Fixed-Parameter Tractability of Almost CSP Problem with Decisive Relations

Chihao Zhang and Hongyang Zhang

BASICS, Department of Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, China.

{chihao.zhang, hongyang90}@gmail.com

Abstract. Let I be an instance of binary boolean CSP. Consider the problem of deciding whether one can remove at most k constraints of I such that the remaining constraints are satisfiable. We call it the Almost CSP problem. This problem is **NP**-complete and we study it from the point of view of parameterized complexity where k is the parameter. Two special cases have been studied: when the constraints are inequality relations (Guo et al., WADS 2005) and when the constraints are OR type relations (Razgon and O'Sullivan, ICALP 2008). Both cases are shown to be fixed-parameter tractable (FPT). In this paper, we define a class of *decisive* relations and show that when all the relations are in this class, the problem is also fixed-parameter tractable. Note that the inequality relation is decisive, thus our result generalizes the result of the parameterized edge-bipartization problem (Guo et al., WADS 2005). Moreover as a simple corollary, if the set of relations contains no OR type relations, then the problem remains fixed-parameter tractable. However, it is still open whether OR type relations and other relations can be combined together while the fixed-parameter tractability still holds.

1 Introduction

Consider the following parameterized problem:

p-Almost-CSP	
Input:	An instance of binary boolean CSP, and a nonnegative integer k.
Parameter:	k.
Problem:	Decide whether one can delete at most k constraints such that the remaining constraints are satisfiable.

Many natural problems can be expressed in this setting. For example, p-Almost 2SAT problem[13], which asks whether a CNF formula φ can be satisfied if we are allowed to remove at most k clauses, is a special case of p-Almost CSP. It was noticed in [15] that the fixed-parameter tractability of p-Almost 2SAT is equivalent to the vertex cover problem parameterized above the perfect matching. Another special case which has received extensive attention in the literature is the parameterized edge-bipartization problem[6, 15], which asks whether one can remove at most k edges in an undirected graph such that the remaining graph is bipartite. Both problems have been shown to be fixed-parameter tractable.

The above special cases impose restriction on the type of relations. These results motivate us to explore the parameterized complexity of p-Almost-CSP under various other sets of relations. Let $\mathcal{R}$ be a set of relations. Let p-$\mathcal{R}$-Almost-CSP be the problem of p-Almost-CSP such that all the input CSP instance can only use relations in $\mathcal{R}$. Almost 2SAT is equivalent to the case that the constraints are restricted to OR type relations, which include $R_1(x, y) :=$ "$x \vee y$", $R_2(x, y) :=$ "$x \vee \bar{y}$", $R_3(x, y) :=$ "$\bar{x} \vee y$", $R_4(x, y) :=$ "$\bar{x} \vee \bar{y}$" (we denote the set of these four OR type relations by $\mathcal{R}_{or}$). Edge-bipartization problem corresponds to the case that $\mathcal{R}$ contains only inequality relation.

Our Results We define a class of *decisive* relations. A binary relation R is decisive if for $x \in \{0, 1\}$, at most one of $(x, 0)$ and $(x, 1)$ is in R and at most one of $(0, x)$ and $(1, x)$ is in R. Intuitively, if we fixed one component of a pair (x, y) where $x, y \in \{0, 1\}$, there is at most one choice for the other component to make the pair in R. We denote the set of decisive relations by $\mathcal{R}_{decisive}$. Decisive relations are quite expressive, including AND type relations, equality relation and inequality relation.

We present an $O^*(4^{k^2})$ ($O^*(\cdot)$ suppresses the polynomial term) algorithm for p-$\mathcal{R}_{decisive}$-Almost-CSP, hence it is fixed-parameter tractable.

Interestingly, based on the algorithms for decisive relations, it easily follows that if $\mathcal{R}$ contains no OR type relations, then p-$\mathcal{R}$-Almost-CSP is fixed-parameter tractable.

Our approach is based on the technique of iterative compression, which was first introduced in [14] to deal with the odd cycle transversal problem. Following the standard routine of this technique, we reduce p-$\mathcal{R}_{decisive}$-Almost-CSP to a variant edge-separation problem on graphs, we call this problem p-MinMixedCut. We then show that p-MinMixedCut is fixed-parameter tractable. The most important ingredient of our algorithm is the edge version of *important separator* introduced in [10].

Related Work The question of whether the Almost 2SAT problem is fixed-parameter tractable, as mentioned above, was regarded as a long standing open problem [9, 12, 3], and finally solved by Razgon and O'Sullivan[13]. For the parameterized edge-bipartization problem, a reduction to odd cycle transversal was first noticed in [15]. Guo et al. presented a better FPT algorithm in [6]. It is also shown in [8] that there is a parameterized reduction from the edge-bipartization problem to the Almost 2-SAT problem. All these algorithms rely on the framework of iterative compression, which was introduced in [14]. See [7] for a survey of this technique.

Important separator was first introduced in [10] but implicitly used in [2, 1, 13]. It has been widely used in designing algorithms for graph separation problems. See [11] for a gentle introduction to this concept.

This paper is organized as follows: In Section 2, we present the statement of the problem, give some necessary definitions and introduce the notations. In Section 3, we use iterative compression technique to reduce the problem to p-MinMixedCut and then present a $O^*(4^{k^2})$ algorithm to solve it in Section 4. In Section 5, we give an algorithm based on previous sections to prove the main theorem and evaluate its running time. Finally, we conclude in Section 6 with some open problems.

2 Preliminaries

2.1 Parameterized problems and fixed-parameter tractability

A parameterized problem is a pair (Q, κ), where $Q \subseteq \Sigma^*$ is a classic decision problems and $\kappa : \Sigma^* \to \mathbb{N}$ is a polynomial-time computable function. An instance of (Q, κ) is denoted by (x, k) where $k = \kappa(x)$. A fixed-parameter tractable (FPT) algorithm decides whether $x \in Q$ in time $O(f(k) \cdot |x|^c)$, where c is a constant and f is an arbitrary computable function that only depends on k. We may use $O^*(f(k))$ to suppress the polynomial term. The notion of FPT relaxes the polynomial-time tractability in the classic setting. Readers may refer to [4, 5, 12] for more information on parameterized complexity and algorithms.

2.2 Constraint Satisfaction Problem

An instance of Constraint Satisfaction Problem (CSP) is defined as a triple $I := (X, D, \mathcal{C})$ where X is a set of variables, D is a domain of values, and $\mathcal{C}$ is a set of constraints. Every constraint is a pair $\langle t, R\rangle$, where t is a c-tuple of variables and R is a c-ary relation on D. An evaluation of the variables is a function from the set of variables to the domain of values $v : X \to D$. An evaluation v satisfies a constraint $\langle (x_1, \ldots, x_c), R\rangle$ if $(v(x_1), \ldots, v(x_c)) \in R$. A solution is an evaluation that satisfies all constraints. An instance I is satisfiable if it has a solution.

In this paper, we only consider *binary boolean* CSP, namely $D = \{0, 1\}$ and $c \leq 2$ for all relations R.

To explain why we focus on binary boolean case, note that the decision version of CSP remains NP-hard when $|D| \geq 3$ and $c = 2$, or when $|D| = 2$ and $c \geq 3$, therefore in both cases p-Almost-CSP is not fixed-parameter tractable unless **PTIME** = **NP**.

2.3 Binary boolean relations

There are 16 different binary boolean relations in total, listed in Table 1.

We divide the relations into three categories, namely $\mathcal{R}_{or}, \mathcal{R}_{decisive}, \mathcal{R}_{other}$, as shown in the table. Let $\mathcal{R}_{decisive} := \{R_5, \ldots, R_{11}\}$, a binary boolean relation R is decisive if $R \in \mathcal{R}_{decisive}$. This set of relations can be defined as follows in a more intuitive way:

	$\mathcal{R}_{or}$				$\mathcal{R}_{decisive}$							$\mathcal{R}_{other}$				
	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}
(0,0)	0	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0
(0,1)	1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	0
(1,0)	1	1	0	1	0	1	0	0	1	0	0	1	0	1	0	1
(1,1)	1	1	1	0	1	0	0	0	0	1	0	1	0	0	1	1

Table 1. 16 binary boolean relations

Definition 1 (Decisive Relation). *Let R be a binary boolean relation. We say R is* decisive *if for every $u \in \{0,1\}$, $\neg(R(u,0) \wedge R(u,1))$ and $\neg(R(0,u) \wedge R(1,u))$.*

Intuitively, if we fix one component of the relation, there is at most one choice for the other component such that the pair is in R.

Decisive relations have very simple interpretations: $R_5(x,y) :=$ "$x = y$", $R_6(x,y) :=$ "$x \neq y$", $R_7 :=$ "$\bar{x} \wedge \bar{y}$", $R_8 :=$ "$\bar{x} \wedge y$", $R_9 :=$ "$x \wedge \bar{y}$", $R_{10}(x,y) :=$ "$x \wedge y$", $R_{11} := \varnothing$. Let $R_{and} := \{R_7, R_8, R_9, R_{10}\}$.

2.4 Problem statement and main result

Let $\mathcal{R}$ be a set of relations, consider the problem

p-$\mathcal{R}$-Almost-CSP	
Input:	An instance of binary boolean CSP, and a nonnegative integer k.
Parameter:	k.
Problem:	Find a set of at most k constraints such that the remaining constraints are satisfiable after removing them, or report no such set exists.

The main result of this paper is

Theorem 1. *Let $\mathcal{R} = \mathcal{R}_{decisive}$ be the set of binary boolean decisive relations. Then p-$\mathcal{R}$-Almost-CSP is fixed-parameter tractable.*

The relations in $\mathcal{R}_{other}$ are very special and easy to handle in our model. Based on the algorithm for decisive case, we obtain the following corollary:

Corollary 1. *Let $\mathcal{R} = \mathcal{R}_{decisive} \cup \mathcal{R}_{other}$, then p-$\mathcal{R}$-Almost-CSP is fixed-parameter tractable.*

2.5 Graph and separator

Let $G := (V, E)$ be an undirected graph, $U \subseteq V$ be a set of vertices, $S \subseteq E$ be a set of edges. A path $P := \{e_1, \ldots, e_s\}$ of length s from u to v is a set of s edge such that $u \in e_1, v \in e_s, e_i \cap e_{i+1} \neq \varnothing$ for $1 \leq i < s$.

We denote the set of vertices reachable from U in $G' := (V, E \backslash S)$ by $R(U, S)$. Let $X, Y \subset V$ and $X \cap Y = \varnothing$, a set of edges T is called an (X, Y)-separator if $Y \cap R(X, T) = \varnothing$. An (X, Y)-separator is minimal if none of its proper subsets is an (X, Y)-separator. An (X, Y)-separator S' dominates an (X, Y)-separator S if $|S'| \leq |S|$ and $R(X, S) \subsetneq R(X, S')$. For singleton set $\{u\}$, we may write it as u for simplicity.

3 Reduction by Iterative Compression

In this section, we use the method of iterative compression to reduce p-$\mathcal{R}$-Almost-CSP to a variant edge-separation problem. Similar reductions can be found in [13]. Unless otherwise specified, all the relations in this section belong to $\mathcal{R}_{decisive} \backslash \{R_{11}\}$ because constraints of type R_{11} are unsatisfiable and can be removed in advance.

Given a CSP instance $I = (X, \mathcal{C})$, where $\mathcal{C} = \{\langle t_1, R_1 \rangle, \ldots, \langle t_n, R_n \rangle\}$ consists of n decisive constraints and an integer $k \geq 0$. Then consider $n+1$ instances $I_0, \ldots, I_n$ where $I_i = (X, \mathcal{C}_i)$ and $\mathcal{C}_i$ consists of first i constraints of $\mathcal{C}$. Note that $I_n = I$. We solve (I_i, k) for $i = 1, \ldots, n$ one by one.

Since $k \geq 0$, (I_0, k) is obviously a true instance. If for some $i \leq n$, (I_i, k) is a false instance, then we know that (I, k) is also a false instance. Now assume for some $m < n$ all (I_i, k) with $i \leq m$ are true instance, we need to decide (I_{m+1}, k).

We know that (I_m, k) is a true instance, let $\mathcal{S}$ be one of its solution sets where $|\mathcal{S}| \leq k$, then $\mathcal{S}' := \mathcal{S} \cup \{\langle t_{m+1}, R_{m+1} \rangle\}$ is a solution set for I_{m+1}. If $|\mathcal{S}'| \leq k$ then (I_{m+1}, k) is a true instance and we are done. Otherwise, we give an algorithm that either construct a solution set $\mathcal{T}$ of size at most k or report no such set exists.

To this end, we enumerate $\mathcal{ST} \subseteq \mathcal{S}'$ and consider the CSP instance $I' = (X, \mathcal{C}')$, where $\mathcal{C}' := \mathcal{C}_m \backslash \mathcal{ST}$. It is easy to see that the following holds:

Claim 1 *Let $\mathcal{T} \subseteq \mathcal{C}_{m+1}$ be a set of constraints. Then $\mathcal{T}$ is a solution set of I_{m+1} if and only if for $\mathcal{ST} := \mathcal{S}' \cap \mathcal{T}$, $\mathcal{T} \backslash \mathcal{ST}$ is a solution set of $I' = (X, \mathcal{C}')$ where $\mathcal{C}' := \mathcal{C}_m \backslash \mathcal{ST}$.*

Since $(\mathcal{T} \backslash \mathcal{ST}) \cap (\mathcal{S}' \backslash \mathcal{ST}) = \varnothing$, we come to the following problem:

PROBLEM 1

Input: A binary boolean CSP I, a set of constraints $\mathcal{S}$ with $|\mathcal{S}| \leq k_1$ such that I is satisfiable after removing $\mathcal{S}$ and an integer $k_2 \geq 0$.

Parameter: $k_1 + k_2$.

Problem: Find a set of restrictions $\mathcal{T}$ with $|\mathcal{T}| \leq k_2$ such that $\mathcal{S} \cap \mathcal{T} = \varnothing$ and I is satisfiable after removing $\mathcal{T}$.

Lemma 1. *Problem 1 is fixed-parameter tractable.*

We first extend our terminologies. Let $\mathcal{S}$ be a set of constraints, then $V(\mathcal{S})$ is the set of variables appearing in $\mathcal{S}$. Let $I := (X, \mathcal{C})$ be a *satisfiable* binary boolean CSP instance, then I has a satisfiable assignment $F : X \to \{0, 1\}$.

Now let $(I := (X, \mathcal{C}), \mathcal{S}, k_1, k_2)$ be an instance of Problem 1, and let $I' := (X, \mathcal{C} \backslash \mathcal{S})$. We enumerate all the assignments $F : V(\mathcal{S}) \to \{0, 1\}$ such that F satisfies $\mathcal{S}$. The following claim is straightforward:

Claim 2 *The instance $(I := (X, \mathcal{C}), \mathcal{S}, k_1, k_2)$ has a solution set $\mathcal{T}$ if and only if for some $F : V(\mathcal{S}) \to \{0, 1\}$ that satisfies $\mathcal{S}$, I' contains a set of constraints $\mathcal{T}'$ such that (1) $|\mathcal{T}'| \leq k_2$ and (2) after removing $\mathcal{T}'$ in I', there exists a satisfiable assignment of I' consistent with F.*

Proof. For the forward direction, let $\mathcal{T}$ be a solution set of $(I, \mathcal{S}, k_1, k_2)$ and F_0 be a satisfiable assignment of $(X, \mathcal{C} \backslash \mathcal{T})$. Then $\mathcal{T}' := \mathcal{T}$ and F_0 fulfill our requirement.

Conversely, given $\mathcal{T}'$ and a satisfiable assignment F' of I' after removing $\mathcal{T}'$ such that the restriction of F' on $V(\mathcal{S})$ satisfies $\mathcal{S}$. Then $\mathcal{T} := \mathcal{T}'$ is a solution set of $(I, \mathcal{S}, k_1, k_2)$ since F' is a satisfiable assignment of $(X, \mathcal{C} \backslash \mathcal{T})$. □

Thus it suffices to solve Problem 2 in FPT time:

PROBLEM 2	
Input:	A satisfiable binary boolean CSP I, a partial assignment F and an integer $k \geq 0$.
Parameter:	k.
Problem:	Find a set of constraints $\mathcal{T}$ with $|\mathcal{T}| \leq k$ such that after removing $\mathcal{T}$ in I, there exists a satisfiable assignment of I consistent with F.

Lemma 2. *Problem 2 is fixed-parameter tractable.*

Since $I = (X, \mathcal{C})$ is satisfiable, let $A : X \to \{0, 1\}$ be one of its satisfiable assignment. If A is consistent with F, then we are done. Otherwise, let $D(F)$ be the domain of F, then for some variable $x \in D(F)$, we have $F(x) \neq A(x)$. Let $D \subseteq D(F)$ be the set of all such variables. Let $v \notin X$, for every $x \in D$, if $F(x) = 0$ then replace x by $\bar{v}$ in I; if $F(x) = 1$ then replace x by v in I. Let I' be the new instance after replacement.

Claim 3 *(I, F, k) contains a solution set $\mathcal{T}$ if and only if there is a set of constraints $\mathcal{T}'$ with $|\mathcal{T}'| \leq k$ and after removing $\mathcal{T}'$ in I', there is an assignment A' satisifying $A'(v) = 1$ and A' agrees with F on $D(F) \backslash D$.*

Proof. First assume (I, F, k) contains a solution set $\mathcal{T}$. We construct $\mathcal{T}'$ as follows: for every $C = ((x_1, x_2), R_C) \in \mathcal{T}$, if $x_1, x_2 \notin D$, then add C to $\mathcal{T}'$; otherwise, let C' be the constraint obtained from C by replacing the variable in D by v, and add C' to $\mathcal{T}'$. Let A be a satisfiable assignment of (I, F, k) after removing

$\mathcal{T}$ and A is consistent with F. Define an assignment A' on $X \backslash D \cup \{v\}$ where A' agrees with A on $X \backslash D$ and $A'(v) = 1$. By the definition of I', A' is a satisfiable assignment of I' after removing $\mathcal{T}'$ and A' agrees with F on $D(F) \backslash D$.

The converse can be proved analogously and thus we omit it. □

Therefore we reduce Problem 2 to the following:

PROBLEM 3

Input: A satisfiable binary boolean CSP $I := (X, \mathcal{C})$, a partial assignment F, a variable v and an integer $k \geq 0$. It is known that there is a satisfiable assignment A of I consistent with F and $A(v) = 0$.

Parameter: k.

Problem: Find a set of constraints $\mathcal{T}$ with $|\mathcal{T}| \leq k$ such that after removing $\mathcal{T}$ in I, there exists a satisfiable assignment of I, say A, such that A is consistent with F and $A(v) = 1$.

Next, we interpret Problem 3 as a graph separation problem.

Here each variable corresponds to a vertex and each constraint corresponds to an edge. An edge has an annotated type indicating the constraint upon the edge. Then a satisfiable assignment corresponds to a way to color each vertex with 0 or 1 such that all the edge constraints are satisfied.

First assume the graph is connected, without loss of generality, since between disconnected components there are no constraints. Since all the relations are decisive, if one vertex is assigned with some value, then to satisfy the constraints, the value of all the reachable vertices is determined. Our goal is to flip the value of v in a satisfiable assignment F while keeping the value of some other set of vertices S. To do this vertices set S should be separated from v. We denote this set of vertices by S_1. Furthermore, let $e = \{w, u\}$ be an edge where at least one of w and u is not in S and the type of e is in $\mathcal{R}_{and}$, then we have to either separate $\{w, u\}$ with v or remove edge e. We denote this set of edges by S_2. Therefore, the problem is equivalent to the following:

p-MINMIXEDCUT

Input: An undirected graph $G := (V, E)$, a vertex $t \in V$, a set of vertices $S_1 := \{u_1, \ldots, u_p\}$ and a set of pairs of vertices $S_2 := \{\{v_1, w_1\}, \ldots, \{v_q, w_q\}\}$ where each $\{v_i, w_i\}$ is an edge in G. An integer $k \geq 0$.

Parameter: k.

Problem: Find a set of at most k edges T, such that (1) T is a separator with respect to S_1 and t; (2) For every pair $\{v, w\}$ in S_2, either edge $\{v, w\} \in T$ or T is a separator with respect to $\{v, w\}$ and t.

4 *p*-**MinMixedCut** is Fixed-Parameter Tractable

The algorithm employs the method of bounded search tree. For each pair $\{v, w\} \in S_2$, we branch into two cases: either add $\{v, w\}$ to the solution set or separate them from t. To bound the width of each branch, we use the similar idea of *important separator* in [10].

Definition 2. *Let $G := (V, E)$ be an undirected graph. Let $X, Y \subset V$ and $X \cap Y = \varnothing$, a set of edges S is an important (X, Y)-separator if it is minimal and there is no (X, Y)-separator S' that dominates S.*

We show that it is enough to enumerate all the important separators in the branches.

Lemma 3. *Given an instance (G, t, S_1, S_2, k) of p-MinMixedCut, if there is a solution set T of size at most k, then there exists a solution set T' of size at most k such that (1) for every vertex $u \in S_1$, some subset of T' is an important (u, t)-separator and (2) for every pair $\{v, w\} \in S_2$, if the edge $\{v, w\} \notin T$, then some subset of T' is an important $(\{v, w\}, t)$-separator.*

Proof. We only prove (1), the proof of (2) is analogous. Let u be a vertex in S_1 and $S \subseteq T$ be a minimal (u, t)-separator. If S is an important (u, t)-separator, then we are done, otherwise, there is an edge set S' that dominates S, we show that $T' := (T \backslash S) \cup S'$ is also a solution set of size at most k.

Assume on the contrary that T' is not a solution, we distinguish between two cases:

(a) These is a vertex u' such that u' is separated from t by T but not by T'. This is impossible because every path P from u' to t intersects either $T \backslash S$ or S, and S' dominates S, hence P intersects with T'.
(b) For some edge $e := \{v, w\} \in S_2$, $e \in S$ and $e \notin S'$. Since $e \in S$ and S is minimal, v, w belong to different connected components after removing S. Without loss of generality, assume $v \in R(u, S)$ and $w \in R(t, S)$. Since S' dominates S, $v \in R(u, S')$, hence $w \in R(u, S')$. Therefore both v and w are separated from t by S' and by T' as well.

To prove (2), we can contract $\{v, w\}$ to a single vertex in G and use the same argument above. □

This lemma implies that to separate every vertex u from t, it suffices to enumerate important (u, t)-separator, thus settling the correctness of our algorithm.

Next, the number of important separators can be bounded by a function of k. Essentially, this enables us to bound the number of branches in the search tree.

Lemma 4 ([11, 2]). *Let $G := (V, E)$ be an undirected graph. There are at most 4^k important (X, Y)-separator of size at most k for every $X, Y \subseteq V$. Furthermore, all the important separators can be enumerated in $O^*(4^k)$ time.*

Therefore the following algorithm solves p-MinMixedCut in FPT time.

MinMixedCut(G, t, S_1, S_2, k)

Input: An undirected graph $G := (V, E)$, a vertex $t \in V$, a set of vertices S_1, and a set of pairs of vertices S_2. An integer $k \geq 0$.
Output: A set of edges T that fulfills our requirement, or return 'NO' if no such set exists.

1. if S_2 is nonempty and $k > 0$, choose $p = \{u, v\} \in S_2$ such that $p \in E$ and t is reachable from $\{u, v\}$ in G
   1.1 $T \leftarrow$ **MinMixedCut**$(G' := (V, E \backslash \{p\}), t, S_1, S_2 \backslash \{p\}, k-1)$
   1.2 if T is not 'NO' then return $T \cup \{p\}$
   1.3 for all important $(\{u, v\}, t)$-separator S such that $|S| \leq k$
      1.3.1 $T \leftarrow$ **MinMixedCut**$(G' := (V, E \backslash S), t, S_1, S_2 \backslash S, k - |S|)$
      1.3.2 if T is not 'NO' then return $T \cup S$
   1.4 return 'NO'
2. $T \leftarrow$ minimum edge cut from S_1 to $\{t\}$ in G
3. if $|T| \leq k$ return T else return 'NO'

To evaluate the running time of the above algorithm, consider its search tree T. The depth of T is at most k since in every recursive call for **MinMixedCut**, k decreases by 1 at least.

Next we consider the number of nodes in T. Since there are two branches in step 1.1, 1.3, respectively, and by Lemma 4 there are at most 4^{k+1} branches in step 1.3, so the total number of branches is at most $1 + 4^{k+1}$. Thus the size of T is $O(4^{k^2})$ and the total running time of the algorithm is $O^*(4^{k^2})$.

5 Main Theorem

In this section, we prove Theorem 1 and Corollary 1.

Proof (of Theorem 1).
Given a p-Almost-CSP instance (I, k), the main algorithm first reduces it to an instance of p-MinMixedCut (G, t, S_1, S_2, k'), following the procedure described in Section 3. Then it solves the instance by using the algorithm described in Section 4.

Now we evaluate the running time of above algorithm step by step:

1 p-Almost-CSP to Problem 1
Let (I, k) be an instance of p-Almost-CSP. There are at most $|I|$ iterations. For each iteration, we enumerate at most 2^k $\mathcal{ST}$. The resulting instance $(I_1, \mathcal{S}, k_1, k_2)$ of Problem 1 satisfies $|I_1| \leq |I|, |\mathcal{S}| \leq k, k_1 + k_2 \leq k$.

2 Problem 1 to Problem 2
Let $(I_1, \mathcal{S}, k_1, k_2)$ be an instance of Problem 1. We need to enumerate at most $2^{|\mathcal{S}|} \leq 2^k$ assignments F, and for each F, we get a new instance (I_2, F, k_3) of Problem 2 where $|I_2| \leq |I_1|, k_3 = k_2 \leq k$.

3 Problem 2 to Problem 3
Let (I_2, F, k_3) be an instance of Problem 2, we reduce it to an instance (I_3, F, v, k_4) of Problem 3 where $|I_3| \leq |I_2|, k_4 = k_3$ in $O(|I_2|)$ time.

4 Problem 3 to p-MinMixedCut
Let (I_3, F, v, k_4) be an instance of Problem 3, we reduce it to an instance (G, t, S_1, S_2, k') of p-MinMixedCut where $|G| + |S_1| + |S_2| = O(|I_3|))$ and $k' = k_4$ in $O(|I_3|)$ time.

So the total runtime of this procedure is $O(|I| \cdot 2^k \cdot 2^k \cdot |I|) = O(4^k |I|^2)$.

For every instance $(G := (V, E), t, S_1, S_2, k')$, we can solve it in $O^*(4^{k^2}) = O^*(4^{k^2})$. Therefore our algorithm for p-$\mathcal{R}$-Almost-CSP where $\mathcal{R}$ is the set of decisive relations runs in $O^*(4^k \cdot 4^{k^2}) = O^*(4^{k^2})$. □

Now we prove Corollary 1.

Proof (of Corollary 1). Let $I := (X, \mathcal{C})$ be an instance of binary boolean CSP. We have five more relations now, i.e. $\mathcal{R}_{other}$. First R_{12} can be ignored since it is always satisfied. For other four relations, note that $R_{13}(x, y) =$ "$x = 0$", $R_{14}(x, y) =$ "$y = 0$", $R_{15}(x, y) =$ "$y = 1$",$R_{16}(x, y) =$ "$x = 1$", thus they can be reduce to equality relation by adding two variables $\mathbf{1}$ and $\mathbf{0}$ into X. For all constraints in $\mathcal{C}$ that is of type $R_{13}(x, y), R_{14}(x, y), R_{15}(x, y), R_{16}(x, y)$, replace them by $R_5(x, 0), R_5(y, 0), R_5(y, 1), R_5(x, 1)$ respectively. Then this instance can be solved in the same way as Theorem 1, except in the reduction from Problem 1 to Problem 2, we enumerate all $F : V(\mathcal{S}) \cup \{\mathbf{0}, \mathbf{1}\} \to \{0, 1\}$ such that $F(\mathbf{0}) = 0, F(\mathbf{1}) = 1$ instead. □

6 Conclusions and Open Problems

In this paper we discussed the p-$\mathcal{R}$-Almost-CSP problem. By utilizing the powerful techniques of iterative compression and important separators, we solved for the case of decisive relations. To deal with the general case, however, the biggest technical challenge is about how to deal with OR type relations and decisive relations together.

7 Acknowledgements

This research was partially supported by the National Nature Science Foundation of China (60970011 & 61033002).

We are grateful to anonymous referees for pointing out some mistakes and their suggestion for presentation.

References

1. J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.
2. J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of the ACM (JACM)*, 55(5):21, 2008.
3. E. Demaine, G. Gutin, D. Marx, and U. Stege. Open problems from dagstuhl seminar 07281, available electronically at http://drops. dagstuhl. de/opus/volltexte/2007/1254/pdf/07281. Technical report.
4. R.G. Downey and M.R. Fellows. *Parameterized complexity*. Springer New York, 1999.
5. J. Flum and M. Grohe. *Parameterized complexity theory*. Springer-Verlag New York Inc, 2006.
6. J. Guo, J. Gramm, F. Huffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386–1396, 2006.
7. J. Guo, H. Moser, and R. Niedermeier. Iterative compression for exactly solving np-hard minimization problems. *Algorithmics of Large and Complex Networks*, pages 65–80, 2009.
8. S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.
9. M. Mahajan and V. Raman. Parametrizing above guaranteed values: Maxsat and maxcut. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 4, 1997.
10. D. Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006.
11. D. Marx. Important separators and parameterized algorithms (http://www.cs.bme.hu/ dmarx/papers/marx-mds-separators-slides.pdf), February 2011.
12. R. Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31. Oxford University Press, USA, 2006.
13. I. Razgon and B. O'Sullivan. Almost 2-sat is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009.
14. B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.
15. S. Wernicke. *On the algorithmic tractability of single nucleotide polymorphism (SNP) analysis and related problems*. PhD thesis, 2003.