-P* will be called *Max-kCSP-P*, if in P exactly k literals—then *Max-EkCSP-P*.

Along with the problems of the type *Max-CSP-P* discussed problems such as *CSP-P*, where the goal is to find such assignment, that all constraints are satisfied (*kCSP-P* and *EkCSP-P* similarly defined).

Definition 2. Two k -arity predicates P and P' have the same type if and only if there exists a permutation π on $[k] = \{1, \dots, k\}$ and $a \in \{-1,1\}^k$, such that $P(x_1, \dots, x_k) = P'(a_1 x_{\pi(1)}, \dots, a_k x_{\pi(k)})$ for all $x \in \{-1,1\}^k$.

If P and P' have the same type, then an instance *Max-CSP-P* can be expressed as an instance *Max-CSP-P'*, rearranging the tuples according to the mask, i.e., these problems are equivalent.

Definition 3. Problem *Max-kCSP-P*, where each constraint is disjunction of no more than k literals is a problem *Max-k-SAT*. If each constraint contains exactly k literals, that is the problem *Max-Ek-SAT*.

Definition 4. Problem *Max-kCSP-P*, where each constraint is a product of no more than k literals equal to a

constant, is the problem *Max-k-LIN*. If each constraint contains exactly k literals, that is the problem *Max-Ek-LIN*.

Let $w_{opt}(I)$ is the value of the optimal solution of instance I .

Definition 5. The algorithm A is C -approximation algorithm for the maximization problem if for all instances I of the problem $w(A,I) \geq \frac{1}{C} \cdot w_{opt}(I)$, where $w(A,I)$ —

the value of the solution of algorithm A for the input I . In this talk, that A has the approximation ratio C . For probabilistic algorithms $w(A,I)$ —the expected value of random elections of algorithm A .

The predicate P is approximation resistant (and the corresponding problem *Max-CSP-P*), if to find a solution *Max-CSP-P*, that is much better than expected value of random assignment, is *NP*-hard. Because of random assignment accept any P -constraint with probability $d(P) = 2^{-k} |P^{-1}(1)|$, we have the following definition.

Definition 6. The predicate $P: \{-1,1\}^k \rightarrow \{0,1\}$ is called approximation resistant if for any constant $\varepsilon > 0$ to find a solution x of instance I of the problem *Max-CSP-P* such, that the value x no more than

$$\left((d(P))^{-1} - \varepsilon \right)^{-1} w_{opt}(I), \text{ is } NP\text{-hard.}$$

Definition 7. The problem *Max-CSP-P* is always approximable, if for any $\delta > 0$ there exists $\varepsilon_\delta > 0$ and an efficient algorithm, which is based on an instance, where some $\left((d(P))^{-1} - \delta \right)^{-1}$ -part of constraints may be simultaneously accepted, find the assignment, that accept no more than $\left((d(P))^{-1} - \varepsilon_\delta \right)^{-1}$ -part of the constraints.

Definition 8. The predicate $P: \{-1,1\}^k \rightarrow \{0,1\}$ is called hereditary approximation resistant if all the predicates P' which are consequences of P (i.e. $(P(y) = 1) \Rightarrow (P'(y) = 1)$, for all y) are approximation resistant.

Theorem 1 [8]. The problem *Max-CSP-P* admits a polynomial approximation algorithm with approximation ratio $(d(P))^{-1}$.

Proof. Let us have an instance with m constraints. Random assignment accepted any given constraint with a probability $d(P)$ and, thus, accepted $d(P) \cdot m$ constraints on average. Since the optimal assignment accepted no more than m constraints, we have the random $(d(P))^{-1}$ -approximation algorithm. That random algorithm may be derandomized by the method of conditional expectation.

Remark 1. For approximation resistant predicates P threshold approximation ratio of the problem *Max-CSP-P* is attained (Theorem 1) and equals

$$q(P) = (d(P))^{-1} = 2^k \left(|P^{-1}(1)| \right)^{-1}.$$

This value is called the threshold “random” approximation ratio.

We have the following results on approximation resistant predicates of *Max-EkCSP-P* problems. There is no predicate of arity $2(k=2)$, which are approximation resistant [8]. If $k=3$ the problem *Max-E3-LIN* is hereditary approximation resistant [6], and it exhausts all of approximation resistant predicates of arity 3 [29]. Approximation resistant predicates of arity 4 ($k=4$) are studied in [28]. There are 400 different predicates (up to permutations of variables and their negations). Among them, 79 were identified as approximation resistant, 275-not as approximation resistant, the status of the remaining 46 predicates could not be determined.

Present the main results for approximation resistant predicates of arity 3 (*Max-E3CSP-P*).

Theorem 2 [6]. For an arbitrary $\varepsilon > 0$ it is NP-hard to approximate *Max-E3-LIN* with ratio $2 - \varepsilon$. In other words *Max-E3-LIN* is approximation resistant.

Theorem 3 [6]. For an arbitrary $\varepsilon > 0$ it is NP-hard to approximate *Max-E3-LIN* with ratio $\frac{8}{7} - \varepsilon$. In other words *Max-E3-LIN* is approximation resistant.

Theorem 4 [6]. Let P the predicate of arity 3 such, that $P(x, y, z) = 1$ for any x, y, z , satisfying the equation $xyz = 1$, then CSP, determined by P , is approximation resistant.

Theorem 4 remains true if we replace the equation $xyz = 1$ by $xyz = -1$. Generalization of Theorem 4 is the following theorem.

Theorem 5 [8]. Predicate P of arity 3 is approximation resistant, if and only if it is a consequence of the odd parity or even parity.

Consider the following predicates of arity 3:

$$\begin{aligned} XOR(x_1, x_2, x_3) &= x_1 \oplus x_2 \oplus x_3 \\ NTW(x_1, x_2, x_3) &= (\bar{x}_1 \vee \bar{x}_2) \bar{x}_3 \vee (x_1 \equiv x_2) x_3 \\ OXR(x_1, x_2, x_3) &= x_1 \vee (x_2 \oplus x_3) \\ OR(x_1, x_2, x_3) &= x_1 \vee x_2 \vee x_3. \end{aligned}$$

The above results can be summarized in the following assertion.

Theorem 6. Predicates *XOR*, *NTW*, *OXR*, *OR* are approximation resistant predicates. Among them *XOR*, *NTW*, *OXR*-hereditary approximation resistant predicates.

Following [30,31] we introduce a generalization of the CSP problem (GCSP problems), where instead of predicates with values from $\{0,1\}$, payoff functions with values from $[-1,1]$ to be used.

Definition 9 (Λ -GCSP problem). Λ -GCSP problem is defined as $\Lambda = ([q], P, k)$ where $[q] = \{0, 1, \dots, q-1\}$,

$P = \{P: [q]^t \rightarrow [-1, 1] \mid t \leq k\}$, a lot of payoff functions.

The maximum number of inputs of the payoff function $P \in P$ is the dimension of the problem Λ .

Definition 10. An instance Φ of the problem Λ -GCSP is defined as $\Phi = (V, P_V, W)$, where

- $V = \{y_1, \dots, y_m\}$: variables taking values from $[q]$;
- P_V consists of the payoff functions that are applied to subsets S of variables V of size no more than k . More precisely, for a subset $S = \{s_1, s_2, \dots, s_t\} \subset \{1, \dots, m\}^t$ payoff function $P_S \in P_V$ is applied to the variables $y_S = (y_{s_1}, \dots, y_{s_t})$;
- positive weights $W = \{w_S\}$ satisfy $\sum_{S \in \mathcal{W}, |S| \leq k} w_S = 1$, $S \in \mathcal{W}$ denote the set S , chosen according to probability distribution W .

The goal is to find the assignment of variables that maximize the expected total weight or total weight, i.e.

$$\text{maximize } E_{S \in \mathcal{W}} [P_S(y_S)] = \sum_{S \in [m], |S| \leq k} w_S P(y|S) \text{ (denote this}$$

maximum as $opt(\Phi)$).

Note, that if the payoff functions P are predicates, and the problem Λ -GCSP is unweighted ($w_S = 1$), then $opt(\Phi)$ will be just the maximum number of accepted constraints.

We introduce the predicate $XOR(x_1, x_2) = x_1 \oplus x_2$. In the future, as an example, we consider the problem Max Cut.

Definition 11 (Max Cut). For a given undirected graph $G = (V, E)$ with set vertices V and edges E of Max Cut is the problem of finding a partition $C = (V_1, V_2)$ of the vertices $V (V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset)$, that maximizes the size of the set $(V_1 \times V_2) \cap E$. For a given weight function $w: E \rightarrow R^+$, weighted Max Cut problem is to maximize $\sum_{e \in (V_1 \times V_2) \cap E} w(e)$.

Let us consider in more detail the problem Max Cut. For a graph $G = (V, E)$ with set vertices V and edges E this problem (the maximum cut in the graph) is defined as follows: find a partition V on V_1 and V_2 to maximize the number of edges, that form a cut, i.e. lies between the two parts. If each vertex i associate a Boolean variable $x_i (x_i = 1, i \in V_1, x_i = -1, i \in V_2)$, then the problem can be viewed as *Max-E2CSP-XOR* or *Max-E2-LIN* with the equations of the form $x_i x_j = -1$.

3. On Computational Complexity of Reoptimization

Consider an arbitrary unweighted *Max-EkCSP-P* problem Z (Definition 1). Let $V = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$ the set of variables, E -the set of constraints. The constraint $e \in E$ is denoted as $e = (x_{e_1}, \dots, x_{e_k}), e_i \in [2 \cdot n]$ with a special order of the variables (relative to V). The assignment is a map $\rho: V \rightarrow \{0, 1\}$, the assignment ρ accepted con-

straint e , if $P(\rho(x_{e_1}), \dots, \rho(x_{e_k})) = 1$. We denote by $OPT(I)$ the maximum part of the constraints accepted by any assigning an arbitrary instance I of the problem Z .

Let an instance I of the problem Z such, that $E = \{e^{(1)}, e^{(2)}, \dots, e^{(m)}\}$ - the set of m constraints. The constraint $e^{(j)} \in E$ is denoted as $e^{(j)} = (x_{e_1^{(j)}}, \dots, x_{e_k^{(j)}})$,

$e_i^{(j)} \in [2 \cdot n]$, $(1 \leq j \leq m, 1 \leq i \leq k)$ with a special order of the variables (relative to V). An instance I' of the problem is obtained from I by adding an arbitrary $(m+1)$ -th constraint $e^{(m+1)}$ (the same structure, as $e^{(j)}, 1 \leq j \leq m$). Define reoptimization version of the problem *Max-EkCSP-P*.

Problem *Ins-Max-EkCSP-P*. Input. Arbitrary instance I of the problem *Max-EkCSP-P*, x^* —the optimal solution of instance I .

Result. Find the optimal solution of instance I' (obtained on the basis of I , as described above) of the problem *Max-EkCSP-P*, using x^* .

Purpose. Find x , that maximizes the number of accepted constraints of instance I' .

Useful and interesting are challenges to establishing of *NP*-hardness of reoptimization versions of optimization problems. Using the results of [27] (in particular Theorem 2), we propose a criterion for determining of *NP*-hardness of reoptimization. The essence of the criterion for the most of *NP*-hard problems is that in order to show *NP*-hardness of reoptimization versions suggestions are based on polynomial Turing reducibility of the original problem to its reoptimization version.

Lemma 1. *Let P -NP-hard problem and mod- P -some local modification to P . If there exists a polynomial algorithm A , which for any instance I of the problem P computes:*

- 1) instance I' for mod- P ;
- 2) the optimal solution x' for I' ;
- 3) a sequence of local modifications of the type mod (no more than a polynomial), that transforms I' into I , then the problem mod- P is *NP*-hard.

Proof. Reduce P to mod- P using a polynomial Turing reducibility. Because of P is *NP*-hard, and that such (i.e., *NP*-hard) will be mod- P .

Let q —the number of local modifications of the type mod for A that I' is converted into an instance I . Suppose that there exists a polynomial algorithm A_1 (with complexity p) for mod- P . Then, using A_1 exactly q times since with I' , we find the optimal solution for I . At the same time as the number of calculations (q) and time of each calculation (p), polynomial in the size of P , we obtained polynomial reducibility (with the complexity $q \cdot p$). Lemma is proved.

Using Lemma 1, it is possible for specific predicates P from *NP*-hardness of *Max-EkCSP-P* (at $k \geq 2$) set *NP*-hardness of *Ins-Max-EkCSP-P*. For example, we show

how to do it for *Max-Ek-Lin*.

Theorem 7. *The problem *Ins-Max-Ek-Lin* is *NP*-hard.*

Proof. We use Lemma 1. As P we take a *NP*-hard problem *Max-Ek-Lin* [8], but as a mod- P -problem *Ins-Max-Ek-Lin*. Let I —an arbitrary instance of the problem *Max-Ek-Lin* (it corresponds to a system L of m linear equations). Let $x_{i_1} \oplus x_{i_2} \oplus \dots \oplus x_{i_k} = b$ -one of these equations (we take it as I'). Construct in polynomial time an assignment of values to vector $x = (x_1, \dots, x_n)$, which makes this equation acceptable. We assign the set of $\{x_1, \dots, x_n\}$ arbitrary values of truth. If $\{x_{i_1}, x_{i_2}, \dots, x_{i_k}\}$ satisfies equation I' then the build is completed, otherwise any value x_{i_l} ($l \in [k]$) is reversed, resulting in the equation I' will be accepted in polynomial time, i.e. point 1) and 2) of Lemma 1 are satisfied. As I' can be transformed into I no more than m modifications mod- P (i.e., add no more than m equations), point 3) of Lemma 1 is also fulfilled and the theorem is proved.

4. Reoptimization of Constraint Satisfaction Problems with Approximation Resistant Predicates

Theorem 8. *If $k = O(\log n)$ and for the problem *Max-EkCSP-P* exists a polynomial ρ -approximation algorithm, then for the problem *Ins-Max-EkCSP-P* (reoptimization *Max-EkCSP-P*) exists a polynomial $\psi(\rho)$ —approximation algorithm, where $\psi(\rho) = 2 - \frac{1}{\rho}$.*

Proof. We apply the approach discussed in [25,26]. Let I - an instance of the problem *Max-EkCSP-P*, which consists of a system of constraints $E = \{e^{(i)}, i \in [m]\}$ and optimal solution x^* , $w(x^*)$ - the number of accepted constraints in the system E by solution x^* . Adds a constraint $e^{(m+1)}$ to the system, the result is an instance I' of the problem *Max-EkCSP-P*, let $x_{I'}$ - the best solution of it. If $x_{I'}$ does not accepted constraint $e^{(m+1)}$, then x^* is the optimal solution of instance I' of the problem *Ins-Max-EkCSP-P*, then

$$w(x^*) \geq w(x_{I'}) - 1 \tag{1}$$

(on the left side write down the condition, that x^* - the best solution, and the right, that the optimal solution does not accepted constraint $e^{(m+1)}$). Suppose $x_{I'}$ is accepted the constraint $e^{(m+1)}$ and there are l ways in which the constraint is satisfied (obviously, $l < 2^k$).

We construct l approximate solutions x^i ($i \in [l]$) as follows. Take i -th assignment, which accepted $e^{(m+1)}$. From the constraint system we remove $e^{(m+1)}$ and for the constraints, that remain (including the result of assignment) use a polynomial ρ - approximation algorithm, we obtain an approximate solution x^i . The result is

$$w(x^i) \geq \frac{1}{\rho} (w(x_{I'}) - 1) + 1 = \frac{1}{\rho} w(x_{I'}) + 1 - \frac{1}{\rho} \tag{2}$$

Multiplying (1) on $1 - \frac{1}{\rho}$ and adding with (2) we obtain

$$\begin{aligned} & \left(1 - \frac{1}{\rho}\right)w(x^*) + w(x^i) \\ & \geq \left(1 - \frac{1}{\rho}\right)w(x_{r'}) - \left(1 - \frac{1}{\rho}\right) + \frac{1}{\rho}w(x_{r'}) + 1 - \frac{1}{\rho} \\ & = w(x_{r'}) \end{aligned}$$

Among the solutions x^* and x^i choose the best (i.e., with the largest value of the objective function w) and is denoted by \bar{x} . We have

$$\begin{aligned} w(x_{r'}) & \leq \left(1 - \frac{1}{\rho} + 1\right) \max\{w(x^*), w(x^i)\} \\ & = \left(2 - \frac{1}{\rho}\right)w(\bar{x}), \end{aligned}$$

and $w(\bar{x}) \geq \frac{1}{2 - 1/\rho}w(x_{r'})$. To the algorithm is polynomial was sufficient to require that $2^k \leq n^c$ (n -the total number of variables, $c = \text{const}$), which means $k = O(\log n)$ in the theorem. Thus, as a result of this algorithm, an approximate solution \bar{x} of the instance I' with approximation ratio $\psi(\rho) = 2 - \frac{1}{\rho}$ is obtained. It is clear that at all times $2 - \frac{1}{\rho} < \rho$ ($\rho \neq 1$).

Theorem 9. *If for a problem Max-EkCSP-P exists a polynomial threshold (optimal) ρ -approximation algorithm, and for the problem Ins-Max-EkCSP-P (reoptimization Max-EkCSP-P), there exists a polynomial γ -approximation algorithm, then $\gamma \geq \psi(\rho)$.*

Proof. Let I - an instance of the problem Max-EkCSP-P, which consists of a system of constraints $E = \{e^{(i)}, i \in [m]\}$ and optimal solution x^* . Adds a constraint $e^{(m+1)}$ to the system, the result is an instance I' of the problem Ins-Max-EkCSP-P. Let \bar{x} - the solution of Ins-Max-EkCSP-P, obtained by the algorithm of Theorem 6. The solution \bar{x} is the best (more on the value of the objective function) of the solutions x^* , and x^i ($i \in [l], l < 2^k$), it is obtained by a polynomial approximation algorithm with approximation ratio $\psi(\rho) = 2 - \frac{1}{\rho}$. The proof is by contradiction. Let $\gamma < \psi(\rho)$ and ρ^* - such, that $\psi(\rho^*) = \gamma$. Since the function $\psi(\rho)$ is increasing in ρ and $\psi(\rho^*) = \gamma < \psi(\rho)$, it follows, that $\rho^* < \rho$. But this contradicts the fact, that for the problem Max-EkCSP-P exists a polynomial threshold (optimal) ρ -approximation algorithm (i.e., for solutions x^i to be applied poly-

nomial-time algorithm with approximation ratio ρ^* , less than ρ , that is impossible).

Theorem 10. *If for a problem Max-EkCSP-P exists a polynomial threshold (optimal) ρ -approximation algorithm and $k = O(\log n)$, then for the problem Ins-Max-EkCSP-P (reoptimization Max-EkCSP-P), there exists a polynomial threshold (optimal) $\psi(\rho)$ -approximation algorithm, where $\psi(\rho) = 2 - \frac{1}{\rho}$.*

The proof follows from Theorems 8 and 9

Corollary 1. *If $k = O(\log n)$ and the predicate P approximation resistant, then for the problem Ins-Max-EkCSP-P (reoptimization of Max-EkCSP-P), there exists a polynomial optimal $r(P)$ -approximation algorithm, where $r(P) = \frac{2^{k+1} - |P^{-1}(1)|}{2^k}$.*

Proof. Since the predicate P is approximation resistant, according to Remark 1, the algorithm of Theorem 1 for Max-EkCSP-P is optimal $q(P)$ -approximation algorithm, where $q(P) = (d(P))^{-1}$, $d(P) = 2^{-k} |P^{-1}(1)|$. Hence, by Theorem 10 for Ins-Max-EkCSP-P there exists a polynomial threshold (optimal) $\psi(q(P))$ -approximation algorithm, where

$$\psi(q(P)) = 2 - \frac{1}{q(P)} = 2 - d(P) = 2 - 2^{-k} |P^{-1}(1)|.$$

Example 1. Consider the problem Max-E3CSP-XOR with the appropriate reoptimization version Ins-Max-E3CSP-XOR. By Theorem 6, the predicate XOR- is hereditary approximation resistant (there is proof of this fact in [28]). We apply Theorem 10 (or more precisely, Corollary 1) ($k = 3, |P^{-1}(1)| = 4$) and obtain proposition.

Proposition 1. *For the problem Ins-Max-E3CSP-XOR (reoptimization of Max-E3CSP-XOR) there exists a polynomial optimal approximation algorithm with an approximation ratio $\frac{3}{2}$.*

5. Integrality Gaps of Semidefinite Relaxation

For each instance $\Phi = (V, P_V, W)$ (Definition 10) is constructed semidefinite (SDP) relaxation [30] (which is not presented here). Let $sdp(\Phi)$ -solution of SDP relaxation (clearly, $sdp(\Phi) \geq opt(\Phi)$). We introduce the notion of integrality gap for semidefinite relaxation of the constraint satisfaction problems.

Definition 12. Integrality gap of Λ -GCSP problem is defined as $\alpha_\Lambda = \sup_{\Phi \in \Lambda} \left\{ \frac{sdp(\Phi)}{opt(\Phi)} \right\}$

The notion of integrality gap for some relaxation (not

only semidefinite) is important, because it allows to design approximation algorithms for solving discrete optimization problems with a given approximation ratio. The following theorem holds.

Theorem 11 [31]. *For the problem Λ -GCSP with non-negative payoff functions there exists a polynomial approximation algorithm with approximation ratio no more than integrality gap α_Λ .*

This theorem can comment on such arguments. First, we solve the problem SDP_{gen} (general SDP relaxation of the problem Λ -GCSP), let Φ_{gen} the solution of it. Applying some probabilistic scheme of rounding, from solution Φ_{gen} we obtain an approximate solution Φ_{appr} of the original problem Λ . By Definition 12 we obtain $w(\Phi_{gen}) \leq \alpha_\Lambda \cdot w(\Phi_{appr})$ (where $w(\cdot)$ denotes the weight of the solution), then

$$w(\Phi_{appr}) \geq \frac{1}{\alpha_\Lambda} \cdot w(\Phi_{gen}) \geq \frac{1}{\alpha_\Lambda} \cdot opt(\Phi)$$

and, by definition 5, we received an α_Λ —approximation algorithm.

Note, that the calculation (estimation) of integrality gaps of relaxations is in itself a difficult research task. For many problems it is still unsolvable. However, even without knowing the specific values of the integrality gaps of relaxations, one can argue about the existence of threshold (optimal) approximation algorithms for optimization problems (which will be noted later).

To illustrate, consider the Max Cut problem. Let v_i -a unit vector in Euclidean space, which corresponds to a Boolean variable x_i . We have the following SDP relaxation of Max Cut:

$$\max \left\{ \frac{1}{|E|} \sum_{(i,j) \in E} \frac{1 - v_i \cdot v_j}{2} \right\}, \forall i \in [n], \|v_i\| = 1,$$

where $v_i \cdot v_j$ - the scalar product v_i and v_j . We define an integrality gap α_{MC} of this relaxation:

$\alpha_{MC} = \sup_G \left\{ \frac{SDP(G)}{OPT(G)} \right\}$, where $SDP(G)$ —the optimum of relaxation.

Theorem 12 [5].

$$\begin{aligned} \alpha_{MC} &\leq \max_{\theta \in [0, \pi]} \left\{ \frac{\pi \cdot 1 - \cos \theta}{2} \right\} \\ &= \frac{\pi}{2} \cdot \frac{1 - \cos \theta_c}{\theta_c} \approx 1.138, \end{aligned}$$

where θ_c is the “critical angle” at which the maximum is attained.

Goemans and Williamson give random rounding algorithm (now known as the random hyperplane rounding algorithm) that for any solution of SDP relaxation find a cut in a graph with a value no less, than $1/\alpha_{MC}$ times

once expected SDP solution (note that $\alpha_{GW} = 1/\alpha_{MC}$, where $\alpha_{GW} \approx 0.878567$ a known Goemans-Williamson constant). Thus, the approximation algorithm not only finds an approximate optimum value, but also gives an approximate cut. This feature is characteristic of most algorithms based on SDP and LP (linear) relaxation.

Theorem 13 [10]. *For any $\varepsilon > 0$ there exists a graph*

$G(V, E)$ such, that $\frac{SDP(G)}{OPT(G)} \geq \frac{\pi}{2} \cdot \frac{1 - \cos \theta_c}{\theta_c} - \varepsilon$. Thus

$$\alpha_{MC} = \sup_G \left\{ \frac{SDP(G)}{OPT(G)} \right\} \geq \frac{\pi}{2} \cdot \frac{1 - \cos \theta_c}{\theta_c}, \text{ combining with}$$

Theorem 12, we obtain $\alpha_{MC} = \frac{\pi}{2} \cdot \frac{1 - \cos \theta_c}{\theta_c}$.

A lower bound for integrality gap is a graph $G(V, E)$, where the bound is attained. Corresponding instance of the problem is Integrality Gap Instance (IGI).

So, for Max Cut managed to find the exact value of the integrality gap of SDP relaxation.

6. Unique Games Conjecture and Reoptimization

Unique Games Conjecture (UGC) was introduced by Khot [11] as a possible way to obtain new results on strong innapproximability. We formulate the UGC in terms of Unique Game Problem.

Definition 13. A Unique Game Problem is a constraint satisfaction problem, which is defined as follows. There is a directed graph $G(V, E)$, whose vertices represent variables and edges-constraints. The purpose is to assigning a label to each vertex from the set $[n]$. Constraint on the edge $e = (v, w) \in E$ is described by a bijection $\pi_e : [n] \rightarrow [n]$. Labeling the vertices $L : V \rightarrow [n]$ satisfies (accepts) a constraint on the edge $e = (v, w)$, iff $\pi_e(L(v)) = L(w)$. Let $OPT(U)$ denote the maximum part of constraints, which may be satisfied by any labeling:

$$OPT(U) = \max_{L:V \rightarrow [n]} \left\{ \frac{1}{|E|} \cdot |\{e \in E \mid L \text{ satisfied } e\}| \right\}.$$

Unique Games Conjecture (UGC) [11]. For any $\varepsilon, \delta > 0$ there exists a constant $n = n(\varepsilon, \delta)$ such that, for this instance of unique game problem $U(G(V, E), [n], \{\pi_e \mid e \in E\})$ is NP-hard to distinguish between two cases:

- YES case: $OPT(U) \geq 1 - \varepsilon$
- NO case: $OPT(U) \leq \delta$.

A typical technique to obtain the results on innapproximability can be described as follows. The source is the following argument. Suppose P - an arbitrary optimization (to be specific to a maximum of) problem. Under the (c.s)-gap version of the problem P (notation

$Gap-P_{c,s}$) we understand the problem, for which either $OPT(I) \geq c$, or $OPT(I) \leq s$ for any instance $I \in P$. Consider the NP-complete problem 3-Sat (3-Satisfiability). Arbitrary 3-Sat formula (E3-CNF formula) is the conjunction of a set of clauses, where each clause is the disjunction of three Boolean variables or their negations. The goal is to determine the assignment of a Boolean variable, such that the formula is logically true (acceptable). Suppose that there exists a polynomial reducibility of 3-Sat to $Gap-P_{c,s}$ for some $0 < s < c$, that is, reducibility, which displays a 3-Sat formula ψ to an instance I of the problem P such that:

(YES case): If ψ has an assignment, that makes it acceptable, then $OPT(I) \geq c$;

(NO case): If ψ has no assignments, that make it acceptable, then $OPT(I) \leq s$.

This reducibility implies that if there exists a polynomial algorithm with approximation ratio strictly less than $\frac{c}{s}$ for the problem P , then it is possible to efficiently determine whether a 3SAT formula is satisfiable, and hence $P = NP$. Thus, under the standard assumption, that $P \neq NP$ this reducibility—the source of results on inapproximability of the problem P . We start from the PCP (Prababilistically Checkable Proof) Theorem [1] in one form or another for some NP-complete language (for example, 3-Sat). We construct a reducibility to the problem (language), which inapproximability to install (for example, $Gap-P_{c,s}$). Constructed PCP verifier for the problem P , which is in the form of a test (dictatorship) to the Boolean function that is responsible to P . Using the elements and some of the results of Fourier analysis of Boolean functions, estimated completeness c of the verifier (the lower bound of the probability of accepting the test, that a Boolean function-dictatorship or YES case) and soundness s of verifier (the upper bound of the probability of not accepting the test, that Boolean function is far from dictatorship or NO case). It follows that P NP-hard to approximate with a ratio smaller than c/s . This is a common inapproximability.

If not proceed from the PCP theorem, but from the unique game conjecture (UGC) in the above reducibility, we receive inapproximability based on UGC or conditional inapproximability.

Let $sdp(\Phi)$ the solution of SDP relaxation of instance Φ of GCSP problem Λ . In [30] to get closer to the optimal solution proposed scheme of rounding (Rounding Scheme, RS). In this paper, studies are being conducted in the language of integrality gap curve and unique games hardness curve. We describe the result with an integrality gap coefficient α_Λ . We will assume that $k = \text{const}$.

Theorem 14 [30]. *Assuming the UGC, for any GCSP problem Λ and any $\eta > 0$ it is NP-hard to approximate*

Λ with approximation ratio $\alpha_\Lambda - \eta$.

Using Theorems 11 and 14, we get a result.

Corollary 2 [30]. *Assuming the UGC for any GCSP and any $\eta > 0$ rounding scheme RS determines the approximation ratio in the range η of optimal polynomial algorithm, i.e. for any GCSP problem Λ there exists a polynomial threshold (optimal) α_Λ -approximation algorithm.*

Consider an arbitrary unweighted Max-EkCSP-P problem Z (definition 1). Let $V = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$ the set of variables, E —the set of constraints. The constraint $e \in E$ is denoted as $e = (x_{e_1}, \dots, x_{e_k})$, $e_i \in [2 \cdot n]$ with special order on the variables (relative to V). Assignment is a map $\rho: V \rightarrow \{0, 1\}$, assignment ρ accepts constraint e , if. $P(\rho(x_{e_1}), \dots, \rho(x_{e_k})) = 1$. We denote by $OPT(I)$ the maximum part of constraints accepted by an arbitrary assigning for instance I of the problem Z . Let $SDP(I)$ denote the optimum SDP relaxation of Raghavendra [30], we define an integrality gap

$$\alpha_Z = \sup_{I \in Z} \left\{ \frac{SDP(I)}{OPT(I)} \right\}.$$

In [31] showed how to round a solution and find assignment with the approximation ratio close to α_Z (theorem 11). Let $\alpha_Z = \frac{c}{s}$, then the result of

Raghavendra [30] in this case can be presented as a theorem.

Theorem 15 [14]. *Suppose there is an instance I^* of Max-EkCSP-P problem Z such, that $SDP(I^*) \geq c$ and $OPT(I^*) \leq s$. Then for any $\gamma > 0$ there exist $\varepsilon, \delta > 0$ and polynomial reducibility from the instance of unique game problem to the instance I of problem Z such, that:*

- (YES case): If $OPT(U) \geq 1 - \varepsilon$, then $OPT(I) \geq c - \gamma$;
- (NO case): If $OPT(U) \leq \delta$, then $OPT(I) \leq s + \gamma$.

In particular, assuming the UGC, it is NP-hard to approximate Z with ratio strictly less than α_Z .

Corollary 3. *Assuming the UGC, for every Max-EkCSP-P problem Z there exists a polynomial threshold (optimal) α_Z —approximation algorithm.*

The proof follows from theorems 11, 14 and 15.

Note that theorem 15 converts the integrality gap into inapproximability gap. Roughly speaking the idea is to use integrality gap instance (IGI) of SDP relaxation for the construction of a dictatorship test and combining it with an instance of unique game problem. The value of the result of Raghavendra is that even without knowing explicitly the exact value of integrality gap, you can set an optimality of corresponding polynomial approximation algorithm (using IGI). For example, in [32] showed that although for Grothendieck's Problem integrality gap K_G (the famous Grothendieck's constant) are still unknown, based on the UGC it is NP-hard to approximate the problem of Grothendieck with an arbitrary ratio less than K_G (K_G -approximation algorithm is optimal). Constant

K_G can be calculated with some error η in the time dependent only on η .

Theorem 16. Suppose that a unique game conjecture (UGC) is hold. Let Z is any unweighted Max-EkCSP-P

problem with integrality gap $\alpha_Z = \sup_{I \in Z} \left\{ \frac{SDP(I)}{OPT(I)} \right\}$ and

$k = \text{const}$, then for the problem Ins-Max-EkCSP-P (reoptimization of Max-EkCSP-P) there exists a polynomial threshold (optimal) $\psi(\alpha_Z)$ -approximation algorithm,

where $\psi(\alpha_Z) = 2 - \frac{1}{\alpha_Z}$.

The proof follows by applying corollary 3 to theorem 10.

Example 2. Consider the problem Max Cut. In our notation, this is a problem Max-E2CSP-XOR, and reoptimization version—the problem Ins-Max-E2CSP-XOR, obtained by adding an arbitrary edge to Max Cut. By theorems 12 and 13 integrality gap of SDP relaxation of problem Max Cut is $\alpha_{MC} = \frac{\pi}{2} \cdot \frac{1 - \cos \theta_c}{\theta_c} \approx 1.138$. Then

from theorem 16 follows proposition 2.

Proposition 2. Suppose that a unique games conjecture (UGC) is hold. Then for the problem Ins-Max-E2CSP-XOR (reoptimization of Max Cut) there exists a polynomial threshold (optimal) $\psi(\alpha_{MC})$ -approximation algorithm, where $\psi(\alpha_{MC}) = 2 - \frac{1}{\alpha_{MC}} \approx 1.121$.

7. Conclusions

To solve the problem Ins-Max-EkCSP-P (reoptimization of Max-EkCSP-P) with approximation resistant predicates there exists a polynomial threshold (optimal) $\psi(q(P))$ -approximation algorithm, where

$$\psi(q(P)) = 2 - \frac{1}{q(P)} = 2 - d(P) = 2 - 2^{-k} |P^{-1}(1)|$$

($d(P)$)—the threshold “random” approximation ratio of P). Note that the property of the approximation resistance may be stronger than the property of NP-completeness of corresponding decision problem. Since in this case the effective computation can not gives anything more than a random assignment of truth values to variables.

Using the results of [30,31] it can be argued, that if the unique games conjecture (UGC) is hold, then there exists a polynomial threshold (optimal) $\psi(\alpha_Z)$ —approximation algorithm to solve the problem Ins-Max-EkCSP-P (reoptimization of Max-EkCSP-P), where α_Z —the integrality gap of SDP relaxation of Max-EkCSP-P problem.

The results of this work greatly depend on the truth of the unique game conjecture (UGC). Note that if U - the unique games problem, the UGC can be formulated as

follows: $(1 - \varepsilon, \delta)$ —gap version of the unique games problem $U(\text{Gap-}U_{1-\varepsilon, \delta})$ is NP-hard problem. Along with the problems of complexity class relationships with respect to inclusion (for example, $P \neq NP?$) it is one of the major open problems of modern theoretical computer science. Even if the UGC is false, you may find that $\text{Gap-}U_{1-\varepsilon, \delta}$ is hard in the sense of undecidability in polynomial time, and such (a weak) hardness can be applied to all problems, where the hardness show up on the basis of UGC.

REFERENCES

- [1] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, “Proof Verification and Intractability of Approximation Problems,” *Journal of the ACM*, Vol. 45, No. 3, 1998, pp. 501-555. [doi:10.1145/278298.278306](https://doi.org/10.1145/278298.278306)
- [2] O. Goldreich, S. Goldwasser and D. Ron, “Property Testing and Its Connection to Learning and Approximation Abstract,” *Journal of the ACM*, Vol. 45, No. 4, 1998, pp. 653-750. [doi:10.1145/285055.285060](https://doi.org/10.1145/285055.285060)
- [3] O. Goldreich and M. Sudan, “Locally Testable Codes and PCPs of Almost-Linear Length,” *Journal of the ACM*, Vol. 53, No. 4, 2006, pp. 558-655. [doi:10.1145/1162349.1162351](https://doi.org/10.1145/1162349.1162351)
- [4] M. X. Goemans and D. P. Williamson, “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming,” *Journal of the ACM*, Vol. 42, No. 6, 1995, pp. 1115-1145. [doi:10.1145/227683.227684](https://doi.org/10.1145/227683.227684)
- [5] M. X. Goemans and D. P. Williamson, “0.878 Approximation Algorithms for MAX-CUT and MAX-2SAT,” In: *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC)*, ACM, Montreal, 1994, pp. 422-431.
- [6] J. Hastad, “Some Optimal Inapproximability Results,” *Journal of the ACM*, Vol. 48, No. 4, 2001, pp. 798-859. [doi:10.1145/502090.502098](https://doi.org/10.1145/502090.502098)
- [7] J. Hastad, “Complexity Theory, Proofs and Approximation,” European Congress of Mathematics, Stockholm, 2005.
- [8] J. Hastad, “On the Efficient Approximability of Constraint Satisfaction Problems,” In: A. Hilton and J. Talbot, Eds., *Surveys in Combinatorics 2007, London Mathematical Society Lecture Notes Series* (No. 346), Cambridge University Press, Cambridge, 2007, pp. 201-222. [doi:10.1017/CBO9780511666209.008](https://doi.org/10.1017/CBO9780511666209.008)
- [9] U. Feige, “A Threshold of $\ln n$ for Approximating Set Cover,” *Journal of the ACM*, Vol. 45, No. 4, 1998, pp. 634-652. [doi:10.1145/285055.285059](https://doi.org/10.1145/285055.285059)
- [10] U. Feige and G. Schechtman, “On the Integrality Ratio of Semidefinite Relaxations of MAX CUT,” In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, ACM, New York, 2001, pp. 433-442.
- [11] S. Khot, “On the Power of Unique 2-Prover 1-Round Games,” In: *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2002,

- pp. 767-775.
- [12] S. Khot and O. Regev, "Vertex Cover Might be Hard to Approximate to within $2 - \epsilon$," *Journal of Computer and System Sciences*, Vol. 74, No. 3, 2008, pp. 335-349. [doi:10.1016/j.jcss.2007.06.019](https://doi.org/10.1016/j.jcss.2007.06.019)
- [13] S. Khot, G. Klendler, E. Mossel and R. O'Donnell, "Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs?" *Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Rome, 17-19 October 2004, pp. 146-154. [doi:10.1109/FOCS.2004.49](https://doi.org/10.1109/FOCS.2004.49)
- [14] S. Khot, "On the Unique Games Conjecture," *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, Cambridge, 9-12 June 2010, pp. 99-121.
- [15] A. Samorodnitsky and L. Trevisan, "Gowers Uniformity, Influence of Variables, and PCPs," In: *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2006, pp. 11-20.
- [16] S. Chawla, R. Krauhgamer, R. Kumar, Y. Rabani and D. Sivakumar, "On the Hardness of Approximating Multicut and Sparsest-Cut," *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, San Jose, 11-15 June 2005, pp. 144-153. [doi:10.1109/CCC.2005.20](https://doi.org/10.1109/CCC.2005.20)
- [17] P. Austrin, "Balanced Max 2-Sat Might Not be the Hardest," *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, San Diego, 11-13 June 2007, pp. 189-197.
- [18] P. Austrin, "Towards Sharp Inapproximability for Any 2-CSP," In: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington DC, 2007, pp. 307-317.
- [19] M. Lewin, D. Livnat and U. Zwick, "Improved Rounding Techniques for the MAX 2-SAT and MAX DI-CUT Problems," *Proceedings of 9th International Integer Programming and Combinatorial Optimization Conference (IPCO), Lecture Notes in Computer Science*, Vol. 2337, Cambridge, 27-29 May 2002, pp. 67-82. [doi:10.1007/3-540-47867-1_6](https://doi.org/10.1007/3-540-47867-1_6)
- [20] G. Ausiello, B. Escoffier, J. Monnot and V. Th. Paschos, "Reoptimization of Minimum and Maximum Traveling Salesman's Tours," *Journal of Discrete Algorithms*, Vol. 7, No. 4, 2009, pp. 453-463. [doi:10.1007/11785293_20](https://doi.org/10.1007/11785293_20)
- [21] H. J. Bockenhauer, L. Forlizzi, J. Hromkovic, et al., "On the Approximability of TSP on Local Modifications of Optimal Solved Instances," *Algorithmic Operations Research*, Vol. 2, No. 2, 2007, pp. 83-93.
- [22] H.-J. Bockenhauer, J. Hromkovic, T. Momke and P. Widmayer, "On the Hardness of Reoptimization," In: V. Geffert et al., Eds., *SOFSEM, Lecture Notes in Computer Science*, Vol. 4910, Springer, Berlin, 2008, pp. 50-65.
- [23] B. Escoffier, M. Milanic and V. Th. Paschos, "Simple and fast Reoptimizations for the Steiner Tree Problem," *Algorithmic Operations Research*, Vol. 4, No. 2, 2009, pp. 86-94.
- [24] C. Archetti, L. Bertazzi and M. G. Speranza, "Reoptimizing the Travelling Salesman Problem," *Networks*, Vol. 42, No. 3, 2003, pp. 154-159. [doi:10.1002/net.10091](https://doi.org/10.1002/net.10091)
- [25] G. Ausiello, V. Bonifacci and B. Escoffier, "Complexity and Approximation in Reoptimization," In: S. B. Cooper and A. Sorbi, Eds., *Computability in Context: Computation and Logic in the Real World*, Imperial College Press, London, 2011, pp. 101-130.
- [26] V. A. Mikhailyuk, "Reoptimization of Set Covering Problems," *Cybernetics and Systems Analysis*, Vol. 46, No. 6, 2010, pp. 879-883. [doi:10.1007/s10559-010-9269-z](https://doi.org/10.1007/s10559-010-9269-z)
- [27] V. A. Mikhailyuk, "General Approach to Estimating the Complexity of Postoptimality Analysis for Discrete Optimization Problems," *Cybernetics and Systems Analysis*, Vol. 46, No. 2, 2010, pp. 290-295. [doi:10.1007/s10559-010-9206-1](https://doi.org/10.1007/s10559-010-9206-1)
- [28] G. Hast, "Beating a Random Assignment," Doctoral Thesis, Royal Institute of Technology, Stockholm, 2005.
- [29] U. Zwick, "Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables per Constraint," In: *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, 1998, pp. 551-560.
- [30] P. Raghavendra, "Optimal Algorithms and Inapproximability Results for Every CSP?" In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2008, pp. 245-254.
- [31] P. Raghavendra and D. Steurer, "How to Round Any CSP?" In: *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington DC, 2009, pp. 586-594.
- [32] P. Raghavendra and D. Steurer, "Towards Computing the Grothendieck Constant," *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, 2009, pp. 525-534.