On the Approximability of Parameterized Minimum Monotone Satisfying Assignment

Venkatesan Guruswami
Simons Institute for the Theory of Computing, University of California, Berkeley, CA, USA
Departments of EECS and Mathematics, University of California, Berkeley, CA, USA

Bingkai Lin
State Key Laboratory of Novel Software Technology, Nanjing University, China

Xuandi Ren
Department of EECS, University of California, Berkeley, CA, USA

Xin Zheng
State Key Laboratory of Novel Software Technology, Nanjing University, China

Abstract

The parameterized Minimum Monotone Satisfying Assignment (k-MMSA) problem asks whether a monotone Boolean circuit admits a satisfying assignment of Hamming weight at most k. The MMSA hierarchy is defined by allowing a bounded number of alternations between AND and OR gates in the circuit. While the polynomial-time approximability of the MMSA hierarchy has been studied extensively, much less is known in the parameterized setting. In particular, k-MMSA$_2$ is the well-known k-SetCover problem, whose parameterized inapproximability lies in the $\mathsf{polylog}(n)$ regime. In contrast, k-MMSA$_4$ captures k-MinLabel, for which known lower bounds give $\mathsf{poly}(n)$ inapproximability. Sandwiched by k-MMSA$_2$ and k-MMSA$_4$, the inapproximability of k-MMSA$_3$ remained comparatively unexplored.

In this paper, we give an FPT-time $O(2^k \log n)$-approximation algorithm for k-MMSA$_3$, suggesting that in the fixed-parameter regime, the third level of MMSA remains surprisingly close to the second level. Complementing this algorithm, we also give an FPT-time gap-preserving reduction from k-MMSA$_3$ to k-MMSA$_2$. Thus, stronger inapproximability for k-MMSA$_3$ would imply new hardness for k-MMSA$_2$, potentially offering a route around the current barriers for the latter problem.

Revisiting Marx's reduction from k-MMSA$_t$ to gap k-MMSA$_{t+2}$, we also show that k-MMSA$_4$ admits no $n^{o(1)}$-factor FPT approximation unless W[2]=FPT, and no $n^{O(1/k)}$-factor approximation running in $n^{o(k)}$ time under ETH. These results separate the parameterized approximability behavior of the third and fourth levels and clarify where stronger inapproximability enters the k-MMSA hierarchy.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Approximation algorithms analysis; Theory of computation → Problems, reductions and completeness

Keywords and phrases Parameterized approximation, Minimum Monotone Satisfying Assignment, Set Cover, inapproximability

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.20

Category APPROX

Funding *Venkatesan Guruswami*: Research supported in part by NSF grant CCF-2211972, a DARPA grant under Contract No. HR0011262E031, and a Simons Investigator award.
Xuandi Ren: Supported in part by NSF grant CCF-2228287 and V.G.'s Simons Investigator award.

Acknowledgements We thank Yican Sun for helpful early discussions with the authors on the k-MMSA problem.

1 Introduction

Given a Boolean circuit C consisting of only $\wedge$ and $\vee$ gates on n input bits, the Minimum Monotone Satisfying Assignment (MMSA) problem asks to find a satisfying assignment of C with minimum Hamming weight.

MMSA was first introduced by Alekhnovich, Buss, Moran and Pitassi [1]. In a later work, Dinur and Safra [9] defined a hierarchy of MMSA problems by restricting the number of alternations of the monotone circuit[1]. Specifically, MMSA_t refers to the problem where the alternation starts with an $\wedge$, and has at most t levels. This hierarchy turns out to be equivalent to an AND/OR scheduling hierarchy suggested by Goldwasser and Motwani [15].

Looking into this hierarchy, MMSA_1 can be solved easily by simply outputting the width of the $\wedge$ gate. MMSA_2 is equivalent to the classical Set Cover problem, by identifying each set with a variable and each element to cover with an inner disjunction. Thus, MMSA_2 inherits the $O(\log n)$-approximation [16, 21, 7] and $(1 - o(1)) \ln n$ NP-inapproximability [8] of Set Cover. In [9], Dinur and Safra showed a $2^{\log^{1-o(1)} n}$ inapproximability for MMSA_3 via a reduction from PCP, and placed the Label-Cover problem[2] between MMSA_3 and MMSA_4. Recently, Chlamtáč, Makarychev, and Vakilian [6] revisited the polynomial-time approximability of this hierarchy. They gave an $\widetilde{O}(N^{1/3})$-approximation for MMSA_3 and MMSA_4, and more generally an $\widetilde{O}(N^{1-\delta_t})$-approximation for MMSA_t, where N is the total number of gates and variables, and $\delta_t = \frac{1}{3} \cdot 2^{3-\lceil t/2 \rceil}$.

In a separate line of work, Umans [23] studied the *semantic* variant of Minimum Monotone Satisfying Assignment problem. In his formulation, the input is a circuit that is promised to accept a monotone set of strings. In contrast, [1, 9] studies the syntactic variant where the circuit itself is required to be monotone. Umans showed the NP-hardness of an $n^{1/5-\varepsilon}$-approximation for the semantic problem.

In the parameterized world, there is even more connection between MMSA and other important problems. Parameterized by the target Hamming weight k, the unbounded-alternation circuit version of k-MMSA is $\mathsf{W}[\mathsf{P}]$-complete, whereas the corresponding formula version is $\mathsf{W}[\mathsf{SAT}]$-complete; imposing constant-depth and weft-t restrictions on weighted circuit satisfiability yields the usual $\mathsf{W}[t]$-hierarchy [13, 11]. As for the bounded-depth variants, k-MMSA_2 is equivalent to k-SETCOVER, the canonical $\mathsf{W}[2]$-complete problem [10, 13, 11]. Inapproximability for k-SETCOVER was known under different complexity assumptions: there are no $n^{o(k)}$ time (respectively, $n^{k-\varepsilon}$ time) $(\frac{\log n}{\log \log n})^{1/k}$-approximation under ETH (respectively, SETH) [19]; no FPT time $(\log n)^{o(1)}$-approximation under $\mathsf{W}[1] \neq \mathsf{FPT}$ [17, 19][3]; and no FPT time constant approximation under $\mathsf{W}[2] \neq \mathsf{FPT}$ [20].

Despite these substantial results, the parameterized inaprpoximability of k-SETCOVER is not yet fully understood. Several central questions remain open and have been raised repeatedly in the literature [12, 17, 19, 18, 20]. Can the FPT-time inapproximability factor be strengthened to match the $O(\log n)$-approximation by the greedy algorithm? Can the total-FPT-inapproximability[4] be based on the minimal assumption $\mathsf{W}[2] \neq \mathsf{FPT}$? Under

---

[1] Strictly speaking, [9] studied monotone *formulae* rather than general *circuits*. For fixed depth t, the two are interconvertible with polynomial blow-up. The distinction matters only in the unbounded-depth setting, e.g., in our later discussion on $\mathsf{W}[\mathsf{P}]$/$\mathsf{W}[\mathsf{SAT}]$-completeness of unbounded-alternation k-MMSA.

[2] This is the minimization Label-Cover problem in the Arora–Lund sense [3], rather than the Projection-game formulation, where the value is the maximum fraction of satisfied constraints.

[3] The factor in the statement of [17] is $(\log n)^{1/\mathsf{poly}(k)}$, but by a simple padding trick, we can actually get any FPT-time $(\log n)^{o(1)}$-factor inapproximability. See Lemma 2.8 for a formal discussion.

[4] By total-FPT-inapproximability, we mean ruling out FPT-time $f(k)$-approximation algorithms for every computable function f.

ETH, or even Gap-ETH, can one go beyond the current $(\log n)^{1/k}$-scale barrier for $n^{o(k)}$-time inapproximability? These questions motivate the study of k-MMSA$_3$, the closest natural extension of k-MMSA$_2$ in the k-MMSA hierarchy. In particular, we ask whether the additional expressive power of k-MMSA$_3$ enables reductions that bypass the known barriers for k-SetCover, and whether the techniques developed for k-MMSA$_3$ can in turn shed light on k-SetCover itself.

We remark that Dinur and Safra's approach [9] for non-parameterized MMSA$_3$ relies on building one Boolean variable $L[x, a]$ for each (variable, value) pair in the PCP construction. In the yes case, selecting exactly one value per PCP variable yields a satisfying assignment of weight n, where n is the number of variables. However, this approach does not extend to the parameterized setting: to keep the solution weight bounded by k, one would need to start from a *parameterized* CSP with only k variables, but for such CSPs, even just greedily satisfying the constraints adjacent to the maximum degree variable gives a $k^{O(1)}$ factor approximation[5], which precludes deriving any inapproximability factor that grows with n.

With an extra level of alternation, k-MMSA$_4$ possesses a stronger inapproximability. In his paper studying parameterized monotone and antimonotone circuit satisfiability, Marx [22] gave a gap-creating reduction from any k-MMSA$_t$ to k-MMSA$_{t+2}$, and thus ruled out any FPT-time $f(k)$-approximation for k-MMSA$_4$ under $\mathsf{W}[2] \neq \mathsf{FPT}$. We remark that k-MMSA$_4$ also captures the k-MinLabel problem defined in [4]. It was known in [4] that, under Gap-ETH, any $f(k)$-approximation of k-MinLabel requires $n^{\Omega(k)}$ time. It was also implicit in [17, 18] that, k-MinLabel is hard to approximate within an $n^{o(1)}$-factor[6] in FPT time under $\mathsf{W}[1] \neq \mathsf{FPT}$, or an $n^{1/\mathrm{poly}(k)}$-factor in $n^{o(k)}$ time under ETH, via reductions from k-MaxCover.

Our Contributions

In this paper, we give an FPT time $O(2^k \log n)$-approximation algorithm for k-MMSA$_3$ (Theorem 3.1), which shows, in contrast to the non-parameterized world, that approximating k-MMSA$_3$ is not significantly harder than approximating k-MMSA$_2$. On the hardness side, we give a reduction from (k, h)-gap k-MMSA$_3$ to $(2^k, h/k)$-gap k-MMSA$_2$ (Theorem 4.1). This connection has two implications. First, it helps explain why stronger inapproximability for k-MMSA$_3$ has not appeared separately from k-MMSA$_2$. Second, via this reduction, any sufficiently strong lower bound for k-MMSA$_3$ would yield new FPT inapproximability results for k-SetCover. In particular, a $(\log n)^{\Omega(1)}$-factor FPT inapproximability for k-MMSA$_3$ under $\mathsf{W}[1] \neq \mathsf{FPT}$ would imply the same for k-SetCover. Similarly, under $\mathsf{W}[2] \neq \mathsf{FPT}$, an $\omega(2^k)$ FPT inapproximability for k-MMSA$_3$ would translate into a superconstant FPT inapproximability for k-SetCover, resolving long standing open problems.

We also revisit Marx's reduction [22] and show that it yields stronger consequences than previously recognized. By carefully picking the parameters, we show k-MMSA$_4$ is hard to approximate within an $n^{o(1)}$-factor[6] in FPT time under $\mathsf{W}[2] \neq \mathsf{FPT}$, and hard to approximate within an $n^{O(1/k)}$-factor in $n^{o(k)}$ time under ETH, improving the previous $f(k)$-factor [22] and $n^{1/\mathrm{poly}(k)}$-factor [17, 18], respectively.

We summarize the (in)approximability results for parameterized MMSA hierarchy in Table 1.

[5] We may assume without loss of generality that the parameterized CSP has arity 2. Arity-q CSPs reduce to arity 2 by the standard label-coverization trick, with only an $O(q)$-factor loss in the gap.

[6] Requires padding, see Lemma 2.8.

Table 1 Approximability and Inapproximability Results for Parameterized k-MMSA.

Problems	Approximability			Inapproximability			
	Ratio	**Runtime**	**Reference**	**Ratio**	**Runtime**	**Assumption**	**Reference**
k-MMSA$_2$	$O(\log n)$	$\mathsf{poly}(n)$	[16, 21] [7]	$\left(\frac{\log n}{\log\log n}\right)^{1/k}$	$n^{o(k)}$	ETH	[19]
				$(\log n)^{1/\mathsf{poly}(k)}$			[17]
				$(\log n)^{o(1)}$	FPT	W[1] $\neq$ FPT	[17, 19]
				any $O(1)$		W[2] $\neq$ FPT	[20]
k-MMSA$_3$	$O(2^k \log n)$	FPT	Theorem 3.1	Inherited from above			
k-MMSA$_4$	n	$\mathsf{poly}(n)$	Trivial	$n^{1/\mathsf{poly}(k)}$	$n^{o(k)}$	ETH	[17, 18]
				$n^{O(1/k)}$			Corollary 5.6
				$n^{o(1)}$	FPT	W[1] $\neq$ FPT	[17, 18]
				any $f(k)$		W[2] $\neq$ FPT	[22]
				$n^{o(1)}$			Corollary 5.7

2 Preliminaries

2.1 Perfect Hash Families

▶ **Definition 2.1** ((n, m, Σ, h)-perfect hash family [14]). *An (n, m, Σ, h)-perfect hash family is a family of m functions $\mathcal{F} = \{f_i : [n] \to \Sigma \mid i \in [m]\}$, such that for every subset $T \subseteq [n]$ where $|T| \leq h$, there is a function $f \in \mathcal{F}$ such that*

$$\forall x, y \in T \text{ with } x \neq y,\ f(x) \neq f(y).$$

In other words, $\mathcal{F}$ is an (n, m, Σ, h)-perfect hash family if for any h distinct elements in $[n]$, there is a function $f \in \mathcal{F}$ which maps them to h distinct values.

When $|\Sigma| \geq h^2$, an (n, m, Σ, h)-perfect hash family can be constructed efficiently and deterministically:

▶ **Lemma 2.2** ([2]). *An (n, m, Σ, h)-perfect hash family with $|\Sigma| = h^2, m = h^{O(1)} \log n$ can be constructed deterministically in time $h^{O(1)} \cdot n \log n$.*

2.2 Parameterized Problems

We begin with a definition of the parameterized version of the MMSA problem.

▶ **Definition 2.3** (k-MMSA$_t$). *For any $t, k \in \mathbb{N}^+$, k-MMSA$_t$ asks whether there is a satisfying assignment with Hamming weight at most k, for a monotone circuit C of the following form.*

$$C(x_1, \dots, x_n) = \begin{cases} \bigwedge_{i_1=1}^{n} \bigvee_{i_2=1}^{n} \cdots \bigwedge_{i_t=1}^{n} x_{w_{i_1,\dots,i_t}}, & t \text{ is odd,} \\ \bigwedge_{i_1=1}^{n} \bigvee_{i_2=1}^{n} \cdots \bigwedge_{i_{t-1}=1}^{n} \bigvee_{i_t=1}^{n} x_{w_{i_1,\dots,i_t}}, & t \text{ is even,} \end{cases}$$

where for every $i_1, \dots, i_t \in [n]$, $w_{i_1,\dots,i_t}$ is an index in $[n]$[7].

[7] We generally allow the width of each gate to be polynomial in the number of input bits, but for simplicity, we can pad unused bits and thus use n for both of them.

We write $\mathrm{SAT}(C)$ *to denote the set of satisfying assignments of* C*, and write* $\mathrm{OPT}(C)$ *to denote the minimum weight of any satisfying assignment.*

For any $h > k$*, we write* (k,h)*-gap MMSA*$_t$ *for the task of distinguishing between* $\mathrm{OPT}(C) \le k$ *and* $\mathrm{OPT}(C) > h$*, or equivalently, finding a satisfying assignment of* C *with weight at most* h *given the promise that* $\mathrm{OPT}(C) \le k$*.*

For the special case of k-MMSA$_3$, it will be convenient to replace the ordered list of literals feeding into each bottom conjunction by the set of input indices appearing in that gate. Specifically, a k-MMSA$_3$ instance can be written as

$$C(x) = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} \bigwedge_{t \in N_{i,j}} x_t,$$

where $N_{i,j} \subseteq [n]$ for every $i, j \in [n]$.

The following lemma shows that given a k-MMSA$_3$ instance, one can prune any bottom conjunction term with $|N_{i,j}| > k$ without hurting the gap.

▶ **Lemma 2.4.** *There is a polynomial-time deterministic algorithm which, given as input a* k*-MMSA*$_3$ *instance*

$$C(x) = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} \bigwedge_{t \in N_{i,j}} x_t,$$

outputs a k*-MMSA*$_3$ *instance*

$$C'(x) = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} \bigwedge_{t \in N'_{i,j}} x_t,$$

where for every $i \in [n]$ *and* $j \in S_i$*,* $|N_{i,j}| \le k$*. Furthermore,*

- *If* $\mathrm{OPT}(C) \le k$*, then* $\mathrm{OPT}(C') \le k$*.*
- $\mathrm{SAT}(C') \subseteq \mathrm{SAT}(C)$*. In particular, for every* $h > k$*, if* $\mathrm{OPT}(C) > h$*, then* $\mathrm{OPT}(C') > h$*.*

Proof. C' is obtained from C by simply removing every bottom conjunction with $|N_{i,j}| > k$. That is, let $S_i := \{j \in [n] \mid |N_{i,j}| \le k\}$, and let

$$C'(x) := \bigwedge_{i=1}^{n} \bigvee_{j \in S_i} \bigwedge_{t \in N_{i,j}} x_t.$$

To pad the size of the second layer to n, we simply duplicate its inputs.

Obviously, $C(x) \ge C'(x)$, hence $\mathrm{SAT}(C') \subseteq \mathrm{SAT}(C)$. On the other hand, suppose $\mathrm{OPT}(C) \le k$, and let x be a satisfying assignment to C with $\|x\|_0 \le k$. Since $|N_{i,j}| > k$ for every $i \in [n]$ and $j \notin S_i$, we have $\bigwedge_{t \in N_{i,j}} x_t = 0$, thus

$$\begin{aligned} C'(x) &= \bigwedge_{i=1}^{n} \left(\left(\bigvee_{j \in S_i} \bigwedge_{t \in N_{i,j}} x_t \right) \vee \left(\bigvee_{j \notin S_i} \bigwedge_{t \in N_{i,j}} x_t \right) \right) \\ &= \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} \bigwedge_{t \in N_{i,j}} x_t = C(x). \end{aligned}$$

Hence x is also a satisfying assignment to C', which implies that $\mathrm{OPT}(C') \le k$. ◀

We next introduce two closely related parameterized problems: k-MaxCover and k-MinLabel.

▶ **Definition 2.5** (k-MaxCover and k-MinLabel)**.** *k-MaxCover and k-MinLabel are both defined on a partitioned bipartite graph $G = (A, B, E)$ where $A = A_1\dot\cup \dots \dot\cup A_k, B = B_1\dot\cup \dots \dot\cup B_m$ and $m \le n$, with different optimizing objectives.*

We say a set $X \subseteq A$ (possibly contain multiple vertices from the same part) covers a right part B_j if there exists some $b \in B_j$, such that for every $i \in [k]$, there exists a vertex $a_i \in A_i \cap X$ which is connected to b.

The goal of k-MaxCover is to pick one vertex a_i from each left part A_i, to cover the maximum fraction of right parts B_j, while the goal of k-MinLabel is to pick minimum-sized subset $X \subseteq A$, to cover every right part B_j. We write the two values as $\mathsf{MaxCover}(G)$ *and* $\mathsf{MinLabel}(G)$*, respectively. Specifically,*

$$\mathsf{MaxCover}(G) = \frac{1}{m}\left(\max_{a_1\in A_1,\dots,a_k\in A_k} \left|\left\{j \in [m] \;\middle|\; \exists b \in B_j \text{ such that } \forall i \in [k],\ (a_i, b) \in E\right\}\right|\right),$$

$$\mathsf{MinLabel}(G) = \min_{X\subseteq A, X \text{ covers every } B_j,\ j\in[m]} |X|\,.$$

There is a canonical reduction from $(1, \varepsilon)$-gap k-MaxCover to $(1, \varepsilon^{-1/k})$-gap k-MinLabel, as formalized in [4, 17]. We omit the proof here and refer the reader to [17, Proposition A.1].

▶ **Lemma 2.6** ([4, 17])**.** *Let $G = (A, B, E)$ be a k-MaxCover/k-MinLabel instance with $A = A_1\dot\cup \cdots \dot\cup A_k$ and $B = B_1\dot\cup \cdots \dot\cup B_m$.*

- *If* $\mathsf{MaxCover}(\Gamma) = 1$*, then* $\mathsf{MinLabel}(\Gamma) = k$*.*
- *If* $\mathsf{MaxCover}(\Gamma) \le \varepsilon$ *for some $\varepsilon > 0$, then* $\mathsf{MinLabel}(\Gamma) \ge (1/\varepsilon)^{1/k} \cdot k$*.*

We remark that k-MinLabel can be written as a restricted special case of MMSA_4. Therefore, the inapproxiambility of k-MinLabel applies to k-MMSA_4 as well. This observation is formalized in the following lemma.

▶ **Lemma 2.7.** *Let $G = (A, B, E)$ be a k-MinLabel instance with $A = A_1\dot\cup \cdots \dot\cup A_k$ and $B = B_1\dot\cup \cdots \dot\cup B_m$. There is a polynomial-time construction of a k-MMSA_4 circuit C_G such that*

$$\mathrm{OPT}(C_G) = \mathsf{MinLabel}(G).$$

Proof. For every left vertex $a \in A$, introduce one Boolean variable x_a. For a right vertex $b \in B$, write

$$N_G(b) = \{a \in A \mid (a, b) \in E\}.$$

Define the monotone circuit

$$C_G(x) = \bigwedge_{j\in[m]} \bigvee_{b\in B_j} \bigwedge_{i\in[k]} \bigvee_{a\in A_i\cap N_G(b)} x_a.$$

After padding dummy gates/variables to make the width of each gate equal to the number of variables, $C_G(x)$ becomes a k-MMSA_4 circuit and can be constructed in polynomial time. It's also easy to check the one-to-one correspondence between a subset $X \subseteq A$ covering every B_j in G and a satisfying assignment of C_G. ◀

We next present a canonical padding lemma which boosts any FPT-time inapproximability for k-MMSA with factor $f(n)^{1/g(k)}$ to $f(n)^{o(1)}$.

▶ **Lemma 2.8** (Boosting the inapproximability of k-MMSA$_t$ by padding). *Fix $t \in \mathbb{N}^+$. Let $f : \mathbb{N} \to \mathbb{R}^+$ be a computable, nondecreasing, unbounded, and padding-stable function in the following sense: for every constant $a \in \mathbb{N}$, there is a sufficiently large $n_0(a)$ such that*

$$f(n+a) \leq f(n)^2 \qquad \forall n \geq n_0(a).$$

Let $g : \mathbb{N} \to \mathbb{R}^+$ be a computable, nondecreasing and unbounded function. Suppose for k-MMSA$_t$, no FPT algorithm can distinguish between

$$\mathrm{OPT}(C) \leq k \qquad \textit{and} \qquad \mathrm{OPT}(C) > k \cdot f(n)^{1/g(k)}.$$

Then, for every computable function $\alpha : \mathbb{N} \to \mathbb{R}^+$ with $\alpha(k) = o(1)$, no FPT algorithm can distinguish between

$$\mathrm{OPT}(C) \leq k \qquad \textit{and} \qquad \mathrm{OPT}(C) > k \cdot f(n)^{\alpha(k)}.$$

In other words, k-MMSA$_t$ doesn't admit an FPT-time $f(n)^{o(1)}$-approximation.

Proof. Fix $\alpha(k) = o(1)$ and choose K sufficiently large such that $\alpha(K) \leq 1/(4g(k))$. Given an MMSA$_t$ circuit C on n variables, add $K - k$ fresh variables which are forced to be one in any satisfying assignment:

$$\widehat{C}(x, y) = C(x) \wedge y_1 \wedge \cdots \wedge y_{K-k}.$$

By representing each new variable as a dummy length-t unary path, the new instance $\widehat{C}$ is still a MMSA$_t$ instance with new parameter K. Furthermore, $\widehat{C}$ has $N = n + K - k$ many variables, satisfies

$$\mathrm{OPT}(\widehat{C}) = \mathrm{OPT}(C) + K - k,$$

and can be constructed in FPT time. Suppose for contradiction that there is an FPT-time algorithm which can distinguish between

$$\mathrm{OPT}(\widehat{C}) \leq K \qquad \text{and} \qquad \mathrm{OPT}(\widehat{C}) > K \cdot f(N)^{\alpha(K)}.$$

Since for sufficiently large N depending only on K, padding-stability gives $f(N) = f(n + K - k) \leq f(n)^2$, and f is unbounded, we have

$$K \cdot f(N)^{\alpha(K)} \leq K \cdot f(n)^{2\alpha(K)} \leq K \cdot f(n)^{1/(2g(k))} \leq K - k + k \cdot f(n)^{1/g(k)}.$$

Thus, the same algorithm can also distinguish between

$$\mathrm{OPT}(C) \leq k \qquad \text{and} \qquad \mathrm{OPT}(C) > k \cdot f(n)^{1/g(k)}$$

in FPT time, a contradiction. ◀

3 An FPT-time $O(2^k \log n)$-Approximation for k-MMSA$_3$

▶ **Theorem 3.1.** *There is a deterministic FPT-time $O(2^k \log n)$-approximation algorithm for k-MMSA$_3$ on n input bits.*

Proof. Write the input instance as

$$C(x) = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} \bigwedge_{t \in N_{i,j}} x_t,$$

where $N_{i,j} \subseteq [n]$ for every $i, j \in [n]$. By Lemma 2.4, we can without loss of generality assume that $|N_{i,j}| \leq k$.

The algorithm proceeds by repeatedly assigning some variables to 1, until the whole instance C is satisfied. Let

$$S = \{i \in [n] \mid x_i = 1\}$$

be the set of variables that are assigned to 1, the i-th clause

$$\left(\bigvee_{j=1}^{n} \bigwedge_{t \in N_{i,j}} x_t\right)$$

is covered if and only if there exists some term $j \in [n]$ such that $N_{i,j} \subseteq S$. Let

$$\mathrm{Cov}(S) = \{i \in [n] \mid \exists j \in [n] \text{ with } N_{i,j} \subseteq S\}.$$

The algorithm starts with $S_0 = \emptyset$. In the r-th iteration, it takes the set $N_{a,b}$ maximizing the number of newly covered indices

$$|\mathrm{Cov}(S_{r-1} \cup N_{a,b}) \setminus \mathrm{Cov}(S_{r-1})|,$$

and sets $S_r = S_{r-1} \cup N_{a,b}$. Given the fact that $|N_{i,j}| \leq k$ for every $i, j \in [n]$, it suffices to prove that the algorithm stops after $O(2^k \log n)$ iterations, and thus outputs a satisfying assignment of C with weight at most $O(k \cdot 2^k \log n)$.

Let $x \in \{0,1\}^n$ be a satisfying assignment of C and let $X = \mathrm{supp}(x)$. Fix the r-th iteration, let

$$U_r = [n] \setminus \mathrm{Cov}(S_r)$$

be the set of currently uncovered indices. For every $i \in U_r$, since x satisfies C, there exists some witness $j = j(i) \in [n]$ with $N_{i,j(i)} \subseteq X$. Every chosen witness is a subset of X, and X has at most 2^k distinct subsets. Therefore, by an averaging argument, some set $W \subseteq X$ appears as $N_{i,j(i)}$ for at least $|U_r|/2^k$ indices $i \in U_r$. The best enumerated set covers at least $|U_r|/2^k$ currently uncovered indices. Consequently,

$$|U_{r+1}| \leq \left(1 - \frac{1}{2^k}\right)|U_r|.$$

Iterating this inequality, for $T = \lceil 2^k \ln n \rceil$,

$$|U_T| \leq n\left(1 - \frac{1}{2^k}\right)^T \leq n \exp\left(-\frac{T}{2^k}\right) < 1.$$

Since $|U_T|$ is an integer, $U_T = \emptyset$. Thus, the algorithm returns a satisfying assignment for C with weight at most $kT = O(k2^k \log n)$.

The algorithm is deterministic and runs in FPT time: it performs $O(2^k \log n)$ iterations, and in each iteration it enumerates at most n^2 candidate sets and computes their gains in polynomial time. This proves the theorem. ◀

4 A Reduction from (k, h)-gap MMSA$_3$ to $(2^k, h/k)$-gap MMSA$_2$

▶ **Theorem 4.1.** *There is an FPT-time reduction, which takes as input a k-MMSA$_3$ instance*

$$F(z) = \bigwedge_{i \in [n]} \bigvee_{j \in [n]} \bigwedge_{t \in N_{i,j}} z_t,$$

on $z \in \{0,1\}^n$, outputs a 2^k-MMSA$_2$ instance G on $x \in \{0,1\}^{n^2}$, such that

- *(Completeness) If $\mathrm{OPT}(F) \leq k$, then $\mathrm{OPT}(G) \leq 2^k$.*
- *(Soundness) Fix any $h > k$, if $\mathrm{OPT}(F) > h$, then $\mathrm{OPT}(G) > h/k$.*

The reduction contracts each distinct bottom conjunction of F into one variable of the k-MMSA$_2$ instance G. Namely, for every distinct set $N_{i,j}$ we create a variable intended to mean that all original variables in $N_{i,j}$ are set to 1. Since there are at most n^2 bottom conjunctions, this gives at most n^2 variables.

For completeness, suppose F has a satisfying assignment z whose support Z has size at most k. Every bottom conjunction that is true under z corresponds to a set $N_{i,j} \subseteq Z$ of size $\leq k$, and there are at most 2^k many such sets. Setting the variables representing these true conjunctions to one gives a satisfying assignment of G of weight at most 2^k.

For soundness, suppose G has a satisfying assignment x of weight at most h/k. Decode x back to an assignment z for F by taking the union of all sets represented by the ones in x. Since every represented set has size at most k, the decoded assignment has weight at most h.

Proof of Theorem 4.1. Let

$$\mathcal{S} := \{N_{i,j} \mid i, j \in [n]\}.$$

By Lemma 2.4, we can without loss of generality assume that every set in $\mathcal{S}$ has size at most k. Since $|\mathcal{S}| \leq n^2$, fix an injective map $f : \mathcal{S} \to [n^2]$. For every $i, j \in [n]$, write

$$s_{i,j} := f(N_{i,j}).$$

For every $s \in \mathrm{Im}(f)$, let T_s denote the unique set in $\mathcal{S}$ such that $f(T_s) = s$.

Define the k-MMSA$_2$ instance

$$G(x) := \bigwedge_{i\in[n]} \bigvee_{j\in[n]} x_{s_{i,j}}.$$

The instance uses at most n^2 variables and has size $n^{O(1)}$. It can clearly be constructed in polynomial time, and hence in FPT time.

Completeness. Suppose there exists $z \in \{0,1\}^n$ such that $F(z) = 1$ and $\|z\|_0 \leq k$. Let

$$Z := \mathrm{supp}(z) = \{t \in [n] : z_t = 1\}.$$

Define $x \in \{0,1\}^{n^2}$ by

$$x_s := \begin{cases} 1, & \text{if } s \in \mathrm{Im}(f) \text{ and } T_s \subseteq Z, \\ 0, & \text{otherwise.} \end{cases}$$

We first show that $G(x) = 1$. Fix any $i \in [n]$. Since $F(z) = 1$, there exists some $j \in [n]$ such that

$$\bigwedge_{t\in N_{i,j}} z_t = 1,$$

equivalently, $N_{i,j} \subseteq Z$. As $T_{s_{i,j}} = N_{i,j}$, the definition of x gives $x_{s_{i,j}} = 1$. Therefore every outer clause of G is satisfied, and hence $G(x) = 1$.

It remains to bound the weight of x. If $x_s = 1$, then $s \in \mathrm{Im}(f)$ and $T_s \subseteq Z$. Because f is injective, distinct coordinates correspond to distinct subsets of Z. Hence

$$\|x\|_0 \leq 2^{|Z|} \leq 2^k.$$

Thus $\mathrm{OPT}(G) \leq 2^k$.

Soundness. Fix $h > k$, and suppose for contradiction that there exists $x \in \{0,1\}^{n^2}$ such that $G(x) = 1$ and $\|x\|_0 \le h/k$. Let

$$\mathcal{A} := \{T_s \mid s \in \mathrm{Im}(f) \text{ and } x_s = 1\}$$

be the family of original bottom sets represented by the selected variables of G, and let

$$U := \bigcup_{T \in \mathcal{A}} T.$$

Define $z \in \{0,1\}^n$ to be the indicator vector of U.

Since every $T \in \mathcal{A}$ has size at most k, and since $|\mathcal{A}| \le \|x\|_0$, we have

$$\|z\|_0 = |U| \le \sum_{T \in \mathcal{A}} |T| \le k \cdot |\mathcal{A}| \le k\|x\|_0 \le h.$$

We now show that $F(z) = 1$. Fix any $i \in [n]$. Since $G(x) = 1$, there exists some $j \in [n]$ such that $x_{s_{i,j}} = 1$. By construction, $T_{s_{i,j}} = N_{i,j}$, so $N_{i,j} \in \mathcal{A}$ and therefore $N_{i,j} \subseteq U = \mathrm{supp}(z)$. Hence

$$\bigwedge_{t \in N_{i,j}} z_t = 1.$$

This holds for every $i \in [n]$, so $F(z) = 1$, contradicting $\mathrm{OPT}(F) > h$. ◀

▶ **Corollary 4.2.** *For any computable function $t(n,k)$, if there is an FPT-time $t(n,k)$-approximation for k-MMSA$_2$, then there is an FPT-time $(t(n^2, 2^k) \cdot 2^k)$-approximation for k-MMSA$_3$. Equivalently, if there is no FPT-time $t(n,k)$-approximation for k-MMSA$_3$, then there is no FPT-time $\frac{t(\sqrt{n}, \log k)}{k}$-approximation for k-MMSA$_2$.*

In particular, a $(\log n)^{\Omega(1)}$-factor FPT inapproximability for k-MMSA$_3$ under W[1] $\neq$ FPT, or an $\omega(2^k)$-factor FPT inapproximability for k-MMSA$_3$ under W[2] $\neq$ FPT, could imply new FPT inapproximability results for k-MMSA$_2$ under the respective assumptions, improving [17, 19, 20].

5 A Reduction from k-MMSA$_t$ to Gap k-MMSA$_{t+2}$

In this section, we first summarize known inapproximability results for k-MMSA$_4$, then introduce [22]'s reduction from k-MMSA$_t$ to k-MMSA$_{t+2}$, and finally plug a better set of parameters into it to get improved inapproximability of k-MMSA$_4$ under W[2] $\neq$ FPT and ETH.

5.1 Inapproximability of k-MMSA$_4$ from Literature

The following inapproximability on k-MaxCover was from [17, 18]. We remark that [17] also provides the inapproximability under other assumptions such as SETH or the k-SUM Hypothesis. However, we choose to present only the most relevant assumptions here.

▶ **Lemma 5.1** (Inapproximability of k-MaxCover [17, 18])**.**

- *Assuming* W[1] $\neq$ FPT, *no FPT algorithm can approximate k-MaxCover within an $n^{1/\mathsf{poly}(k)}$ factor.*
- *Assuming* ETH, *no $n^{o(k)}$-time algorithm can approximate k-MaxCover within an $n^{1/\mathsf{poly}(k)}$ factor.*

Passing Lemma 5.1 through the reduction chain of Lemmas 2.6–2.8, we have:

▶ **Corollary 5.2** (Inapproximability of k-MMSA$_4$, implicit in [17, 18]).

- *Assuming* $\mathsf{W}[1] \neq \mathsf{FPT}$*, no FPT algorithm can approximate* k*-MMSA*$_4$ *within an* $n^{o(1)}$ *factor.*
- *Assuming* ETH*, no* $n^{o(k)}$*-time algorithm can approximate* k*-MMSA*$_4$ *within an* $n^{1/\mathsf{poly}(k)}$ *factor.*

Marx [22] also showed a total-FPT-inapproximability for k-MMSA$_4$ under the weaker assumption $\mathsf{W}[2] \neq \mathsf{FPT}$.

▶ **Lemma 5.3** ([22]). *Assuming* $\mathsf{W}[2] \neq \mathsf{FPT}$*, for any computable function* f*, no FPT algorithm can approximate* k*-MMSA*$_4$ *within an* $f(k)$ *factor.*

5.2 Revisiting Marx's Reduction

▶ **Theorem 5.4** (Adapted from [22]). *Let* $t \in \mathbb{N}^+$ *and let* $h \geq k$*. Given a* k*-MMSA*$_t$ *instance* C *on* n *variables and an* (n, m, Σ, h)*-perfect hash family* $\mathcal{F} = \{f_1, \ldots, f_m\}$ *with* $|\Sigma| \geq k$*, there is a reduction that outputs a* k*-MMSA*$_{t+2}$ *instance* C' *on the same* n *variables such that:*

- *the size and running time are at most* $\left(m \cdot \binom{|\Sigma|}{k} \cdot |C|\right)^{O(1)}$*;*
- *if* C *has a satisfying assignment of weight at most* k*, then the same assignment satisfies* C'*;*
- *if every satisfying assignment of* C *has weight at least* $k+1$*, then every satisfying assignment of* C' *has weight more than* h*.*

Proof. For every $i \in [m]$ and every $T \in \binom{\Sigma}{k}$, write

$$N_{i,T} := f_i^{-1}(T).$$

Define

$$C'(x) := \bigwedge_{i \in [m]} \bigvee_{T \in \binom{\Sigma}{k}} C\big(x \wedge \mathbf{1}_{N_{i,T}}\big).$$

This is a monotone circuit with two additional outer layers, hence an MMSA$_{s+2}$ instance, and its size is bounded by $\left(m \cdot \binom{|\Sigma|}{k} \cdot |C|\right)^{O(1)}$.

For completeness, let x be a satisfying assignment of C with support $X \subseteq [n]$ and $|X| \leq k$. For every $i \in [m]$, choose $T_i \in \binom{\Sigma}{k}$ such that

$$f_i(X) \subseteq T_i.$$

This is possible because $|f_i(X)| \leq |X| \leq k$ and $|\Sigma| \geq k$. Then $X \subseteq N_{i,T_i}$, so

$$x \wedge \mathbf{1}_{N_{i,T_i}} = x.$$

Therefore every outer conjunction of C' is satisfied, and hence $C'(x) = 1$.

For soundness, let x be any satisfying assignment of C' and let $X = \mathrm{supp}(x)$. Since $C'(x) = 1$, for every $i \in [m]$ there exists some $T_i \in \binom{\Sigma}{k}$ such that

$$C\big(x \wedge \mathbf{1}_{N_{i,T_i}}\big) = 1.$$

By the soundness assumption on C, the assignment $x \wedge \mathbf{1}_{N_{i,T_i}}$ must have weight at least $k+1$. Equivalently,

$$|X \cap N_{i,T_i}| \geq k+1 \qquad \text{for every } i \in [m].$$

Suppose toward contradiction that $|X| \leq h$. Since $\mathcal{F}$ is an (n, m, Σ, h)-perfect hash family, there exists some $i^\star \in [m]$ such that $f_{i^\star}$ is injective on X. For this $i^\star$, every $T \in \binom{\Sigma}{k}$ can contain images of at most k elements of X. Thus

$$|X \cap f_{i^\star}^{-1}(T)| \leq k \qquad \forall T \in \binom{\Sigma}{k},$$

contradicting

$$|X \cap N_{i^\star, T_{i^\star}}| \geq k+1.$$

Hence $|X| > h$. ◀

▶ Remark 5.5. We remark that the goal of Marx [22] was to rule out $f(k)$-factor FPT approximation algorithms for any computable function f. Thus, it was sufficient for him to choose h to be larger than the approximation factor $f(k)$. In his hardness proof, h was used as the parameter of an intermediate *FPT cost-approximation algorithm*, which in turn forced h to depend only on k. In contrast, our argument proceeds more directly, allowing us to choose h to be as large as $n^{O(1/k)}$.

We apply Theorem 5.4 using an (n, m, Σ, h)-perfect hash family from Lemma 2.2 with $h = n^{\Theta(1/k)}, |\Sigma| = h^2$ and $m = h^{O(1)} \log n$. Then

$$\binom{|\Sigma|}{k} \leq h^{2k},$$

so the reduction size and running time are bounded by

$$\left(m \cdot \binom{|\Sigma|}{k} \cdot |C|\right)^{O(1)} = \left(h^{O(1)} \log n \cdot h^{2k} \cdot |C|\right)^{O(1)} \leq n^{O(1)}.$$

It was known [5] that under ETH, k-MMSA$_2$ (a.k.a. k-SetCover) cannot be solved in $n^{o(k)}$ time. We thus have the following immediate corollary:

▶ **Corollary 5.6.** *Assuming* ETH*, no $n^{o(k)}$-time algorithm can approximate k-MMSA$_4$ within some $n^{O(1/k)}$ factor.*

Using the same set of parameters and invoking the padding lemma (Lemma 2.8), we also have the following W[2]-hardness:

▶ **Corollary 5.7.** *Assuming* W[2] $\neq$ FPT*, no FPT algorithm can approximate k-MMSA$_4$ within any $n^{o(1)}$ factor.*

Corollary 5.6 and Corollary 5.7 improve upon Corollary 5.2 and Lemma 5.3, respectively.

References

1 Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is np-hard to linearly approximate. *J. Symb. Log.*, 66(1):171–191, 2001. doi:10.2307/2694916.

2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

3 Sanjeev Arora and Carsten Lund. Hardness of approximations. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 399–446. PWS Publishing, Boston, MA, 1997.

4 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-ETH to FPT-inapproximability: Clique, dominating set, and more. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.74.

5 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006. doi:10.1016/j.jcss.2006.04.007.

6 Eden Chlamtáč, Yury Makarychev, and Ali Vakilian. Approximating Red-Blue Set Cover and Minimum Monotone Satisfying Assignment. In Nicole Megow and Adam Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.11.

7 V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/MOOR.4.3.233.

8 I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014. doi:10.1145/2591796.2591884.

9 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5):247–254, 2004. doi:10.1016/J.IPL.2003.11.007.

10 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM J. Comput.*, 24(4):873–921, 1995. doi:10.1137/S0097539792228228.

11 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

12 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/A13060146.

13 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, 2006. doi:10.1007/3-540-29953-X.

14 Michael L. Fredman and János Komlós. On the size of separating systems and families of perfect hash functions. *SIAM Journal on Algebraic Discrete Methods*, 5(1):61–68, 1984. doi:10.1137/0605009.

15 Michael H. Goldwasser and Rajeev Motwani. Intractability of assembly sequencing: Unit disks in the plane. In Frank K. H. A. Dehne, Andrew Rau-Chaplin, Jörg-Rüdiger Sack, and Roberto Tamassia, editors, *Algorithms and Data Structures, 5th International Workshop, WADS '97, Halifax, Nova Scotia, Canada, August 6-8, 1997, Proceedings*, volume 1272 of *Lecture Notes in Computer Science*, pages 307–320. Springer, 1997. doi:10.1007/3-540-63307-3_70.

16 D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974. doi:10.1016/S0022-0000(74)80044-9.

17 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019. doi:10.1145/3325116.

18 Karthik C. S. and Inbal Livni Navon. On hardness of approximation of parameterized set cover and label cover: Threshold graphs from error correcting codes. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 210–223. SIAM, 2021. doi:10.1137/1.9781611976496.24.

19 Bingkai Lin. A simple gap-producing reduction for the parameterized set cover problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 81:1–81:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.81.

20 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. Constant approximating parameterized k-SETCOVER is W[2]-hard. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3305–3316. SIAM, 2023. doi:10.1137/1.9781611977554.ch126.

21 L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975. doi:10.1016/0012-365X(75)90058-8.

22 Dániel Marx. Completely inapproximable monotone and antimonotone parameterized problems. *J. Comput. Syst. Sci.*, 79(1):144–151, 2013. doi:10.1016/J.JCSS.2012.09.001.

23 Christopher Umans. Hardness of approximating sigma$_2{}^{\mathrm{p}}$ minimization problems. In *40th Annual Symposium on Foundations of Computer Science, FOCS 1999, New York, NY, USA, October 17-18, 1999*, pages 465–474. IEEE Computer Society, 1999. doi:10.1109/SFFCS.1999.814619.