Optimally Repurposing Existing Algorithms to Obtain Exponential-Time Approximations

Barış Can Esmer[*1], Ariel Kulik[1], Dániel Marx[†1], Daniel Neuen[2], and Roohani Sharma[3]

[1]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany.
{baris-can.esmer|ariel.kulik|marx}@cispa.de
[2]University of Bremen, Bremen, Germany. dneuen@uni-bremen.de
[3]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany.
rsharma@mpi-inf.mpg.de

June 28, 2023

Abstract

The goal of this paper is to understand how exponential-time approximation algorithms can be obtained from existing polynomial-time approximation algorithms, existing parameterized exact algorithms, and existing parameterized approximation algorithms. More formally, we consider a monotone subset minimization problem over a universe of size n (e.g., VERTEX COVER or FEEDBACK VERTEX SET). We have access to an algorithm that finds an α-approximate solution in time $c^k \cdot n^{\mathcal{O}(1)}$ if a solution of size k exists (and more generally, an extension algorithm that can approximate in a similar way if a set can be extended to a solution with k further elements). Our goal is to obtain a $d^n \cdot n^{\mathcal{O}(1)}$ time β-approximation algorithm for the problem with d as small as possible. That is, for every fixed $\alpha, c, \beta \geq 1$, we would like to determine the smallest possible d that can be achieved in a model where our problem-specific knowledge is limited to checking the feasibility of a solution and invoking the α-approximate extension algorithm. Our results completely resolve this question:

1. For every fixed $\alpha, c, \beta \geq 1$, a simple algorithm ("approximate monotone local search") achieves the optimum value of d.

2. Given $\alpha, c, \beta \geq 1$, we can efficiently compute the optimum d up to any precision $\varepsilon > 0$.

Earlier work presented algorithms (but no lower bounds) for the special case $\alpha = \beta = 1$ [Fomin et al., J. ACM 2019] and for the special case $\alpha = \beta > 1$ [Esmer et al., ESA 2022]. Our work generalizes these results and in particular confirms that the earlier algorithms are optimal in these special cases.

We compare the performance of the resulting algorithms to what is obtainable by brute force, that is, in a setting where we have no problem-specific knowledge beyond checking the feasibility of a solution. We show that, except in the case $\alpha > \beta = 1$, the resulting d is strictly better than what can be obtained by brute force. For example, somewhat counterintuitively, given access to a 1000-approximate extension algorithm running in time $1000^k \cdot n^{\mathcal{O}(1)}$ allows us to obtain a 1.001-approximation algorithm with running time $d^n \cdot n^{\mathcal{O}(1)}$ strictly better than what is possible by brute force. Our technique gives novel results for a wide range of problems including FEEDBACK VERTEX SET, DIRECTED FEEDBACK VERTEX SET, ODD CYCLE TRAVERSAL and PARTIAL VERTEX COVER.
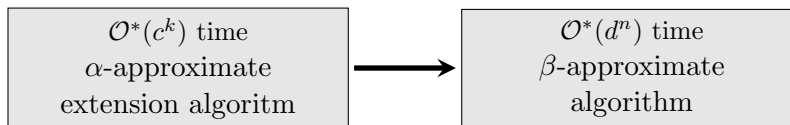

Contents

1 Introduction

It is widely believed that NP-hard problems cannot be solved in polynomial time and any algorithm solving them has some form of exponential running time. During the past decades, there has been a great deal of interest in trying to obtain improved exponential-time algorithms for basic NP-hard problems, see for example the monograph of Fomin and Kratsch [32]. Typically, for *subset problems*, where the goal is to find a subset of a given n-sized universe U that satisfies some property Π, a solution can be found by enumerating all 2^n subsets of U. Therefore, the goal is to design algorithms that beat this exhaustive search and run in time $\mathcal{O}^*(d^n)$[1] for as small $1 < d < 2$ as possible. More recently, there has been interest in *exponential-time approximation algorithms* [3, 5, 9, 21, 25, 26, 56] to obtain approximation ratios that are better than what is considered possible in polynomial time. In this paper, we analyze how the simple technique of monotone local search can be used to derive exponential-time approximation algorithms by repurposing existing exact parameterized algorithms, existing polynomial-time approximation algorithms, and existing parameterized approximation algorithms. Furthermore, we show that monotone local search is the optimal way to convert between those types of algorithms.

Our setting is the following. We consider *subset minimization* problems where the goal is to find a subset of the n-sized universe U of *minimum cardinality* that satisfies some additional property Π. To make approximation feasible, we consider only monotone properties, that is, if $S \subseteq U$ satisfies Π, then so does any superset of S. For any approximation ratio $\beta \geq 1$, we say that a subset $S \subseteq U$ satisfying the property Π is a β-*approximate solution* if $|S| \leq \beta \cdot |\texttt{OPT}|$, where $\texttt{OPT} \subseteq U$ is an optimum solution.

An *exponential β-approximation algorithm* for a subset minimization problem returns a β-approximate solution and runs in time $\mathcal{O}^*(d^n)$ for some $1 < d < 2$. We assume that we are given access to an algorithm with the following specification: given a problem instance and an integer k, if the optimum solution has size at most k, then the algorithm returns a solution of size at most $\alpha \cdot k$ in time $\mathcal{O}^*(c^k)$. Let us observe that in the special case of $c = 1$, it is equivalent to the notion of polynomial-time constant-factor approximation algorithm [63] and in the special case of $\alpha = 1$, it is equivalent to an exact fpt-algorithm [20]. In general, the definition covers constant-factor parameterized approximation algorithms, which have received increased attention recently [7,10,11,14,17,18,28,29,41,42,45,50–52, 54,57]. For technical reasons, instead of an algorithm finding a small solution, we need an algorithm finding a small *extension*: given a set X that can be extended to a solution by k further elements, it returns such an extension with at most $\beta \cdot k$ further elements. For many problems that are defined in terms of deletions (e.g., VERTEX COVER, FEEDBACK VERTEX SET, MULTICUT etc.), the two notions are equivalent via a simple reduction: the extension problem is equivalent to solving the problem on $G - X$.

Our main goal is to understand, for a given α, β, and c, what is the best $\mathcal{O}^*(d^n)$ time β-approximation algorithm we can obtain if we have access to a parameterized α-approximate extension algorithm running in time $\mathcal{O}^*(c^k)$.

$$\boxed{\begin{array}{c} \mathcal{O}^*(c^k) \text{ time} \\ \alpha\text{-approximate} \\ \text{extension algoritm} \end{array}} \longrightarrow \boxed{\begin{array}{c} \mathcal{O}^*(d^n) \text{ time} \\ \beta\text{-approximate} \\ \text{algorithm} \end{array}}$$

The special case when $\alpha = \beta = 1$, that is, using exact fpt-algorithms to obtain exact exponential-time algorithms, was treated by Fomin et al. [31]: they give a very simple procedure, monotone local search, that repurposes an exact fpt-algorithm with running time $\mathcal{O}^*(c^k)$ to obtain an exponential-time algorithm with $d = 2 - \frac{1}{c}$. Monotone local search was extended to an approximate version by Esmer et al. [26] to handle the case $\alpha = \beta > 1$, with a much more complicated (non-closed-form) expression for d, which we denote by $\texttt{amls}_{\alpha=\beta}(\beta, c)$. Some simulated results for the case $\alpha = 1$ and $\beta > 1$ were given in the thesis of Lee [48]. Table 1 shows various special cases of our setting.

These previous results suggest two obvious further research goals. First, one would like to extend the understanding to the $\alpha \neq \beta$ case. For example, Esmer et al. [26] showed how to obtain an

---

[1]The $\mathcal{O}^*$ notation hides polynomial factors in the expression.

c	α, β	
$c = 1$	$\alpha > \beta > 1$	polytime approximation $\rightsquigarrow$ exptime approximation with *better* ratio
$c > 1$	$\alpha = \beta = 1$	fpt exact $\rightsquigarrow$ exptime exact [31]
$c > 1$	$\alpha = 1, \beta > 1$	fpt exact $\rightsquigarrow$ exptime approximation
$c > 1$	$\alpha > 1, \beta = 1$	fpt approximation $\rightsquigarrow$ exptime exact useless, cannot improve 2^n brute force
$c > 1$	$\alpha = \beta > 1$	fpt approximation $\rightsquigarrow$ exptime approximation with the *same* ratio [26]
$c > 1$	$1 < \alpha < \beta$	fpt approximation $\rightsquigarrow$ exptime approximation with *worse* ratio
$c > 1$	$\alpha > \beta > 1$	fpt approximation $\rightsquigarrow$ exptime approximation with *better* ratio

Table 1: Special cases of our setting.

exponential 5-approximation algorithm if we are given an $\mathcal{O}^*(2^k)$ time 5-approximate parameterized extension algorithm (i.e., $c = 2$, $\alpha = \beta = 5$). We would like to understand whether we can obtain a *faster* 5-approximation algorithm if the extension algorithm is 3-approximate ($c = 2$, $\alpha = 3$, $\beta = 5$) and whether the 5-approximate extension algorithm is useful *at all* for obtaining an exponential 3-approximation ($c = 2$, $\alpha = 5$, $\beta = 3$).

Second, the previous results [26,31] did not provide any lower bounds. Is the $\mathcal{O}^*((2-\frac{1}{c})^n)$ algorithm obtained by Fomin et al. [31] really the best we can have without any problem-specific knowledge? We can formalize this question in a model where all we can do is checking the validity of a solution in polynomial time and using an α-approximate extension algorithm running in time $\mathcal{O}^*(c^k)$. If we have lower bounds in this model, then we can evaluate whether the previous results [26,31] really repurposed the extension algorithms in an optimal way and we can compare how the algorithms resulting from two sets of parameters (α, c, β) and (α', c', β') relate to each other.

Our main result fully achieves both of these goals: for every combination of parameters, we provide tight upper and (unconditional) lower bounds on the best possible exponential-time approximation algorithm.

For every fixed $\alpha, \beta, c \geq 1$, we determine the best possible $d = \texttt{best}(\alpha, c, \beta)$ such that a $\mathcal{O}^*(d^n)$ time β-approximation algorithm can be obtained from an α-approximate extension algorithm running in time $\mathcal{O}^*(c^k)$.

Similar to [26], we do not expect a simple closed-form expression for $\texttt{best}(\alpha, c, \beta)$. Indeed, it may very well be that $\texttt{best}(\alpha, c, \beta)$ has no closed-form description similar to, for example, the running time of certain branching algorithms where the base corresponds to the root of a polynomial of degree at least five (see, e.g., [20]). This raises the philosophical question of when can we consider the problem of determining $\texttt{best}(\alpha, c, \beta)$ "resolved." Our answer consists of two parts:

(1) For every $\alpha, \beta, c \geq 1$, a simple approximate mononotone local search algorithm (which naturally extends existing algorithms [26, 31, 48]) achieves the optimal running time (up to polynomial factors; Theorem 2.3).

(2) This algorithm runs precisely in time $\mathcal{O}^*((\texttt{best}(\alpha, c, \beta))^n)$ and given $\alpha, \beta, c \geq 1$ and $\varepsilon > 0$, we can compute $\texttt{best}(\alpha, c, \beta)$ up to an additive error of $\varepsilon > 0$, in time polynomial in the total encoding length of the input (Theorem 2.5).

That is, we describe the optimal algorithm and show how to analyze its running time. Arguably, these two results satisfy any intuitive expectation of resolving the problem. The basic approximate

monotone local search algorithm in inherently randomized, but it can be derandomized at the cost of a subexponential factor in the running time (Theorem 2.4). To attain statement (1), we show that the running time of approximate monotone local search is optimal, up to polynomial factors, independently of the computation of the running time itself. This lower bound proof uses a simple combinatorial argument that lower bounds the running time of *any* repurposing algorithm in terms of the (unknown) running time of approximate monotone local search.

To reach statement (2), we describe $\texttt{best}(\alpha, c, \beta)$ as the solution of a continuous, convex optimization problem, which allows us to evaluate $\texttt{best}(\alpha, c, \beta)$ up to any precision $\varepsilon > 0$ in time polynomial in the encoding length of α, c, β and ε using standard tools from convex optimization (see, e.g., [36]).

We show that $\texttt{best}(1, c, 1) = 2 - \frac{1}{c}$ and, more generally, $\texttt{best}(\beta, c, \beta) = \texttt{amls}_{\alpha=\beta}(\beta, c)$ which implies that previous algorithms [26, 31] already exploited existing algorithms in an optimal way in their respective restricted setting. These lower bounds are unconditional and do not rely on any complexity assumption such as the (Strong) Exponential-Time Hypothesis: the lower bounds are proved in a formal setting where our only problem-specific knowledge is being able to test the feasibility of a solution and invoke the α-approximation extension algorithm.

To further appreciate the running time of our algorithm, we mathematically compare $\texttt{best}(\alpha, c, \beta)$ to existing benchmarks.

Benchmark 1: Brute-Force for Exponential Approximation. A key feature of the bound $d = 2 - \frac{1}{c}$ obtained by Fomin et al. [31] is that it is always strictly better than the brute-force search running in time $\mathcal{O}^*(2^n)$. We extend this result to the approximate setting.

If our goal is to find a β-approximation for some $\beta > 1$, then the $\mathcal{O}^*(2^n)$ brute force search is certainly not optimal: for example, if $\beta = 2$ it suffices to only iterate over subsets of size at most $\frac{1}{3}n$ and at least $\frac{2}{3}n$, which only takes $\mathcal{O}^*(1.8899^n)$. This approach can be further optimized. Indeed, Esmer et al. [26] showed that for every monotone subset minimization problem, the classic brute-force approach can be generalized to a β-*approximation brute-force algorithm* running in time $\mathcal{O}^*(\texttt{brute}(\beta)^n)$, where $\texttt{brute}(\beta) := 1 + \exp\left(-\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right)\right)$ and $\mathcal{H}(\beta) := -\beta \ln \beta - (1 - \beta) \ln(1 - \beta)$ denotes the entropy function. Moreover, this running time is optimal if the family of the solution sets can only be accessed via a membership oracle. Note that $\texttt{brute}(1) = 2$, i.e., in the exact setting, this recovers the standard brute-force algorithm running in time $\mathcal{O}^*(2^n)$.

We compare approximate monotone local search to the β-approximation brute-force algorithm for every choice of $\alpha, c \geq 1$.

> For every fixed $\alpha, c \geq 1$ and $\beta > 1$, we have $\texttt{best}(\alpha, c, \beta) < \texttt{brute}(\beta)$:
> approximate monotone local search is strictly faster than what can be obtained by brute force.

In other words, our main finding is that repurposing an α-approximation algorithm always leads to a β-approximation algorithm strictly better than brute force except in the degenerate case $\alpha > \beta = 1$: an approximation algorithm cannot be used to obtain an exact algorithm better than the $\mathcal{O}^*(2^n)$ brute force. That is, somewhat counterintuitively, even a 1000-approximation algorithm running in time $\mathcal{O}^*(1000^k)$ is actually useful for obtaining an exponential-time 1.001-approximation algorithm better than brute force. Intuitively, brute force corresponds to the limit $c \to \infty$ and indeed approximate local search converges to brute force as c goes to ∞. This also implies that even if the parameterized extension algorithm is exact (i.e., $\alpha = 1$), a running time $\mathcal{O}^*(2^{\omega(k)})$ is not sufficient to obtain a β-approximation algorithm running in time $\mathcal{O}^*((\texttt{brute}(\beta) - \varepsilon)^n)$ for any fixed $\varepsilon > 0$.

Benchmark 2: AMLS with Equal Approximation Ratios. The results of [26] can also be used to derive an exponential-time β-approximation algorithm from a α-approximate parameterized extension algorithm with running time $\mathcal{O}^*(c^k)$, in case $\alpha \leq \beta$. This is done by interpreting the α-approximate parameterized extension algorithm as a β-approximate parameterized extension algorithm (which is correct as $\alpha \leq \beta$), therefore leading to an exponential-time β-approximation algorithm which runs in time $\mathcal{O}^*(d^n)$, where $d = \texttt{amls}_{\alpha=\beta}(\beta, c)$. Since $\texttt{amls}_{\alpha=\beta}(\beta, c) < \texttt{brute}(\beta)$ for all $\beta > 1$ and $c \geq 1$ (see [26]), this approach leads to a better than brute-force β-approximation for a wide

range of problems for which there is an exact (i.e., $\alpha = 1$) parameterized algorithm with running time $\mathcal{O}^*(c^k)$.

For example, the best known exact parameterized algorithm for ODD CYCLE TRAVERSAL runs in time $\mathcal{O}^*(2.3146^k)$ [55]. In particular, this algorithm is a parameterized 1.5-approximation algorithm for ODD CYCLE TRAVERSAL. Thus, using the result of [26] the algorithm can be used to derive an exponential time 1.5-approximation algorithm for ODD CYCLE TRAVERSAL which runs in time $\mathcal{O}^*(d^n)$ where $d = \texttt{amls}_{\alpha=\beta}(1.5, 2.3146) \approx 1.340 < \texttt{brute}(1.5) \approx 1.3849$. Intuitively, using the result of [26] in such a setting appears suboptimal. We confirm this intuition.

For every $\beta > \alpha \geq 1$ and every $c > 1$ it holds that $\texttt{best}(\alpha, c, \beta) < \texttt{amls}_{\alpha=\beta}(\beta, c)$.

Using Multiple Parameterized Approximation Algorithms. So far, all algorithms we described only use a single parameterized extension algorithm as a subroutine. However, since with our new results any α-approximate extension algorithm can be used to obtain a β-approximation algorithm, a natural extension is to use multiple α-approximate extension algorithms for different values of α and c at the same time. For example, FEEDBACK VERTEX SET can be solved exactly in time $\mathcal{O}^*(2.7^k)$ [49] (i.e., $\alpha_1 = 1$ and $c_1 = 2.7$) and admits a polynomial-time 2-approximation algorithm [4] (i.e., $\alpha_2 = 2$ and $c_2 = 1$). Instead of using only one of these subroutines to design an exponential approximation, it seems much more natural to allow an algorithm to rely on both subroutines together.

We extend all of our results to the setting where any finite number of parameterized extension subroutines may be used by a single approximation algorithm. Maybe surprisingly, this allows us to obtain further improvements over using only a single extension algorithm as a subroutine. That is, there are parameter settings where given two extension algorithms with (α_1, c_1) and (α_2, c_2), we can obtain a $\mathcal{O}^*(d^n)$ time β-approximation algorithm with d being *strictly smaller* than both $\texttt{best}(\alpha_1, c_1, \beta)$ and $\texttt{best}(\alpha_2, c_2, \beta)$. Unfortunately, we observe that, for many concrete problems, these improvements are small and often restricted to only a small range of approximation ratios.

Applications. Our results can be used to obtain exponential approximation algorithms for a wide range of problems. For many of these problems, there is no direct previous work on exponential-time approximations, thus our results serve as a baseline for future works. For problems, such as VERTEX COVER or FEEDBACK VERTEX SET, for which there are existing works on exponential approximations, our algorithms attain better running times than the state of art for the majority of approximation ratios.

The most natural application are deletion problems to hereditary graph classes, where the input is a graph G (which may be undirected or directed, and may contain labeled vertices), and we wish to delete the minimum number of vertices to ensure a certain hereditary property (i.e., the family of solution sets is closed under supersets). For example, we obtain exponential β-approximation algorithms for FVS, TOURNAMENT FVS, SUBSET FVS, d-HITTING SET , INTERVAL VERTEX DELETION, PROPER INTERVAL VERTEX DELETION, BLOCK GRAPH VERTEX DELETION, CLUSTER GRAPH VERTEX DELETION, SPLIT VERTEX DELETION, EDGE MULTICUT ON TREES, SUBSET DFVS, DOCT and MULTICUT.

To demonstrate the wide applicability, let us briefly discuss three illustrative examples here (a more thorough discussion of the applications to our results can be found in Section 3; also running times for all problems listed above and various approximation ratios β are listed in Appendix B).

- ODD CYCLE TRANSVERSAL has no constant-factor polynomial-time approximation under UGC [43], but it can be solved exactly in time $\mathcal{O}^*(2.3146^k)$ [55]. We obtain an exponential β-approximation algorithm for every $\beta > 1$, which significantly improves upon brute force. For example, we obtain a 1.1-approximation running in time $\mathcal{O}^*(1.3689^n)$ while $\texttt{brute}(1.1) \approx 1.7153$, and the 1.1-approximation obtained via Benchmark 2 runs in time $\mathcal{O}^*(1.4223^n)$.

- DIRECTED FEEDBACK VERTEX SET (DFVS) has no constant-factor polynomial-time approximation under UGC [38], and it is an open question to determine if DFVS can be solved exactly

in time $\mathcal{O}^*(c^k)$ for any constant (see, e.g., [16]). But DFVS has a parameterized 2-approximation algorithm running in time $\mathcal{O}^*(c^k)$ for some constant c [54]. Hence, we obtain an exponential β-approximation algorithm for every $\beta > 1$ that is faster than the β-approximation brute-force algorithm. For $1 < \beta < 2$, our algorithm is the first non-trivial β-approximation algorithm, and for $\beta > 2$ our algorithm improves over the previous best algorithm from [26]. For $\beta = 2$, our running time matches that of [26].

- PARTIAL VERTEX COVER has a polynomial-time 2-approximation [12], but is known to be W[1]-hard [37]. We obtain the first non-trivial exponential β-approximation algorithm for every $1 < \beta < 2$. For example, for $\beta = 1.1$, our algorithm runs in time $\mathcal{O}^*(1.6588^n)$.

2 Our Results

In this section, we discuss our results in five parts. We define the computational model in Section 2.1, and present the optimal algorithm as well as our the main results in Section 2.2. In Section 2.3 we describe the arguments to show that approximate monotone local search is the optimal way of repurposing existing parameterized approximation algorithms. After that, Section 2.4 deals with the computation of the running time, and Section 2.5 compares the running times to the benchmarks.

2.1 Computation Model

We state our results in an oracle-based computation model that properly reflects the setting described in the introduction. Let U be a universe of elements (i.e., a finite set). A *set system* of U is a family $\mathcal{F} \subseteq 2^U$ of subsets of U. We say the set system $\mathcal{F}$ is *monotone* if (i) $U \in \mathcal{F}$ and (ii) for every $S \subseteq T \subseteq U$, if $S \in \mathcal{F}$ then $T \in \mathcal{F}$. We consider minimization problems in which the objective is to find $S \in \mathcal{F}$ that minimizes $|S|$.

In the computation model, the universe U is given as part of the input to the algorithm. The set system $\mathcal{F}$, however, is not part of the input. Instead, the algorithm can implicitly access $\mathcal{F}$ using *extension oracles*.

Definition 2.1. *Let U be a finite universe, $\mathcal{F}$ be a set system of U and $\ell \in \mathbb{N}$. We say that $S \subseteq U$ is an ℓ-extension of $X \subseteq U$ if $X \cup S \in \mathcal{F}$ and $|S| \leq \ell$.*

Informally, a *random α-extension* oracle of a universe U and a monotone set system $\mathcal{F}$ gets $X \subseteq U$ and $\ell \in \mathbb{N}$ as an input, and returns a set $Y \subseteq U$ such such $X \cup Y \in \mathcal{F}$ and Y satisfies the following property with probability at least $\frac{1}{2}$:

> If there exists an ℓ-extension of X then Y is an $(\alpha \cdot \ell)$-extension of X.

Though intuitive, this definition does not properly define what kind of an object an oracle is, and considers an undefined probability space. These details will be important when proving lower bounds. We provide a formal definition of an extension oracle as a function that, in addition to X and ℓ, also receives a bit-string r as part of its argument. The bit string serves as the source of randomness for the oracle, and we assume the algorithm provides a random bit-string alongside each query.

Definition 2.2 (Random Extension Oracle). *Let U be a set and $\mathcal{F}$ be a monotone set system of U. A* random α-extension oracle *for U and $\mathcal{F}$ is a function $\mathtt{Ext}: 2^U \times \mathbb{N} \times \{0,1\}^m \to 2^U$ where $m \in \mathbb{N}$ that satisfies the following properties:*

1. $\mathtt{Ext}(X, \ell, r) \cup X \in \mathcal{F}$ for every $(X, \ell, r) \in 2^U \times \mathbb{N} \times \{0,1\}^m$, and

2. for every $(X, \ell) \in 2^U \times \mathbb{N}$ such that X has an ℓ-extension it holds that

$$\left| \left\{ r \in \{0,1\}^m \;\middle|\; \mathtt{Ext}(X, \ell, r) \text{ is an } \alpha \cdot \ell\text{-extension of } X \right\} \right| \geq \frac{1}{2} \cdot |\{0,1\}^m|.$$

If $m = 0$, then we say the oracle is deterministic.

An algorithm may have access to several extension oracles. We associate a *cost* $c \geq 1$ with each oracle, representing the cost incurred by quering the oracle. The *cost* of the oracle query (X, ℓ) is c^{ℓ}.[2]

---

[2]We commonly omit the third argument to the oracle.

Observe that invocations to an extension oracle with $\ell = 0$ are equivalent to membership queries, and hence extension oracles can be viewed as generalizations of membership oracles. The cost of a query (X, ℓ) represents the running time $\mathcal{O}^*(c^\ell)$ of a parameterized algorithm which emulates the oracle in our applications.

An *(oracle) specification list* is a non-empty and finite set $\mathcal{L} = \{(\alpha_1, c_1), \ldots, (\alpha_s, c_s)\}$ such that $\alpha_j, c_j \geq 1$ for every $j \in [s]$. We define a minimization problem for every oracle specification $\mathcal{L}$. An instance of the $\mathcal{L}$-*subset minimization problem* ($\mathcal{L}$-SUB) consists of a set U and a monotone set system $\mathcal{F}$ of U. The objective is to find a set $S \in \mathcal{F}$ such that $|S|$ is minimized. In our computational model the set U is given to the algorithm as part of the input. Furthermore, the algorithm has access to an α-extension oracle for U and $\mathcal{F}$, associated with cost c, for every $(\alpha, c) \in \mathcal{L}$ (that is, the algorithm is given $|\mathcal{L}|$ oracles for the same set system $\mathcal{F}$). In particular, $\mathcal{F}$ is part of the *instance*, but is not part of the *input*.

Let $\mathcal{A}$ be an algorithm for $\mathcal{L}$-SUB. The *cost of an execution* of $\mathcal{A}$ is the sum of costs over all oracle queries initiated by the algorithm, plus the number of computational operations conducted throughout the execution. That is, if $Q_{\alpha,c} \subseteq 2^U \times \mathbb{N}$ is the set of queries the algorithm makes to the α-extension oracle for every $(\alpha, c) \in \mathcal{L}$ in a specific execution and p is the number of computational operations, then the cost of the execution is $p + \sum_{(\alpha,c) \in \mathcal{L}} \sum_{(X,\ell) \in Q_{\alpha,c}} c^\ell$. We define $\mathtt{cost}_{\mathcal{A}}(n)$ to be the maximal cost of an execution of $\mathcal{A}$ given an input which satisfies $|U| \leq n$. We say $\mathcal{A}$ is of cost $f : \mathbb{N} \to \mathbb{N}$ if $\mathtt{cost}_{\mathcal{A}}(n) \leq f(n)$ for all $n \in \mathbb{N}$.

Following the standard notion of approximation algorithms, we say an algorithm $\mathcal{A}$ is a *(randomized) β-approximation* for $\mathcal{L}$-SUB if for every universe U, monotone set system $\mathcal{F}$ over U, and randomized extension oracles $\mathtt{Ext}_{\alpha,c}$ for every $(\alpha, c) \in \mathcal{L}$, the algorithm always returns $S \in \mathcal{F}$ (with probability 1) and it holds that $|S| \leq \beta \cdot \min_{T \in \mathcal{F}} |T|$ with probability at least $\frac{1}{2}$.

We also consider deterministic algorithms for $\mathcal{L}$-SUB. In this case we restrict our attention to inputs with deterministic oracles. Formally, we say an algorithm $\mathcal{A}$ is a *deterministic β-approximation* for $\mathcal{L}$-SUB if for every universe U, monotone set system $\mathcal{F}$ over U, and *deterministic* extension oracles $\mathtt{Ext}_{\alpha,c}$ for every $(\alpha, c) \in \mathcal{L}$, the algorithms returns $S \in \mathcal{F}$ such that $|S| \leq \beta \cdot \min_{T \in \mathcal{F}} |T|$.

For every specification list $\mathcal{L}$ and $\beta \geq 1$, we define $\mathtt{best}(\mathcal{L}, \beta)$ to be the base of the best cost β-approximation algorithm for $\mathcal{L}$-SUB. Formally,

$$\mathtt{best}(\mathcal{L}, \beta) = \inf \left\{ d \geq 1 \;\middle|\; \text{there is a } \beta\text{-approximation for } \mathcal{L}\text{-SUB with cost } d^n \cdot n^{\mathcal{O}(1)} \right\}. \quad (1)$$

The paper revolves around the value of $\mathtt{best}(\mathcal{L}, \beta)$. Our primary objectives are to attain an algorithm with cost $\mathcal{O}^*(\mathtt{best}((\mathcal{L}, \beta)^n))$, derive a method to compute $\mathtt{best}(\mathcal{L}, \beta)$, and analytically compare it to the benchmarks.

2.2 Approximate Monotone Local Search

Our first main result is that a simple monotone local search algorithm, $\mathtt{Approximate\text{-}MLS}_{\mathcal{L},\beta}$ (see Algorithm 2) is a β-approximation algorithm for $\mathcal{L}$-SUB with *optimal* cost of $\mathcal{O}^*((\mathtt{best}(\mathcal{L}, \beta))^n)$. The algorithm is a natural generalization of the monotone local search algorithms used in [26, 31].

The algorithm is based on a simple sampling procedure (Algorithm 1). Let $\mathtt{OPT} = \mathrm{argmin}_{S \in \mathcal{F}} |S|$ and assume $k = |\mathtt{OPT}|$. The sampling procedure sample a set $X \subseteq U$ of size t uniformly at random, and then extends the set to a solution $Z = X \cup Y$, where Y is attained via a query of the form (X, ℓ) to the α-extension oracle $\mathtt{Ext}_{\alpha,c}$. To keep the intuitive description simple we assume the oracle is deterministic.

The sampling produces a solution of size $\beta \cdot k$ assuming $|X \cap \mathtt{OPT}| \geq x$, for a carefully selected value x. Subject to this assumption, the set $\mathtt{OPT} \setminus X$ is a $(k - x)$-extension of X, thus, according to Definition 2.2, $Y = \mathtt{Ext}_{\alpha,c}(X, k - x)$ is an $\alpha(k - x)$-extension of X. It is therefore guaranteed that $X \cup Y \in \mathcal{F}$ and $|X \cup Y| \leq t + \alpha(k - x)$. As our objective is to find a set in $\mathcal{F}$ of cardinality at most $\beta \cdot k$, the value of x needs to satisfy $t + \alpha(k - x) \leq \beta k$. Indeed, we set $x = x_{\alpha,\beta}(k, t) := \left(1 - \frac{\beta}{\alpha}\right) k + \frac{t}{\alpha}$, which is the minimal value which satisfies $t + \alpha(k - x) \leq \beta k$.

The distribution of $|X \cap \mathtt{OPT}|$ is commonly referred as *hyper-geometric*. Define $\mathtt{hyper}(n, k, t, x)$ to be the probability that a uniformly random set X of t items out of $[n] := \{1, \ldots, n\}$ satisfies

$|X \cap [k]| \geq x$. Note that

$$\texttt{hyper}(n,k,t,x) = \sum_{y=\lceil x \rceil}^{\min\{t,k\}} \frac{\binom{k}{y}\binom{n-k}{t-y}}{\binom{n}{t}} = \sum_{y=\lceil x \rceil}^{\min\{t,k\}} \frac{\binom{t}{y}\binom{n-t}{k-y}}{\binom{n}{k}}. \tag{2}$$

It follows that the sampling procedure returns a solution of cardinality βk or less with probability (at least) $\texttt{hyper}\,(n,k,t,x_{\alpha,\beta}(k,t))$. Furthermore, the cost of the procedure is $\mathcal{O}\left(c^{k-\lceil x_{\alpha,\beta}(k,t)\rceil}\right) = \mathcal{O}\left(c^{\frac{\beta k - t}{\alpha}}\right)$. Thus, to obtain a constant success probability the sampling procedure has to be executed $\approx (\texttt{hyper}\,(n,k,t,x_{\alpha,\beta}(k,t)))^{-1}$ times, leading to a total cost of

$$\mathcal{O}\left(\frac{c^{\frac{\beta \cdot k - t}{\alpha}}}{\texttt{hyper}\,(n,k,t,x_{\alpha,\beta}(k,t))}\right). \tag{3}$$

The sampling procedure is used by Algorithm 2. This algorithm iterates over all possible values of $k = |\texttt{OPT}|$. For each value of k the algorithm selects an oracle $(\alpha,c) \in \mathcal{L}$ and $t \in \mathbb{N}$ which minimizes (3) (Line 3 of Algorithm 2) and then invokes the sampling procedure sufficiently many times to attain success probability of $\frac{1}{2}$. The range of values t can take is restricted to $\left[M^*_{\alpha,\beta} \cdot k, \ \beta \cdot k\right] \cap \mathbb{N}$, where

$$M^*_{\alpha,\beta} := \begin{cases} 0 & \text{if } \alpha \leq \beta \\ \frac{\alpha - \beta}{\alpha - 1} & \text{if } \alpha > \beta \end{cases} \tag{4}$$

for all $\alpha, \beta \geq 1$. This restriction ensures the algorithm only considers values of t for which $x_{\alpha,\beta}(k,t) \leq t$.

Algorithm 1 $\texttt{Sample}(U,k,t,\alpha,\beta,\texttt{Ext}_{\alpha,c})$
___
Input: A universe U , $k \in \mathbb{N}$, $t \in \mathbb{N}$, $\alpha, \beta \geq 1$ and an α-extension oracle $\texttt{Ext}_{\alpha,c}$.
1: Sample a set X of size t from U uniformly at random.
2: $Y \leftarrow \texttt{Ext}_{\alpha,c}\left(X, k - \left\lceil \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha} \right\rceil\right)$.
3: Return $Z \leftarrow X \cup Y$.

Algorithm 2 $\texttt{Approximate-MLS}_{\mathcal{L},\beta}$
___
Input: A universe U and an extension oracle $\texttt{Ext}_{\alpha,c}$ for every $(\alpha,c) \in \mathcal{L}$
1: $\mathcal{S} \leftarrow \emptyset$, $n \leftarrow |U|$.
2: **for** k from 0 to $\frac{n}{\beta}$ **do**
3:   Find $(\alpha,c) \in \mathcal{L}$ and $t \in \left[M^*_{\alpha,\beta} \cdot k, \beta \cdot k\right] \cap \mathbb{N}$ which minimize $\left(\frac{c^{\frac{\beta k - t}{\alpha}}}{\texttt{hyper}\left(n,k,t,\left(1-\frac{\beta}{\alpha}\right)\cdot k + \frac{t}{\alpha}\right)}\right)$.
4:   Run $\mathcal{S} \leftarrow \mathcal{S} \cup \{\texttt{Sample}(U,k,t,\alpha,\beta,\texttt{Ext}_{\alpha,c})\}$ for $2 \cdot \left\lceil \left(\texttt{hyper}\left(n,k,t,(1-\frac{\beta}{\alpha})\cdot k + \frac{t}{\alpha}\right)\right)^{-1}\right\rceil$ times.

5: **Return** a minimum-sized set in $\mathcal{S}$.

Our main theorem asserts that $\texttt{Approximate-MLS}_{\mathcal{L},\beta}$ has the best possible cost of a β-approximation for $\mathcal{L}$-SUB.

Theorem 2.3 (Main result: randomized algorithm). *For every specification list $\mathcal{L}$ and $\beta \geq 1$,* $\texttt{Approximate-MLS}_{\mathcal{L},\beta}$ *is a randomized β-approximation for $\mathcal{L}$-SUB of cost $n^{\mathcal{O}(1)} \cdot (\texttt{best}(\mathcal{L},\beta))^n$.*

Similar to [26, 31], it is possible to derandomize Algorithm 2 with sub-exponential overhead in the running time. The derandomized version of the algorithm, $\texttt{Deterministic-Approximate-MLS}_{\mathcal{L},\beta}$ (Algorithm 3), is given in Section 4.

Theorem 2.4 (Main result: deterministic algorithm). *For every specification list $\mathcal{L}$ and $\beta \geq 1$,* $\texttt{Deterministic-Approximate-MLS}_{\mathcal{L},\beta}$ *(Algorithm 3) is a β-approximation for $\mathcal{L}$-SUB with cost at most* $(\texttt{best}(\mathcal{L},\beta))^n \cdot 2^{o(n)}$.

Though Theorem 2.3 states the cost of Algorithm 2 is the best possible, it does not provide any method by which this cost can be computed. The next theorem addresses this issue.

Theorem 2.5 (**Main result: computing** `best`)**.** *There is an algorithm which given $\beta \geq 1$, a specification list $\mathcal{L}$ and $\varepsilon > 0$ computes $\mathtt{best}(\mathcal{L}, \beta)$ up to additive precision of ε, and runs in polynomial time in the encoding length of $\mathcal{L}$, β and ε.*

2.3 Optimality of the Algorithm

A naive calculation reveals that the cost of Algorithm 2 can be bounded by the function $f_{\mathcal{L}, \beta}$ defined by

$$f_{\mathcal{L},\beta}(n) := \max_{k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}} \quad \min_{(\alpha, c) \in \mathcal{L}} \quad \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta \cdot k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)}. \tag{5}$$

Lemma 2.6. *For every $\beta \geq 1$ and specification list $\mathcal{L}$, it holds that $\mathtt{Approximate\text{-}MLS}_{\mathcal{L},\beta}$ is a β-approximation algorithm for $\mathcal{L}$-SUB with cost at most $n^{\mathcal{O}(1)} \cdot f_{\mathcal{L},\beta}(n)$.*

The proof of Lemma 2.6 is given in Section 4. The same section also proves a variant of Lemma 2.6 which refers to $\mathtt{Deterministic\text{-}Approximate\text{-}MLS}_{\mathcal{L},\beta}$ (Algorithm 3).

Lemma 2.7. *For every $\beta \geq 1$ and specification list $\mathcal{L}$, Algorithm 3 is a deterministic β-approximation for $\mathcal{L}$-SUB with cost at most $f_{\mathcal{L},\beta}(n) \cdot 2^{o(n)}$.*

Maybe surprisingly, one of the main insights in this paper is that these simple algorithms are actually optimal in the oracle model defined above, i.e., we can also use $f_{\mathcal{L},\beta}$ as a lower bound on the cost of any algorithm for $\mathcal{L}$-SUB.

Lemma 2.8. *For any $\beta \geq 1$ and specification list $\mathcal{L}$, every β-approximation algorithm for $\mathcal{L}$-SUB has cost at least $n^{-\mathcal{O}(1)} \cdot f_{\mathcal{L},\beta}(n)$.*

The proof of Lemma 2.8, given in Section 5, follows from the inability of an algorithm for $\mathcal{L}$-SUB to distinguish between instances in which $\mathcal{F}$ contains all sets of size at least $\beta \cdot k + 1$, versus instances in which $\mathcal{F}$ contains a set R of cardinality k, its supersets, and all sets of size at least $\beta \cdot k + 1$. Returning a valid solution for the later requires the algorithm to initiate an oracle query of the form (X, ℓ) to an α-extension oracle such that $|X| + \alpha\ell \leq \beta k$ and X has an ℓ-extension. As we select R to be a random set, we can use this property to lower bound the total cost of the queries the algorithm must initiate in order find an ℓ-extension with a constant probability. The value of k used in the construction is the value which attains the maximum in (5). We note the oracles used in the proof of Lemma 2.8 are deterministic. Hence, the lower bound holds even if the algorithm is guaranteed the oracles are deterministic. Together Lemmas 2.6 and 2.8 indicate that $\mathtt{Approximate\text{-}MLS}_{\mathcal{L},\beta}$ attains the best possible cost of a β-approximation algorithm for $\mathcal{L}$-SUB, up to polynomial factors. Similarly, Lemmas 2.7 and 2.8 imply that Algorithm 3 is optimal up to sub-exponential factors.

It follows from Lemmas 2.6 and 2.8 that

$$\mathtt{best}(\mathcal{L}, \beta) = \lim_{n \to \infty} (f_{\mathcal{L},\beta}(n))^{\frac{1}{n}} \tag{6}$$

for all specification lists $\mathcal{L}$ and $\beta \geq 1$. We note that the above limit does not imply Theorem 2.3, though it can be used to establish as slightly weaker claim. Hypothetically, it is possible that $f_{\mathcal{L},\beta}(n) = 2^{n+\sqrt{n}}$ and thus, $\mathtt{best}(\mathcal{L}, \beta) = 2$ but $f_{\mathcal{L},\beta}(n) \neq \mathcal{O}^*(2^n)$. We will later rule out the existence of such cases.

The proof of Lemma 2.8 can also be adapted to the exact setting of [30]. Given a set U and a subset family $\mathcal{F}$ of U (not necessarily monotone), an *exact extension oracle* for $\mathcal{F}$ takes as an input a set $X \subseteq U$ and $\ell \in \mathbb{N}$. The oracle either returns YES or NO. If X has an ℓ-extension then the oracle returns YES with probability at least $\frac{1}{2}$. If X does not have an ℓ-extension then the oracle returns NO. Similarly to the approximate case, we associate a number $c \geq 1$ with the oracle. The *cost* of an oracle query is c^ℓ.

In the *c-decision problem* (*c*-DEC) the input is a universe U and an exact extension oracle for a set family $\mathcal{F}$ of U. The objective is to determine if $\mathcal{F} \neq \emptyset$ (in particular, the set system does not have to be monotone). The execution cost of an algorithm for *c*-DEC is the sum of costs of all oracle queries plus the number of computational operations, where the cost of a query (X, ℓ) is c^ℓ. Similarly to $\mathcal{L}$-SUB, we say that an algorithm $\mathcal{A}$ for *c*-DEC is of cost $f : \mathbb{N} \to \mathbb{N}$ if every execution of $\mathcal{A}$ with an input for which $|U| \leq n$ has cost at most $f(n)$. In [31] it was shown that there is a randomized algorithm for *c*-DEC of cost $n^{\mathcal{O}(1)} \cdot \left(2 - \frac{1}{c}\right)^n$.

Using the same ideas as in the proof of Lemma 2.8 we can show the following.

Lemma 2.9. *For every $c > 1$, every randomized algorithm for c-DEC has cost of at least $n^{-\mathcal{O}(1)} \cdot f_{\{(1,c)\},1}(n)$.*

As we can also show that $f_{\{(1,c)\},1}(n) \geq n^{-\mathcal{O}(1)} \cdot \left(2 - \frac{1}{c}\right)^n$, we obtain the following theorem.

Theorem 2.10. *For all $c > 1$, every randomized algorithm for c-DEC has cost of at least $n^{-\mathcal{O}(1)} \cdot \left(2 - \frac{1}{c}\right)^n$.*

In particular, Theorem 2.10 indicates the result of [31] cannot be improved. The proofs of Lemma 2.9 and Theorem 2.10 are given in Section 5.

2.4 Evaluating the Running Time of Approximate Monotone Local Search

So far, we showed that $\texttt{Approximate-MLS}_{\mathcal{L},\beta}$ attains the best possible cost of a β-approximation for $\mathcal{L}$-SUB, up to polynomial factors. However, the tools presented so far do not provide a method for evaluating the running time of the algorithm, and do not suffice to show Theorems 2.3 and 2.4.

The proof of Theorem 2.5, which shows $\texttt{best}(\mathcal{L}, \beta)$ can be computed efficiently, consists of two main stages. The first stage shows that $f_{\mathcal{L},\beta} \approx d^n$ (ignoring polynomial factors), where d is a solution for a continuous max-min optimization problem. As a by product, the stage provides the missing ingredient towards the proofs of Theorems 2.3 and 2.4. The second stage shows the minimization part of the optimization problem is a minimization of a convex function, and the maximization part is a maximization of a concave function. Hence, both parts of the optimization problem can be easily solved using known tools from convex optimization (see, e.g., [36]).

Using the standard $\binom{n}{k} \approx \exp\left(n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right)$ estimation for binomial coefficients and basic analysis of the hyper-geometric distribution $\texttt{hyper}$, the discrete optimization problem defined in (5) can be converted to a continuous optimization problem. For every $\alpha, \beta, c \geq 1$ we define the following functions.

$$\delta_{\alpha,\beta}(\kappa, \tau) = \begin{cases} \frac{\frac{\beta}{\alpha}\kappa - \frac{\tau}{\alpha}}{1-\tau} = \frac{\frac{\beta}{\alpha}\kappa - \frac{1}{\alpha}}{1-\tau} + \frac{1}{\alpha} & \text{if } \tau \neq 1 \\ \frac{1}{\alpha} & \text{if } \tau = 1 \end{cases} \tag{7}$$

$$\gamma_{\alpha,\beta}(\kappa, \tau) = \begin{cases} \left(1 - \frac{\beta}{\alpha}\right)\frac{\kappa}{\tau} + \frac{1}{\alpha} & \text{if } \tau \neq 0 \\ \frac{1}{\alpha} & \text{if } \tau = 0 \end{cases} \tag{8}$$

$$g_{\alpha,\beta,c}(\kappa, \tau) = \frac{\beta\kappa - \tau}{\alpha} \ln c - \tau \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\kappa, \tau)\right) - (1 - \tau) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\kappa, \tau)\right) + \mathcal{H}(\kappa) \tag{9}$$

$$M_{\alpha,\beta}(\kappa) = \begin{cases} \frac{\beta - \alpha}{1 - \alpha \cdot \kappa} \cdot \kappa & \text{if } \alpha < \beta \\ 0 & \text{if } \alpha = \beta \\ \frac{\alpha - \beta}{\alpha - 1} \cdot \kappa & \text{if } \alpha > \beta \end{cases} \tag{10}$$

Observe that $M_{\alpha,\beta}(\kappa) = M^*_{\alpha,\beta} \cdot \kappa$ ($M^*_{\alpha,\beta}$ is defined in (4)) if $\alpha \geq \beta$, but $M_{\alpha,\beta}(\kappa) \neq M^*_{\alpha,\beta} \cdot \kappa$ if $\alpha < \beta$. We follow the standard notation in which $0 \ln 0 = 0$ and $\mathcal{H}(0) = \mathcal{H}(1) = 0$. For every $\beta \geq 1$ and specification list $\mathcal{L}$ we define

$$\texttt{amls}(\mathcal{L}, \beta) = \exp\left(\max_{0 \leq \kappa \leq \frac{1}{\beta}} \min_{(\alpha,c) \in \mathcal{L}} \min_{M_{\alpha,\beta}(\kappa) \leq \tau \leq \beta\kappa} g_{\alpha,\beta,c}(\kappa, \tau)\right). \tag{11}$$

If $\mathcal{L} = \{(\alpha, c)\}$, we also write $\texttt{amls}(\alpha, c, \beta)$ instead of $\texttt{amls}(\{(\alpha, c)\}, \beta)$.

Lemma 2.11. *For every $\beta \geq 1$ and specification list $\mathcal{L}$ it holds that*

$$n^{-\mathcal{O}(1)} \cdot (\mathtt{amls}(\mathcal{L}, \beta))^n \;\leq\; f_{\mathcal{L}, \beta}(n) \;\leq\; n^{\mathcal{O}(1)} \cdot (\mathtt{amls}(\mathcal{L}, \beta))^n .$$

Note that the constants represented by $\mathcal{O}(1)$ in Lemma 2.11 may depend on $\mathcal{L}$ and β. The following corollary is an immediate consequence of Lemma 2.11 and (6).

Corollary 2.12. *For every specification list $\mathcal{L}$ and $\beta \geq 1$ it holds that $\mathtt{amls}(\mathcal{L}, \beta) = \mathtt{best}(\mathcal{L}, \beta)$.*

Theorem 2.3 (Theorem 2.4) follows from Lemma 2.6 (Lemma 2.7), Lemma 2.11 and Corollary 2.12.

Now, our next challenge is to show that $\mathtt{amls}$ can be computed. Our first observation towards this goal is that $g_{\alpha,\beta,c}(\kappa, \tau)$ is *convex* as a function of τ for every fixed κ.

Lemma 2.13. *Let $\alpha, c \geq 1$, $\beta > 1$ and $0 < \kappa < \frac{1}{\beta}$. The function $h(\tau) = g_{\alpha,\beta,c}(\kappa, \tau)$ is convex in the domain $[M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ and $\min_{\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]} g_{\alpha,\beta,c}(\kappa, \tau) < h(\beta\kappa) = g_{\alpha,\beta,c}(\kappa, \beta\kappa)$.*

The lemma follows from a standard calculus argument. In fact, we are able to show a slightly stronger claim, which states that, up to some corner cases, the minimum of $h(\tau)$ (as defined in Lemma 2.13) in the interval $[M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ is an interior point (that is, not $M_{\alpha,\beta}(\kappa)$ or $\beta\kappa$). One of the corner cases occurs when $\alpha = \beta$, in which the minimum may be at $\tau = 0 = M_{\alpha,\alpha}(\kappa)$. This distinction provides some evidence that the analysis inevitably has to differ from the approaches taken in [26, 31] which deal with the special case of $\alpha = \beta$.

For any $\alpha, \beta, c \geq 1$ and $0 \leq \kappa \leq \frac{1}{\beta}$ we define

$$g^*_{\alpha,\beta,c}(\kappa) = \min_{M_{\alpha,\beta}(\kappa) \leq \tau \leq \beta\kappa} g_{\alpha,\beta,c}(\kappa, \tau). \tag{12}$$

Therefore,

$$\mathtt{amls}(\mathcal{L}, \beta) = \exp\left(\max_{0 \leq \kappa \leq \frac{1}{\beta}} \min_{(\alpha,c) \in \mathcal{L}} g^*_{\alpha,\beta,c}(\kappa) \right). \tag{13}$$

By Lemma 2.13, $g^*_{\alpha,\beta,c}(\kappa)$ is the solution for a convex minimization over a closed interval. Observe that $g_{\alpha,\beta,c}(\kappa, \tau)$ can be evaluated up to additive precision of ε in polynomial time in the encoding length of α, β, κ, τ, κ and ε. Thus, using standard convex optimization tools (e.g., [36, Theorem 4.3.13]) we attain the following result.

Corollary 2.14. *There exists an algorithm which, given $\alpha, \beta, c \geq 1$, $0 \leq \kappa \leq \frac{1}{\beta}$ and $\varepsilon > 0$, computes $g^*_{\alpha,\beta,c}(\kappa)$ up to an additive precision of $\varepsilon > 0$, and runs in polynomial time in the encoding length of α, β, c, κ and ε.*

The main insight behind the proof of Theorem 2.5 is the following.

Lemma 2.15. *For all $\alpha, c \geq 1$ and $\beta > 1$, it holds that $g^*_{\alpha,\beta,c}(\kappa)$ is concave in the interval $\left[0, \frac{1}{\beta}\right]$.*

To prove Lemma 2.15 we show that if (κ, τ) is a critical point of $g_{\alpha,\beta,c}$ then the determinant of the Hessian matrix of $g_{\alpha,\beta,c}$ at (κ, τ) is negative. Once this argument is established, the lemma follows quite easily. Since the minimum of concave functions is also a concave function, Lemma 2.15 immediately implies the following.

Corollary 2.16. *For every specification list $\mathcal{L}$ and $\beta \geq 1$, the function $h(\kappa) = \min_{(\alpha,c) \in \mathcal{L}} g^*_{\alpha,\beta,c}(\kappa)$ is concave on the interval $\left[0, \frac{1}{\beta}\right]$.*

By Corollary 2.16 it follows that $\mathtt{amls}(\mathcal{L}, \beta)$ (13) is the maximum of a concave function on a closed interval. Furthermore, by Corollary 2.14 it holds that the function being maximized can be computed, up to an additive error of ε, in polynomial time. Thus, using convex optimization once more (e.g., [36, Theorem 4.3.13]), we get the following.

Corollary 2.17. *There is an algorithm which, given a specification list $\mathcal{L}$, $\beta \geq 1$ and $\varepsilon > 0$, computes $\mathtt{amls}(\mathcal{L}, \beta)$ up to an additive error of ε in time polynomial in the encoding length of $\mathcal{L}$, β and ε.*

Theorem 2.5 immediately follows from Lemma 2.11 and Corollary 2.17. In particular, to prove Theorem 2.5 we are left to provide proofs for Lemmas 2.11, 2.13 and 2.15. The proof of Lemma 2.11 is given in Section 6, and the proofs of Lemmas 2.13 and 2.15 are given in Section 7. We note that our computations of specific values of $\texttt{best}(\mathcal{L}, \beta)$ do not implement the theoretical algorithm from [36]. Instead, we use a combination of Golden Section Search [60, Section 10.2] and a simple binary search which finds the root of the derivative. We use $\texttt{mpmath}$ [58] for high-precision arithmetics.

2.5 Comparisons

Since $\texttt{best}(\mathcal{L}, \beta) = \texttt{amls}(\mathcal{L}, \beta)$, we can use the definition of $\texttt{amls}(\mathcal{L}, \beta)$ as the optimum of an optimization problem (11) to compare its value to Benchmarks 1 and 2. We first compare $\texttt{best}$ to $\texttt{brute}$.

Theorem 2.18. *For every $\beta > 1$ and specification list $\mathcal{L}$ it holds that $\texttt{best}(\mathcal{L}, \beta) < \texttt{brute}(\beta)$. Moreover, $\lim_{c \to \infty} \texttt{best}(\alpha, c, \beta) = \texttt{brute}(\beta)$ for every $\alpha \geq 1$ and $\beta > 1$.*

The proof of Theorem 2.18 is given in Section 8.

In [26] the authors showed that $\texttt{Approximate-MLS}_{\{(\beta,c)\},\beta}$ is a β-approximation algorithm for $\{(\beta, c)\}$-SUB of cost $\mathcal{O}^*((\texttt{amls}_{\alpha=\beta}(\beta, c))^n)$, where $\texttt{amls}_{\alpha=\beta}(\beta, c)$ is the unique value $d \in \left(1, 1 + \frac{c-1}{\beta}\right)$ which satisfies $\mathcal{D}\left(\frac{1}{\beta} \middle\| \frac{d-1}{c-1}\right) = \frac{\ln c}{\beta}$, for every $\beta, c > 1$.[3] The next lemma, which we prove in Section Section 7, also implies that the analysis in [26] is tight.

Lemma 2.19. *For every $\beta, c > 1$ it holds that $\texttt{best}(\beta, c, \beta) = \texttt{amls}_{\alpha=\beta}(\beta, c)$.*

Let $\beta > \alpha \geq 1$ and $c \geq 1$. Since an α-extension oracle is also a β-extension oracle, the result of [26] can be used to obtain a β-approximation algorithm for $\{(\alpha, c)\}$-SUB by executing the β-approximation algorithm for $\{(\beta, c)\}$-SUB whose running time is $\texttt{amls}_{\alpha=\beta}(\beta, c) = \texttt{best}(\beta, c, \beta)$. This approach, which views an α-extension oracle as a special case of β-extension oracle intuitively seems suboptimal. The next lemma, proven in Section 9, confirms this intuition.

Lemma 2.20. *For every $\beta > \alpha \geq 1$ and every $c > 1$ it holds that $\texttt{best}(\alpha, c, \beta) < \texttt{amls}_{\alpha=\beta}(\beta, c)$.*

3 Applications

In this section, we demonstrate how our results can be used to obtain exponential approximation algorithms for a wide range of problems. All the problems considered in this section are defined in Appendix A. Moreover, extensive data sets providing the running times of the obtained algorithms are provided in Appendix B.

3.1 Combining Exact FPT and Polynomial-Time Approximation Algorithms

The most common application of our results is to problems that admit a single-exponential FPT algorithm and/or a constant-factor approximation algorithm. Indeed, both types of algorithms have been intensively studied in the literature (see, e.g., [20, 63]), and there is an abundance of problems admitting single-exponential fpt algorithms and/or constant-factor approximation algorithms which we can use to obtain exponential approximation algorithms. Actually, from the view point of applications, this is a key advantage over the previous work [26] that requires a parameterized β-approximation algorithm, since such algorithms are still somewhat rare.

A large class of problems, many of which fall into this category, are deletion problems to some graph property Π.

Π VERTEX DELETION
Input: An (undirected or directed) graph G.
Question: Find a minimum set S of vertices of G such that $G - S \in \Pi$.

---

[3]$\mathcal{D}(a\|b) = a \ln \frac{a}{b} + (1-a) \ln \frac{1-a}{1-b}$ is the Kullback-Leibler divergence between two Bernoulli distributions with parameters a and b.

(α, c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta, 2.69998)$	1.465	1.3861	1.3331	1.294	1.2637	1.2393	1.2193	1.2024	1.188
$(1.0, 2.69998)$	1.4156	1.3289	1.2753	1.2378	1.2099	1.1881	1.1706	1.1561	1.144
$(2.0, 1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298
combined	1.4156	1.3289	1.2753	1.2378	1.2068	1.1507	1.1037	1.064	1.0298

Tournament Feedback Vertex Set

(α, c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta, 1.618)$	1.2912	1.2463	1.2152	1.1918	1.1734	1.1583	1.1458	1.1352	1.126
$(1.0, 1.618)$	1.2348	1.1837	1.1531	1.132	1.1164	1.1042	1.0945	1.0865	1.0798
$(2.0, 1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298
combined	1.2348	1.1837	1.1531	1.132	1.1164	1.1042	1.0945	1.064	1.0298

Table 2: Running times for Feedback Vertex Set and Tournament Feedback Vertex Set. An entry at in row (α, c) and column β is $\texttt{best}(\alpha, c, \beta)$. The last row contains $\texttt{best}(\mathcal{L}_{\mathrm{FVS}}, \beta)$ and $\texttt{best}(\mathcal{L}_{\mathrm{TFVS}}, \beta)$, respectively.

This type problem can be translated into our framework by setting $U := V(G)$ to be the set of vertices of G, and the task is to find a minimum set in the set system $\mathcal{F} := \{S \subseteq U \mid G - S \in \Pi\}$. If Π is a hereditary graph property (i.e., it is closed under subgraphs), the set system $\mathcal{F}$ is monotone which allows us to apply the algorithmic tools described in Section 2. As two illustrative examples, let us consider the Feedback Vertex Set (FVS) problem (over undirected graphs), which corresponds to Π being the class of forests, and the Tourament Feedback Vertex Set (Tournament FVS) where the input is a tournament graph, and Π contains all acyclic tournaments.

The best (randomized) parameterized algorithm for FVS has been obtained by Li and Nederlof [49] and runs in time $\mathcal{O}^*(2.69998^k)$. Moreover, FVS admits a polynomial-time 2-approximation algorithm [4]. The first algorithm provides a 1-extension oracle with cost 2.69998, and second algorithm implements a 2-extension oracle with cost 1. Together, we obtain an oracle specification list $\mathcal{L}_{\mathrm{FVS}} := \{(1, 2.69998), (2, 1)\}$. Using Theorem 2.3, we can use approximate monotone local search to obtain a β-approximation algorithm for FVS running in time $\mathcal{O}^*((\texttt{best}(\mathcal{L}_{\mathrm{FVS}}, \beta))^n)$ for every $\beta \geq 1$.

Similarly, Tournament Feedback Vertex Set can be solved in time $\mathcal{O}^*(1.618^k)$ [46] and admits polynomial-time 2-approximation algorithm [53], which gives rise to the oracle specification list $\mathcal{L}_{\mathrm{TFVS}} := \{(1, 1.618), (2, 1)\}$. Hence, we obtain a β-approximation algorithm for Tournament FVS running in time $\mathcal{O}^*((\texttt{best}(\mathcal{L}_{\mathrm{TFVS}}, \beta))^n)$.

We provide the values of $\texttt{best}(\mathcal{L}_{\mathrm{FVS}}, \beta)$ and $\texttt{best}(\mathcal{L}_{\mathrm{TFVS}}, \beta)$ for selected approximation ratios β in Table 2, and give a graphical visualization in Figure 1. We also compare $\texttt{best}(\mathcal{L}_{\mathrm{FVS}}, \beta)$ and $\texttt{best}(\mathcal{L}_{\mathrm{TFVS}}, \beta)$ with the running times of several other algorithms. As the most basic benchmark, we compare the running times to the brute-force search as described in [26] (see also Benchmark 1). Also, by interpreting an exact single-exponential fpt algorithm as a β-approximation algorithm, we can use Approximate Monotone Local Search for $\alpha = \beta$ [26] as a second benchmark (see also Benchmark 2). It can be observed that $\texttt{best}(\mathcal{L}_{\mathrm{FVS}}, \beta)$ and $\texttt{best}(\mathcal{L}_{\mathrm{TFVS}}, \beta)$ are strictly better than both of these algorithms for all $\beta > 1$ (see also Lemma 2.20 and Theorem 2.18). We remark that another exponential β-approximation algorithm for FVS has been obtained in [25].[4] However, this algorithm is slower than the brute-force β-approximation algorithm described in [26], and thus, our algorithm is also significantly faster than the algorithm from [25].

As a further comparison, we also consider the running of our algorithm when only a single oracle is used. More precisely, let us define $\mathcal{L}'_{\mathrm{FVS}} := \{(1, 2.69998)\}$ and $\mathcal{L}''_{\mathrm{FVS}} := \{(2, 1)\}$. Clearly, $\texttt{best}(\mathcal{L}_{\mathrm{FVS}}, \beta) \leq \min(\texttt{best}(\mathcal{L}'_{\mathrm{FVS}}, \beta), \texttt{best}(\mathcal{L}''_{\mathrm{FVS}}, \beta))$ for all $\beta \geq 1$. Interestingly, this inequality is strict for some values of β. Indeed, while this may not be visible from Figure 1, one can observe from Figure 2 that using both oracles together leads to a better running for β roughly in the range $[1.481, 1.507]$. However, it can also be observed that the improvement obtained this way is rather

---

[4]The running time of this algorithm is not correctly stated in [25, Theorem 3.1]; their β-approximation algorithm runs in time $\mathcal{O}^*(d^n)$ where $d \geq 1$ is the unique solution to the equation $1 = d^{-1} + d^{-\beta}$.
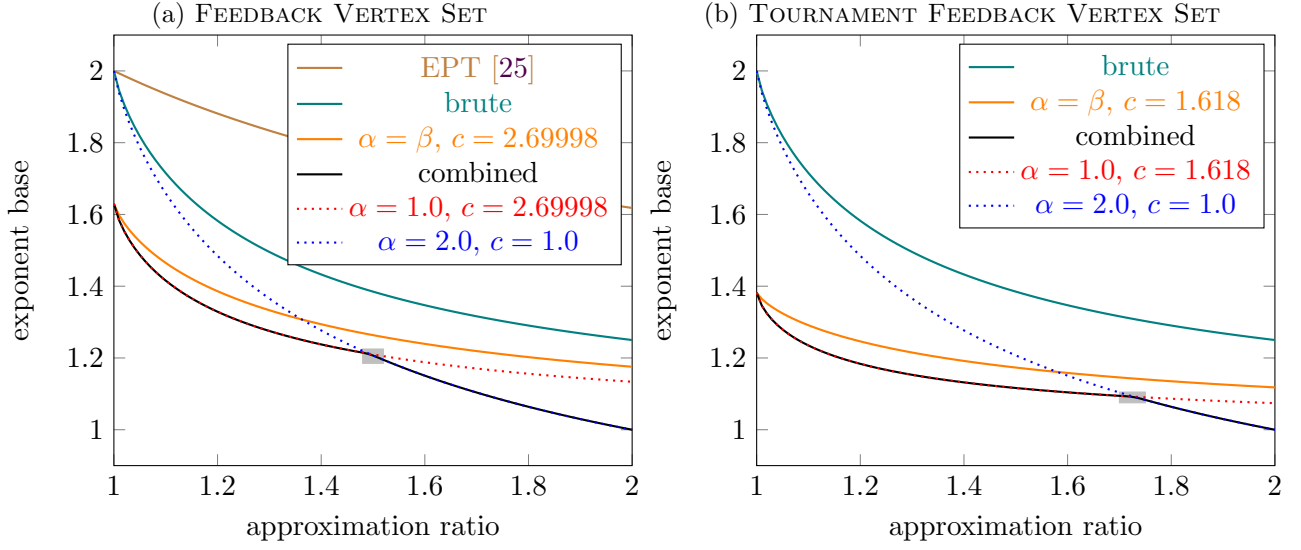
Figure 1: Results for FEEDBACK VERTEX SET and TOURNAMENT FEEDBACK VERTEX SET. A dot at (β, d) means that the respective algorithm outputs an β-approximation in time $\mathcal{O}^*(d^n)$. Figure 2 zooms into the gray regions.
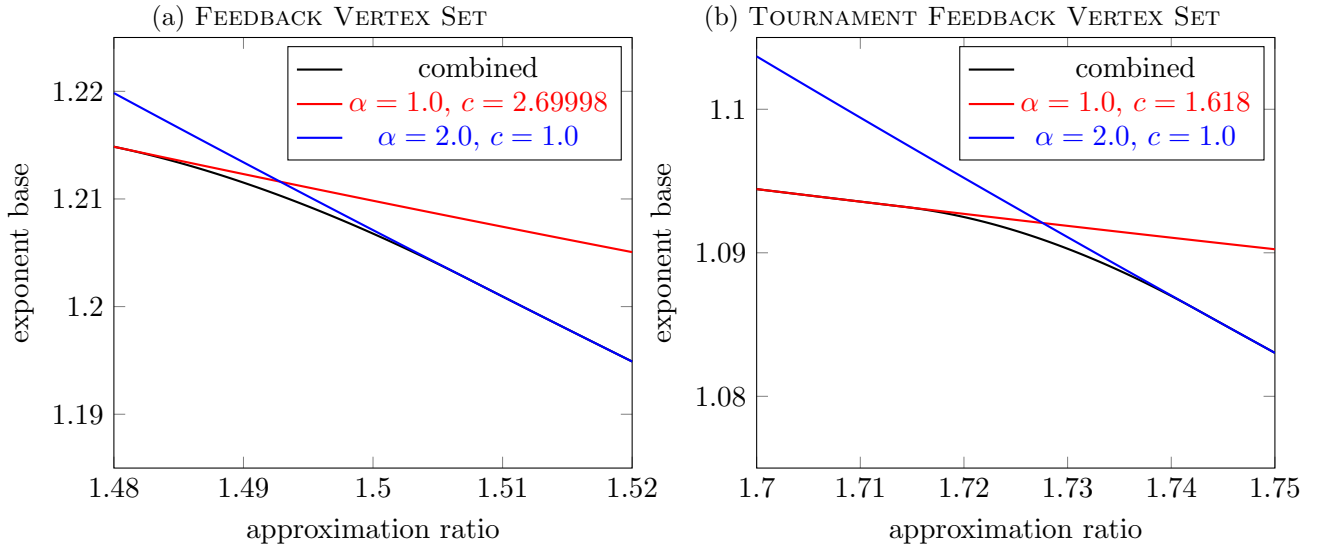

Figure 2: Results for FEEDBACK VERTEX SET and TOURNAMENT FEEDBACK VERTEX SET. A dot at (β, d) means that the respective algorithm outputs an β-approximation in time $\mathcal{O}^*(d^n)$.

Problem	c_1		det.	α_2		det.
FVS	2.69998	[49]	✗	2	[4]	✓
Tournament FVS	1.618	[46]	✓	2	[53]	✗
Subset FVS	4.0	[39]	✓	8	[27]	✓
d-Hitting Set ($d \geq 3$)	$(d - 0.9245)$	[30]	✓	d	[6]	✓
Interval Vertex Deletion	8.0	[13]	✓	8	[13]	✓
Proper Interval Vertex Deletion	6.0	[62]	✓	6	[62]	✓
Block Graph Vertex Deletion	4.0	[1]	✓	4	[1]	✓
Cluster Graph Vertex Deletion	1.9102	[8]	✓	2	[2]	✓
Cograph Vertex Deletion	3.0755	[30]	✓	4		✓
Split Vertex Deletion	2.0	[35]	✓	$2 + \epsilon$	[23]	✓
Edge Multicut on Trees	1.5538	[40]	✓	2	[33]	✓

Table 3: List of deletion problems admitting an single-exponential parameterized algorithm running in time $\mathcal{O}^*(c_1^k)$ and a polynomial-time α_2-approximation algorithm.

small. For example, we have $\texttt{best}(\mathcal{L}_{\text{FVS}}, 1.5) \approx 1.2068$ and $\min(\texttt{best}(\mathcal{L}'_{\text{FVS}}, 1.5), \texttt{best}(\mathcal{L}''_{\text{FVS}}, 1.5)) = \texttt{best}(\mathcal{L}''_{\text{FVS}}, 1.5) \approx 1.2072$. Similar observations can be made for Tournament FVS. Note that the fact that we only obtain small improvements by using multiple oracles is not a shortcoming of the algorithms designed in this paper, but inherent to the problem by Theorem 2.8.

We stress that these results are not limited to FVS and Tournament FVS. Indeed, there is wide range of vertex-deletion problems for which a single-exponential fpt algorithm running in time $\mathcal{O}^*(c_1^k)$ as well as a polynomial-time α_2-approximation is known, for suitable constants $c_1, \alpha_2 > 1$. A list of examples is given in Table 3 (running times for all problems can be found in Appendix B).

Notably, Edge Multicut on Trees is not a vertex-deletion problem, but an edge-deletion problem. This means we set $U := E(G)$ which implies that the running time in Theorem 2.3 is measured with respect to the number of edges rather than the number of vertices. However, for this particular problem, the input graph G is a tree which implies that $|E(G)| \leq |V(G)|$, and hence we obtain the same runtime bound with respect to the number of vertices.

For all the problems listed in Table 3, by Theorem 2.3, monotone local search results in a β-approximation algorithm running in time $\mathcal{O}^*((\texttt{best}(\mathcal{L}_{c_1, \alpha_2}, \beta))^n)$, where $\mathcal{L}_{c_1, \alpha_2} := \{(1, c_1), (\alpha_2, 1)\}$, for all $\beta \geq 1$ that outperforms all previously existing algorithms. Note that the algorithms we obtain are randomized. However, by Theorem 2.4, we can also obtain a deterministic β-approximation algorithm at the cost of an additional subexponential factor if all parameterized extension subroutine are deterministic. Looking at Table 3, this is true for all listed problems except FVS and Tournament FVS.

Note that our results are also applicable if only either a single-exponential fpt algorithm or a polynomial-time constant-factor approximation algorithm is available. As a notable example, Odd Cycle Transversal (OCT) can be solved in time $\mathcal{O}^*(2.3146^k)$ [55], and has no constant-factor approximation algorithm assuming the Unique Games Conjecture [43]. On the other side, the Partial Vertex Cover problem has a polynomial-time 2-approximation [12], and is known to be W[1]-hard [37] which means that it cannot be solved in single-exponential fpt time assuming FPT $\neq$ W[1]. Still, for both problems, we obtain a β-approximation algorithm that is faster than the brute-force search (see Theorem 2.18) and, in the case of Odd Cycle Transversal (OCT), than the algorithm obtained from [26] (see Lemma 2.20). For both problems, the running times of the obtained algorithms can again be found in Appendix B.

3.2 Exploiting Parameterized Approximation Algorithms

We also obtain new algorithms for problems that are neither known to have a single-exponential fpt algorithm nor a polynomial-time approximation algorithm, but admit a single-exponential parameterized constant-factor approximation algorithm. For DFVS, Subset DFVS, DOCT and Multicut, [54] provides a 2-approximation algorithm that runs in time $\mathcal{O}^*(c^k)$ for some constant c. For example, one

(α, c)	1.01	1.02	1.03	1.04	1.05	1.06	1.07	1.08	1.09	1.1
$(\beta, c_{\mathrm{VC}}(\beta))$	1.2038	1.1955	1.183	1.1697	1.158	1.1475	1.138	1.1294	1.1214	1.114
$\mathcal{L}_{\mathrm{VC}}$	1.1891	1.1752	1.1649	1.1566	1.1496	1.1433	1.1358	1.1275	1.1197	1.1125

(α, c)	1.02	1.04	1.06	1.08	1.1	1.12	1.14	1.16	1.18	1.2
$(\beta, c_{\mathrm{HS}}(\beta))$	1.472	1.441	1.4157	1.3863	1.3543	1.3262	1.3013	1.2787	1.2584	1.2399
$\mathcal{L}_{\mathrm{HS}}$	1.4489	1.4083	1.3775	1.3527	1.3319	1.314	1.2984	1.2783	1.2579	1.2393

Table 4: Running times for Vertex Cover and 3-Hitting Set. An entry in row (α, c) and column β is $\mathtt{best}(\{(\alpha, c)\}, \beta)$. The middle row in each table is the result from [26], the last row is the result attained in this paper

can easily observe from the description of the DFVS algorithm in [54] that it runs in time $\mathcal{O}^*(1024^k)$. Using Theorem 2.3, monotone local search results in an exponential β-approximation algorithm that runs in time $\mathcal{O}^*(\mathtt{best}(2, 1024, \beta))$ for all $\beta > 1$. By Theorem 2.18, this algorithm is qualitatively better than the brute-force β-approximation algorithm running in time $\mathcal{O}^*((\mathtt{brute}(\beta))^n)$. For example, $\mathtt{best}(2, 1024, 1.1) \approx 1.71520$ and $\mathtt{brute}(1.1) \approx 1.71527$. Similar results can be obtained for the other problems.

Moreover, for the problem Symmetric Directed Multicut, it is possible to adapt a parameterized 2-approximation algorithm (which runs in time $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$) [24] to obtain a parameterized α-approximation algorithm running in time $\mathcal{O}^*(c^k)$ for some constants $\alpha, c > 1$ [65]. As a consequence, we also obtain an exponential β-approximation algorithm for this problem that beats the brute-force β-approximation algorithm for every $\beta > 1$.

Similarly, using the $\mathcal{O}^*(c^k)$-time 2-approximation algorithm for d-Steiner Multicut in [59, Theorem 37] for some $c > 1$, one can obtain a β-approximation algorithm that beats the brute-force β-approximation algorithm for all $\beta > 1$.

3.3 Vertex Cover and 3-Hitting Set

Finally, we consider the Vertex Cover and 3-Hitting Set problem. Both problems have not only been extensively studied for their exact parameterized complexity [15, 64], but also received significant attention in the area of parameterized approximation algorithms [11, 29, 45]. As a result, these two problems are the main applications considered in [26] for transforming a parameterized β-approximation algorithm into an exponential β-approximation algorithm. Despite a parameterized β-approximation being the natural oracle choice to obtain an exponential β-approximation, our algorithmic framework allows us to obtain further improvements for both problems compared to [26].

For every $\alpha \in [1, 2]$ the best known running time of a parameterized randomized α-approximation algorithm for VC is attained in [45] if $\alpha \gtrsim 1.03$, and in [11] if $\alpha \lesssim 1.03$ (using the exact algorithm from [15] for $\alpha = 1$). Let us denote by $c_{\mathrm{VC}}(\alpha)$ the base of the currently fastest known parameterized α-approximation algorithm for VC (i.e., an α-approximation can be computed in time $\mathcal{O}^*((c_{\mathrm{VC}}(\alpha))^k)$). Similarly, for $\alpha \in [1, 3]$, we write $c_{\mathrm{HS}}(\alpha)$ for the base of the currently fastest known parameterized α-approximation algorithm for 3-HS. This best known base is attained by either [29] if $\alpha \lesssim 1.08$ (using the exact algorithm from [64] for $\alpha = 1$), or [45] if $\alpha \gtrsim 1.08$. Note that $c_{\mathrm{VC}}(2) = c_{\mathrm{HS}}(3) = 1$.

As indicated above, the currently fastest (randomized) exponential β-approximation algorithm for VC (resp. 3-HS) was obtained in [26] and runs in time $\mathcal{O}^*(d^n)$ where $d := \mathtt{best}(\beta, c_{\mathrm{VC}}(\beta), \beta)$ (resp. $d := \mathtt{best}(\beta, c_{\mathrm{HS}}(\beta), \beta)$) using Lemma 2.19.

Now, to apply our algorithmic framework, we need to fix an oracle specification list $\mathcal{L}_{\mathrm{VC}}$ (resp. $\mathcal{L}_{\mathrm{HS}}$). Since we can only provide a finite number of oracles (and it is a priori unclear how to choose those oracles optimally), we adopt the basic approach of equally discretizing the range for α. We set $A_{\mathrm{VC}} := \{1, 1.01, 1.02, 1.03, \ldots, 1.99, 2\}$ and $A_{\mathrm{HS}} := \{1, 1.02, 1.04, 1.06, \ldots, 2.98, 3\}$ (both sets contain 101 elements). Then we define $\mathcal{L}_{\mathrm{VC}} := \{(\alpha, c_{\mathrm{VC}}(\alpha)) \mid \alpha \in A_{\mathrm{VC}}\}$ and $\mathcal{L}_{\mathrm{HS}} := \{(\alpha, c_{\mathrm{HS}}(\alpha)) \mid \alpha \in A_{\mathrm{HS}}\}$.

For the Vertex Cover problem, it can be observed that $\mathtt{best}(\mathcal{L}_{\mathrm{VC}}, \beta) < \mathtt{best}(\beta, c_{\mathrm{VC}}(\beta), \beta)$ for all

(α, c)	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta, c_{\text{vc}}(\beta))$	1.063058	1.036524	1.020288	1.0098549	1.0043411	1.0015504	1.00039597	1.000042813
$\mathcal{L}_{\text{vc}}$	1.061819	1.035901	1.019999	1.0097939	1.0042837	1.0015355	1.00039185	1.000042504

Table 5: Running times for VERTEX COVER. An entry in row (α, c) and column β is $\texttt{best}(\{(\alpha, c)\}, \beta)$. The middle row in each table is the result from [26], the last row is the result attained in this paper

$\beta \in (1, 2) \cap A_{\text{VC}}$ (by evaluating both functions up to a sufficiently large precision). The most significant improvements occur for small values of β (see Table 4), and it seems that this improvement can be mostly attributed to the possibility of using the exact fpt algorithm for VERTEX COVER [15] as a subroutine. For larger approximation ratios, we only obtain small improvements as can be observed from Table 5.

Generally speaking, it is also noteworthy that, even if the algorithm has access to $|A_{\text{VC}}| = 101$ many different oracles, only 2-3 oracles corresponding to tuples from the specification list $\mathcal{L}_{\text{VC}}$ are actually used, and the corresponding approximation ratios α are close to β. For example,

$$\texttt{best}(\mathcal{L}_{\text{VC}}, 1.5) = \texttt{best}(\{(1.49, c_{\text{VC}}(1.49)), (1.5, c_{\text{VC}}(1.5))\}, 1.5)$$
$$< \min\{\texttt{best}(1.49, c_{\text{VC}}(1.49), 1.5), \texttt{best}(1.5, c_{\text{VC}}(1.5), 1.5)\}$$
$$= \texttt{best}(1.49, c_{\text{VC}}(1.49), 1.5) \approx 1.0098063.$$

Similar observations can be made for 3-HITTING SET.

4 Approximate Monotone Local Search

In this section, we analyse Algorithm 2 and describe how it can be derandomized. More precisely, we prove Lemmas 2.6 and 2.7

4.1 Correctness and Basic Analysis

We first analyse the randomized algorithm $\texttt{Approximate-MLS}_{\mathcal{L}, \beta}$ for the $\mathcal{L}$-SUB problem. This algorithm uses algorithm $\texttt{Sample}$ as a subroutine. See Section 2 for the description of these algorithms. Also recall the definitions of $\texttt{hyper}(n, k, t, x)$ and $M^*_{\alpha, \beta}$ from Equations (2) and (4), respectively.

Lemma 4.1 (Correctness). *For every specification list $\mathcal{L}$ and $\beta \geq 1$, $\texttt{Approximate-MLS}_{\mathcal{L}, \beta}$ (Algorithm 2) is a randomized β-approximation algorithm for $\mathcal{L}$-SUB.*

Proof. Let U be a finite set system and $\mathcal{F}$ be a monotone set system of U. Also, for every $(\alpha, c) \in \mathcal{L}$ let $\texttt{Ext}_{\alpha, c}$ be an α-extension oracle of U and $\mathcal{F}$. Let $\mathcal{F}$ be the implicit monotone subset family associated with $\mathcal{L}$. Also, let $\texttt{OPT} = \text{argmin}_{S \in \mathcal{F}} |S|$ be a minimum size solution of the $\mathcal{L}$-SUB instance U and $\mathcal{F}$. Consider an execution of $\texttt{Sample}(U, k, t, \alpha, \beta, \texttt{Ext}_{\alpha, c})$ (Algorithm 1) in which $k = |\texttt{OPT}|$, $M^*_{\alpha, \beta} \cdot k \leq t \leq \beta \cdot k$ and $(\alpha, c) \in \mathcal{L}$.

If the algorithm selects a set X in Step 1 such that

$$|\texttt{OPT} \cap X| \geq \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}$$

then

$$|\texttt{OPT} \setminus X| \leq k - \left\lceil \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha} \right\rceil.$$

Moreover, $(\texttt{OPT} \setminus X) \cup X \in \mathcal{F}$ since $(\texttt{OPT} \setminus X) \cup X = \texttt{OPT} \cup X \supseteq \texttt{OPT}$ as $\mathcal{F}$ is monotone by assumption.

Since $\texttt{Ext}_{\alpha, c}$ is an α-extension oracle for $(U, \mathcal{F})$, given the input $\left(U, X, k - \left\lceil \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha} \right\rceil \right)$ it returns a set Y such that $X \cup Y \in \mathcal{F}$ and with probability at least $\frac{1}{2}$ it holds that

$$|Y| \leq \alpha \cdot \left(k - \left\lceil \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha} \right\rceil \right) \leq \alpha \cdot \left(k - \left(1 - \frac{\beta}{\alpha}\right) \cdot k - \frac{t}{\alpha}\right) = \beta k - t.$$

Let $Z := X \cup Y$ as in Step 3 of Algorithm 1. Then $Z \in \mathcal{F}$ and $|Z| = |X| + |Y| \le t + \beta k - t = \beta k$. It follows that

$$\Pr\left(\texttt{Sample}(U, |\texttt{OPT}|, t, \alpha, \beta, \texttt{Ext}_{\alpha,c}) \text{ returns a set of size at most } \beta \cdot |\texttt{OPT}|\right)$$

$$\ge \frac{1}{2} \cdot \Pr\left(|X \cap \texttt{OPT}| \ge \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right) = \frac{1}{2} \cdot \texttt{hyper}\left(n, |\texttt{OPT}|, t, \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right) \quad (14)$$

where $\texttt{hyper}$ is the function defined in Equation (2).

Now, consider the execution of Algorithm 2 with U as its input and let S be the set returned by Algorithm 2. It is easy to see that $S \in \mathcal{F}$ since Algorithm 1 always returns a set from $\mathcal{F}$. If $|\texttt{OPT}| \ge \frac{n}{\beta}$ then $|S| \le |U| \le \beta \cdot |\texttt{OPT}|$ and the algorithm returns an β-approximate solution as desired. So we may assume that $|\texttt{OPT}| < \frac{n}{\beta}$. Consider the iteration of the for-loop in Step 2 of Algorithm 2 in which $k = |\texttt{OPT}|$. Using Equation (14), at least one of the calls to Algorithm 1 in this iteration returns a set of size at most $\beta \cdot |\texttt{OPT}|$ with probability at least

$$1 - \left(1 - \frac{1}{2} \cdot \texttt{hyper}\left(n, k, t, \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right)\right)^{2/\texttt{hyper}\left(n,k,t,\left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right)} \ge 1 - \exp(-1) > \frac{1}{2}.$$

So the minimum cardinality set in $\mathcal{S}$ (at the end of Algorithm 2) has size at most $\beta \cdot |\texttt{OPT}|$ with probability at least $\frac{1}{2}$. Hence, the set S returned by the algorithm satisfies $|S| \le \beta \cdot |\texttt{OPT}|$ with probability at least $\frac{1}{2}$. $\qquad \square$

Recall the definition of $f_{\mathcal{L},\beta}(n)$ from Equation 5.

Lemma 4.2 (Running time). $\texttt{Approximate-MLS}_{\mathcal{L},\beta}$ *(Algorithm 2) has cost* $f_{\mathcal{L},\beta}(n) \cdot n^{\mathcal{O}(1)}$.

Proof. First consider Algorithm 1. In Line 2, the algorithm calls $\texttt{Ext}_{\alpha,c}$ with parameter

$$k - \left\lceil \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right\rceil \le k - \left(1 - \frac{\beta}{\alpha}\right) \cdot k - \frac{t}{\alpha} = \frac{\beta k - t}{\alpha}.$$

So this step incurs a cost of at most $c^{\frac{\beta k - t}{\alpha}} = \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)$. This means that Step 4 of Algorithm 2 incurs a total cost of

$$2 \cdot \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\texttt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)}.$$

Since t, α, c are chosen to minimize this cost, we obtain that one iteration of the for-loop takes incurs a cost of

$$\min_{(\alpha,c) \in \mathcal{L}} \min_{t \in \left[M^*_{\alpha,\beta} k, \beta k\right] \cap \mathbb{N}} 2 \cdot \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\texttt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)}.$$

As a result, every single iteration costs at most

$$\max_{k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}} \min_{(\alpha,c) \in \mathcal{L}} \min_{t \in \left[M_{\alpha,\beta} k, \beta k\right] \cap \mathbb{N}} 2 \cdot \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\texttt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)}.$$

Since there are at most n iterations of the for-loop, the entire algorithm has cost $f_{\mathcal{L},\beta}(n) \cdot n^{\mathcal{O}(1)}$, as desired. $\qquad \square$

Now, Lemma 2.6 immediately follows from Lemmas 4.1 and 4.2.

4.2 Derandomization

Next, we prove Lemma 2.7, i.e., we argue how to derandomize $\texttt{Approximate-MLS}_{\mathcal{L},\beta}$. Towards this end, the key notion is that of a set-intersection-family.

Definition 4.3. *Let U be a universe of size n and let $p, q, r \geq 1$ such that $n \geq p \geq r$ and $n - p + r \geq q \geq r$. A family $\mathcal{C} \subseteq \binom{U}{q}$ is a (n, p, q, r)-set-intersection-family if for every $T \in \binom{U}{p}$ there is some $X \in \mathcal{C}$ such that $|T \cap X| \geq r$.*

The basic idea of the derandomization is, instead of repeatedly sampling a random set X in Algorithm 1, to compute a suitable set-intersection-family $\mathcal{C}$ and iterate over all its elements X. Towards this, let us define

$$\kappa(n, p, q, r) := \frac{\binom{n}{q}}{\binom{p}{r} \cdot \binom{n-p}{q-r}}.$$

The following theorem computes the desired set-intersection-family of small size.

Theorem 4.4 ([26, Theorem 4.2])**.** *There is an algorithm that, given a set U of size n and numbers $p, q, r \geq 1$ such that $n \geq p \geq r$ and $n - p + r \geq q \geq r$, computes an (n, p, q, r)-set-intersection-family of size $\kappa(n, p, q, r) \cdot 2^{o(n)}$ in time $\kappa(n, p, q, r) \cdot 2^{o(n)}$.*

With the last theorem in hand, we are ready to prove Lemma 2.7. The updated deterministic algorithm is given in Algorithm 3. Observe that it receives deterministic extension oracles.

Algorithm 3 $\texttt{Deterministic-Approximate-MLS}_{\mathcal{L},\beta}$

Input: A universe U and a deterministic extension oracle $\texttt{Ext}_{\alpha,c}$ for every $(\alpha, c) \in \mathcal{L}$
1: $\mathcal{S} \leftarrow \emptyset$.
2: **for** k from 0 to $\frac{n}{\beta}$ **do**
3:      Find $(\alpha, c) \in \mathcal{L}$ and $t \in \left[M^*_{\alpha,\beta} k, \beta k \right] \cap \mathbb{N}$ which minimize $\left(\frac{c^{\frac{\beta k - t}{\alpha}}}{\texttt{hyper}\left(n, k, t, \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right)} \right)$.
4:      Set $x := \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}$.
5:      Find $y \in \{\lceil x \rceil, \ldots, \min\{t, k\}\}$ for which $\kappa(n, k, t, y)$ is minimized.
6:      Compute a (n, k, t, y)-set-intersection-family $\mathcal{C}$.
7:      **for** $X \in \mathcal{C}$ **do**
8:          $Y \leftarrow \texttt{Ext}_{\alpha,c}(U, X, k - \lceil x \rceil)$.
9:          $\mathcal{S} \leftarrow \mathcal{S} \cup \{X \cup Y\}$.
10: **Return** a minimum-sized set in $\mathcal{S}$.

Proof of Lemma 2.7. Let U be a finite set system of size n and $\mathcal{F}$ be a monotone set system of U. Also, for every $(\alpha, c) \in \mathcal{L}$ let $\texttt{Ext}_{\alpha,c}$ be a deterministic α-extension oracle of U and $\mathcal{F}$. Let $\mathcal{F}$ be the implicit monotone subset family associated with $\mathcal{L}$. Also, let $\texttt{OPT} = \text{argmin}_{S \in \mathcal{F}} |S|$ be a minimum size solution of the $\mathcal{L}$-SUB instance U and $\mathcal{F}$.

Consider Algorithm 3. First, observe that $\emptyset \neq \mathcal{S} \subseteq \mathcal{F}$ by Definition 2.2. If $|\texttt{OPT}| \geq \frac{n}{\beta}$ then every set in $\mathcal{S}$ is a valid β-approximation. So suppose $|\texttt{OPT}| \leq \frac{n}{\beta}$ and consider the iteration in which $k = |\texttt{OPT}|$.

By definition of a set-intersection-family, there is some $X \in \mathcal{C}$ such that $|X| = t$ and

$$|\texttt{OPT} \cap X| \geq y \geq \lceil x \rceil.$$

Then

$$|\texttt{OPT} \setminus X| \leq k - \lceil x \rceil.$$

Moreover, $(\texttt{OPT} \setminus X) \cup X \in \mathcal{F}$ since $(\texttt{OPT} \setminus X) \cup X = \texttt{OPT} \cup X \supseteq \texttt{OPT}$ as $\mathcal{F}$ is monotone by assumption.

Since $\texttt{Ext}_{\alpha,c}$ is an α-extension oracle for $(U, \mathcal{F})$, given the input $(U, X, k - \lceil x \rceil)$ it returns a set Y such that $X \cup Y \in \mathcal{F}$ and

$$|Y| \leq \alpha \cdot (k - \lceil x \rceil) \leq \alpha \cdot \left(k - \left(1 - \frac{\beta}{\alpha}\right) \cdot k - \frac{t}{\alpha} \right) = \beta k - t.$$

So $|X \cup Y| \leq t + \beta k - t = \beta k$ which means that $\mathcal{S}$ contains a solution set of size at $\beta \cdot |\mathsf{OPT}|$ as desired.

It remains to analyse the cost of Algorithm 3. Suppose $k \in \{0, \ldots, \lfloor \frac{n}{\beta} \rfloor\}$. The algorithm computes a number $y \in \{\lceil x \rceil, \ldots, \min\{t, k\}\}$ for which $\kappa(n, k, t, y)$ is minimized, i.e., $1/\kappa(n, k, t, y)$ is maximized. By Theorem 4.4 we get that

$$|\mathcal{C}| = \kappa(n, k, t, y) \cdot 2^{o(n)} \leq n \cdot \frac{1}{\mathtt{hyper}\,(n, k, t, x)} \cdot 2^{o(n)}.$$

Also note that the family $\mathcal{C}$ can be computed within the same time bound.

It follows that the execution of the inner for-loop requires cost

$$\frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)} \cdot 2^{o(n)}.$$

Since t, α, c are chosen to minimize this cost, we obtain that one iteration of the outer for-loop incurs a cost of

$$\min_{(\alpha, c) \in \mathcal{L}} \min_{t \in [M^*_{\alpha, \beta} k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)} \cdot 2^{o(n)}.$$

Note that all other steps before the computation of the set-intersection family can be done using polynomially many computation steps. As a result, every single iteration costs at most

$$\max_{k \in [0, \frac{n}{\beta}] \cap \mathbb{N}} \min_{(\alpha, c) \in \mathcal{L}} \min_{t \in [M_{\alpha, \beta} k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}\left(n, k, t, (1 - \frac{\beta}{\alpha}) \cdot k + \frac{t}{\alpha}\right)} \cdot 2^{o(n)}.$$

Since there are at most n iterations of the outer for-loop, the entire algorithm has cost $f_{\mathcal{L}, \beta}(n) \cdot 2^{o(n)}$ as desired. $\qquad\square$

5 Lower Bounds

In this section we prove Lemmas 2.8 and 2.9. We also argue how to derive Theorem 2.10 from Lemma 2.9.

We begin with the proof of Lemma 2.9 as it is technically simpler. We actually prove the following slightly stronger statement. Recall the definition of $f_{\mathcal{L}, \beta}(n)$ (5).

Lemma 5.1. *Let $c > 1$ and let $\mathcal{A}$ be an algorithm for c-DEC. Then $\mathtt{cost}_{\mathcal{A}}(n) \geq \frac{1}{2} \cdot f_{\{(1,c)\}, 1}(n)$ for every $n \geq 1$.*

Proof. Let $n \in \mathbb{N}$. We assume n is fixed throughout this proof. By (5) it holds that

$$f_{\{(1,c)\}, 1}(n) = \max_{k \in [0, n] \cap \mathbb{N}} \min_{t \in [0, k] \cap \mathbb{N}} \frac{\exp\left((k - t) \cdot \ln c\right)}{\mathtt{hyper}\,(n, k, t, t)} = \max_{k \in [0, n] \cap \mathbb{N}} \min_{t \in [0, k] \cap \mathbb{N}} c^{k-t} \cdot \frac{\binom{n}{k}}{\binom{n-t}{k-t}}. \tag{15}$$

The last equality uses $\mathtt{hyper}\,(n, k, t, t) = \frac{\binom{n-t}{k-t}}{\binom{n}{k}}$ by (2).

For all $0 \leq k \leq n$ and $0 \leq t \leq k$ we define

$$\begin{aligned}
G(k, t) &:= \frac{c^{k-t} \cdot \binom{n}{k}}{\binom{n-t}{k-t}}, \\
t^*(k) &:= \operatorname*{argmin}_{t \in [0, k] \cap \mathbb{N}} G(k, t), \\
k^* &:= \operatorname*{argmax}_{k \in [0, n] \cap \mathbb{N}} G(k, t^*(k)).
\end{aligned}$$

By (15) it follows that $f_{\{(1,c)\},1}(n) = G(k^*, t^*(k^*))$. Furthermore, it holds that

$$t^*(k) = \operatorname*{argmin}_{t \in [0,k] \cap \mathbb{N}} \frac{c^{k-t} \cdot \binom{n}{k}}{\binom{n-t}{k-t}} = \operatorname*{argmin}_{t \in [0,k] \cap \mathbb{N}} \frac{c^{k-t}}{\binom{n-t}{k-t}}. \tag{16}$$

We set $U := [n]$. Our lower bound is based on the difficulty that algorithms have to distinguish between the set-systems $\mathcal{F} = \emptyset$ and $\mathcal{F} = \{R\}$ where R is a uniformly sampled random subset of U of size k^*.

For a set system $\mathcal{F}$ of U we define an oracle $\mathrm{Ext}_{\mathcal{F}}$ by setting

$$\mathrm{Ext}_{\mathcal{F}}(X, \ell) := \begin{cases} \mathrm{YES} & \text{if there is an } \ell\text{-extension of } X \text{ with respect to } U \text{ and } \mathcal{F}, \\ \mathrm{NO} & \text{otherwise.} \end{cases}$$

Clearly, $\mathrm{Ext}_{\mathcal{F}}$ is a exact extension oracle for $\mathcal{F}$. Note that $\mathrm{Ext}_\emptyset$ always returns NO.

We assume the algorithm $\mathcal{A}$ gets a string of bits $b \in \{0,1\}^{q(n)}$ as its source of randomness, where q is an arbitrary function. This means $\mathcal{A}$ is deterministic given the input set U, the oracle Ext and the random bits b. Let us denote by $\mathcal{A}(U, \mathrm{Ext}, b) \in \{\mathrm{YES}, \mathrm{NO}\}$ the output of the algorithm $\mathcal{A}$.

Let $Q(b) \subseteq U \times \mathbb{N}$ be the set of oracle queries the algorithm $\mathcal{A}$ makes on input U with oracle $\mathrm{Ext}_\emptyset$ and random bits b. Equivalently, $Q(b)$ is the set of queries $\mathcal{A}$ makes given the universe U in case the oracle always returns NO for an answer. Observe that, in general, if all the responses to the queries the algorithm makes are NO, then it has to return NO, because otherwise it violates the correctness requirement in case its given the oracle $\mathrm{Ext}_\emptyset$ for the set system $\emptyset$.

We define the *coverage* of an oracle query (X, ℓ) by

$$\begin{aligned} \mathrm{coverage}(X, \ell) &:= \left\{ S \subseteq U \,\middle|\, |S| = k^*, \ X \subseteq S, \ |S \setminus X| \leq \ell \right\} \\ &= \left\{ S \subseteq U \,\middle|\, |S| = k^* \text{ and } X \text{ has an } \ell\text{-extension w.r.t. the set system } \{S\} \right\} \\ &= \left\{ S \subseteq U \,\middle|\, \mathrm{Ext}_{\{S\}}(X, \ell) = \mathrm{YES} \right\}. \end{aligned}$$

Given a set $W \subseteq 2^U \times \mathbb{N}$ of queries we define $\mathrm{coverage}(W) := \bigcup_{(X,\ell) \in W} \mathrm{coverage}(X, \ell)$.

Claim 5.2. *Let $b \in \{0,1\}^{q(n)}$. Then*

$$\mathrm{cost}_{\mathcal{A}}(n) \geq \frac{|\mathrm{coverage}(Q(b))|}{\binom{n}{k^*}} \cdot f_{\{(1,c)\},1}(n).$$

Proof. Consider the execution of $\mathcal{A}$ on input U using the oracle $\mathrm{Ext}_\emptyset$ and random bits b. By definition, the cost of the execution is $\sum_{(X,\ell) \in Q(b)} c^\ell$ and thus, $\mathrm{cost}_{\mathcal{A}}(n) \geq \sum_{(X,\ell) \in Q(b)} c^\ell$. Therefore,

$$\begin{aligned} |\mathrm{coverage}(Q(b))| &\leq \sum_{(X,\ell) \in Q(b)} |\mathrm{coverage}(X, \ell)| \\ &= \sum_{(X,\ell) \in Q(b) \text{ s.t. } k^* - \ell \leq |X| \leq k^*} \binom{n - |X|}{k^* - |X|} \\ &= \sum_{(X,\ell) \in Q(b) \text{ s.t. } k^* - \ell \leq |X| \leq k^*} c^{k^* - |X|} \cdot \frac{\binom{n-|X|}{k^*-|X|}}{c^{k^*-|X|}} \\ &\leq \sum_{(X,\ell) \in Q(b) \text{ s.t. } k^* - \ell \leq |X| \leq k^*} c^{k^* - |X|} \cdot \frac{\binom{n-t^*(k^*)}{k^*-t^*(k^*)}}{c^{k^*-t^*(k^*)}} \\ &= \frac{\binom{n}{k^*}}{G(k^*, t^*(k^*))} \cdot \sum_{(X,\ell) \in Q(b) \text{ s.t. } k^* - \ell \leq |X| \leq k^*} c^{k^* - |X|} \\ &\leq \frac{\binom{n}{k^*}}{G(k^*, t^*(k^*))} \cdot \sum_{(X,\ell) \in Q(b)} c^\ell \\ &\leq \frac{\binom{n}{k^*}}{G(k^*, t^*(k^*))} \cdot \mathrm{cost}_{\mathcal{A}}(n). \end{aligned}$$

The second inequality follows from (16). Since $f_{\{(1,c)\},1}(n) = G(k^*, t^*(k^*))$, the assertion of the claim follows. ⌟

Now let $b^* \in \{0,1\}^{q(n)}$ be the bit-string for which $|\texttt{coverage}(Q(b^*))|$ is maximal. In light of the last claim, in order to lower bound the cost of $\mathcal{A}$, it suffices to lower bound the cardinality of $\texttt{coverage}(Q(b^*))$. We use the correctness properties of $\mathcal{A}$ to attain such a lower bound.

Claim 5.3. *It holds that*
$$|\texttt{coverage}(Q(b^*))| \geq \frac{1}{2} \cdot \binom{n}{k^*}.$$

Proof. Consider the execution of $\mathcal{A}$ on input U using the oracle $\texttt{Ext}_{\{S\}}$, where $S \subseteq U$ such that $|S| = k^*$, and a bit-string b. If $S \notin \texttt{coverage}(Q(b))$ then the set of oracle queries the algorithm makes is exactly $Q(b)$ and all the queries return NO. So the algorithm also has to return NO. It follows that
$$\mathcal{A}(U, \texttt{Ext}_{\{S\}}, b) = \texttt{YES} \implies S \in \texttt{coverage}(Q(b)).$$

We define two independent random variable. Let $R \subseteq U$ be a uniformly random subset of U of size k^*. Also, we define $r \in \{0,1\}^{q(n)}$ to be a uniformly random string of bits of length $q(n)$. Then

$$
\begin{aligned}
\Pr\left(\mathcal{A}\left(U, \texttt{Ext}_{\{S\}}, r\right) = \texttt{YES}\right) &\leq \Pr(R \in \texttt{coverage}(Q(r))) \\
&= \sum_{b \in \{0,1\}^{q(n)}} \Pr(r = b) \cdot \Pr(R \in \texttt{coverage}(Q(r)) \mid r = b) \\
&= \sum_{b \in \{0,1\}^{q(n)}} \Pr(r = b) \cdot \Pr(R \in \texttt{coverage}(Q(b))) \\
&= \sum_{b \in \{0,1\}^{q(n)}} \Pr(r = b) \cdot \frac{|\texttt{coverage}(Q(b))|}{\binom{n}{k^*}} \quad\quad (17) \\
&\leq \sum_{b \in \{0,1\}^{q(n)}} \Pr(r = b) \cdot \frac{|\texttt{coverage}(Q(b^*))|}{\binom{n}{k^*}} \\
&= \frac{|\texttt{coverage}(Q(b^*))|}{\binom{n}{k^*}}.
\end{aligned}
$$

The second equality holds since r is independent of R. Furthermore, as $\mathcal{A}$ returns YES with probability at least $\frac{1}{2}$ for YES instances,

$$
\begin{aligned}
\Pr(\mathcal{A}\left(U, \texttt{Ext}_{\{R\}}, r\right) = \texttt{YES}) &= \sum_{S \subseteq U} \Pr(R = S) \cdot \Pr(\mathcal{A}\left(U, \texttt{Ext}_{\{R\}}, r\right)) = \texttt{YES} \mid R = S) \\
&\geq \sum_{S \subseteq U} \Pr(S = R) \cdot \frac{1}{2} = \frac{1}{2}.
\end{aligned} \quad (18)
$$

The assertion of the claim now follows by combining (17) and (18). ⌟

Combining Claims 5.2 and 5.3 we get $\texttt{cost}_{\mathcal{A}}(n) \geq \frac{1}{2} \cdot f_{\{(1,c)\},1}(n)$, which completes the proof. □

Note that Lemma 2.9 immediately follows from Lemma 5.1. Next, we prove Theorem 2.10 using Lemma 2.9.

Proof of Theorem 2.10. Let $c > 1$ and let $\mathcal{A}$ be a randomized algorithm for c-DEC. By Lemma 2.9 we have
$$\texttt{cost}_{\mathcal{A}}(n) \geq n^{-\mathcal{O}(1)} \cdot f_{\{(1,c)\},1}(n) \geq n^{-\mathcal{O}(1)} \cdot (\texttt{amls}(\{(1,c)\}, 1))^n, \quad (19)$$
where second equality follows from Lemma 2.11.

Observe that $2 - \frac{1}{c} \in (1, c+1)$ and
$$\mathcal{D}\left(1 \middle\| \frac{(2 - \frac{1}{c}) - 1}{c - 1}\right) = \mathcal{D}\left(1 \middle\| \frac{1 - \frac{1}{c}}{c - 1}\right) = \mathcal{D}\left(1 \middle\| \frac{1}{c}\right) = \ln c.$$

Therefore, by Lemma 2.19, it holds that $\texttt{amls}(\{(1,c)\}, 1) = 2 - \frac{1}{c}$. In combination with (19) it follows that $\texttt{cost}_{\mathcal{A}}(n) \geq n^{-\mathcal{O}(1)} \cdot \left(2 - \frac{1}{c}\right)^n$. □

The proof of Lemma 2.8 follows the same principles as the proof of Lemma 2.9. It defines a *coverage* for each query, shows the cost of the algorithm is at least the cardinality of the coverage of all queries, and then provides a lower bound on the cardinality of the coverage. The proof is slightly more complicated than the proof of Lemma 2.9 due to the involvement of multiple oracles and since the oracles only provide approximations. As before, we actually prove a slightly stronger statement.

Lemma 5.4. *Let $\beta \geq 1$. Also let $\mathcal{L}$ be a specification list and $\mathcal{A}$ be a randomized β-approximation algorithm for $\mathcal{L}$-SUB. Then*

$$\texttt{cost}_{\mathcal{A}}(n+1) \geq \frac{f_{\mathcal{L},\beta}(n)}{2 \cdot (n+1) \cdot \max_{(\alpha,c) \in \mathcal{L}} c}$$

for every $n \geq 1$.

Proof. Let $n \geq 1$. We assume n is fixed for the remainder of the proof. Also suppose that $\mathcal{L} = \{(\alpha_1, c_1), \ldots, (\alpha_s, c_s)\}$ and define $U := [n+1]$. For every $j \in [s]$ we define the functions

$$G_j(k,t) := \frac{(c_j)^{\frac{\beta k - t}{\alpha_j}}}{\texttt{hyper}\left(n, k, t, k - \frac{\beta k - t}{\alpha_j}\right)}, \tag{20}$$

$$t_j^*(k) := \operatorname*{argmin}_{t \in \left[M_{\alpha_j,\beta}^* \cdot k, \ \beta \cdot k\right] \cap \mathbb{N}} G_j(k,t), \tag{21}$$

where $M_{\alpha_j,\beta}^*$ is as defined in (4). Furthermore, we define

$$j^*(k) = \operatorname*{argmin}_{j \in [s]} G_j(k, t_j^*(k))$$

and

$$k^* = \operatorname*{argmax}_{k \in [0, \frac{n}{\beta}] \cap \mathbb{N}} G_{j^*(k)}(k, t_{j^*(k)}^*(k)).$$

With a slight abuse of notation we write $j^* := j^*(k^*)$ and $t^* := t_{j^*}^*(k^*)$. By (5) we have

$$f_{\mathcal{L},\beta}(n) = G_{j^*}(k^*, t^*).$$

We define the set system $\mathcal{F}_{\texttt{adv}} := \{S \subseteq U \mid |S| \geq \lfloor \beta \cdot k^* \rfloor + 1\}$. Since $\beta \cdot k^* \leq n$ and $U = [n+1]$ it holds that $\mathcal{F}_{\texttt{adv}} \neq \emptyset$. For every $T \subseteq U$ we define the set system $\mathcal{F}_T = \{S \subseteq U \mid T \subseteq S\} \cup \mathcal{F}_{\texttt{adv}}$, i.e., $\mathcal{F}_T$ is the set system containing all supersets of T and all sets of cardinality at least $\lfloor \beta \cdot k^* \rfloor + 1$. Clearly, $\mathcal{F}_{\texttt{adv}}$ and $\mathcal{F}_T$ are monotone set systems of U for every $T \subseteq U$. Our lower bound is based on the fact that $\mathcal{A}$ requires queries of high total cost to distinguish between $\mathcal{F}_{\texttt{adv}}$ and $\mathcal{F}_T$ when T is a random set.

For every $X \subseteq U$ we fix an arbitrary set $Q_X \subseteq U \setminus X$ such that $|Q_X| = \max\{\lfloor \beta \cdot k^* \rfloor + 1 - |X|, \ 0\}$. Consequently, it holds that $X \cup Q_X \in \mathcal{F}_{\texttt{adv}}$. We define an extension oracle $\texttt{Ext}_{\texttt{adv}}$ via $\texttt{Ext}_{\texttt{adv}}(X, \ell) := Q_X$ for all $X \subseteq U$ and $\ell \geq 0$.

Claim 5.5. *For every $j \in [s]$ it holds that $\texttt{Ext}_{\texttt{adv}}$ is an α_j-extension oracle for U and $\mathcal{F}_{\texttt{adv}}$.*

Proof. Let $X \subseteq U$ and $\ell \in \mathbb{N}$. It holds that $X \cup \texttt{Ext}_{\texttt{adv}}(X) = X \cup Q_X \in \mathcal{F}_{\texttt{adv}}$. Furthermore, if there is an ℓ-extension S of X, then $|X| \geq \lfloor \beta k^* \rfloor + 1 - \ell$. Hence, $|Q_X| \leq \lfloor \beta \cdot k^* \rfloor + 1 - |X| \leq \ell \leq \alpha_j \cdot \ell$. That is, Q_X is an $(\alpha_j \cdot \ell)$ extension of X. So $\texttt{Ext}_{\texttt{adv}}$ is an α_j-extension oracle. ⌟

We define the *coverage* of an oracle query (X, ℓ) to the j-th oracle (i.e., to the α_j-extension oracle) by

$$\texttt{coverage}_j(X, \ell) := \begin{cases} \emptyset & |X| + \alpha_j \cdot \ell \geq \lfloor \beta \cdot k^* \rfloor + 1, \\ \{S \subseteq U \mid |S| = k^*, |S \setminus X| \leq \ell\} & \text{otherwise.} \end{cases} \tag{22}$$

Intuitively speaking, $\texttt{coverage}_j(X, \ell)$ contains all subsets $\subseteq U$ of cardinality k^* such that a deterministic α_j-extension oracle for $\mathcal{F}_S$ cannot return $\texttt{Ext}_{\texttt{adv}}(X, \ell)$ to the query (X, ℓ). Given a set $W \subseteq 2^U \times \mathbb{N}$ of queries we define $\texttt{coverage}_j(W) := \bigcup_{(X, \ell) \in W} \texttt{coverage}_j(X, \ell)$.

Claim 5.6. *Let $j \in [s]$ and suppose $(X, \ell) \in 2^U \times \mathbb{N}$ such that $\texttt{coverage}_j(X, \ell) \neq \emptyset$. Then*

$$n - \ell \geq \lfloor \beta \cdot k^* - \alpha_j \cdot \ell \rfloor \geq |X| \geq M^*_{\alpha_j, \beta} \cdot k^*.$$

Proof. Since $\texttt{coverage}_j(X, \ell) \neq \emptyset$ it holds that $|X| + \alpha_j \cdot \ell < \lfloor \beta \cdot k^* \rfloor + 1$. Therefore

$$|X| < \lfloor \beta \cdot k^* \rfloor + 1 - \alpha_j \cdot \ell \leq \beta \cdot k^* - \alpha_j \cdot \ell \leq n - \ell,$$

which also implies $|X| \leq \lfloor \beta \cdot k^* - \alpha_j \cdot \ell \rfloor$. If $\alpha_j \leq \beta$ then $M^*_{\alpha_j, \beta} = 0$, and the trivial inequality $|X| \geq 0$ completes the proof of the claim. So we assume that $\alpha_j > \beta \geq 1$ for the remainder of the proof.

Using $\texttt{coverage}_j(X, \ell) \neq \emptyset$ once more, by (22) there is some $S \subseteq U$ such that $|S| = k^*$ and $|S \setminus X| \leq \ell$. So $|X| \geq k^* - \ell$ and we get that

$$\begin{aligned}
\lfloor \beta \cdot k^* - \alpha_j \cdot \ell \rfloor &\geq |X| \\
&\geq \frac{\alpha_j}{\alpha_j - 1} \cdot |X| - \frac{1}{\alpha_j - 1} \cdot |X| \\
&\geq \frac{\alpha_j}{\alpha_j - 1} \cdot (k^* - \ell) - \frac{1}{\alpha_j - 1} \cdot (\beta \cdot k^* - \alpha_j \cdot \ell) \\
&= \frac{\alpha_j - \beta}{\alpha_j - 1} \cdot k^* \\
&= M_{\alpha_j, \beta} \cdot k^*.
\end{aligned}$$

$\lrcorner$

We assume the input for $\mathcal{A}$ consists of the set U, a collection of s extension oracles $\texttt{Ext}_1, \dots, \texttt{Ext}_s$, where $\texttt{Ext}_j$ is an α_j-extension oracle, and a bit-string $b \in \{0, 1\}^{q(n+1)}$ where q is an arbitrary function. The bit-string b serves as the source of randomness, and we assume the algorithm is deterministic given U, the oracles $\texttt{Ext}_1, \dots, \texttt{Ext}_s$ and b. We write $\mathcal{A}(U, \texttt{Ext}_1, \dots, \texttt{Ext}_s, b) \subseteq U$ to denote the output of $\mathcal{A}$ given U, the oracles $\texttt{Ext}_1, \dots, \texttt{Ext}_s$ and b.

For every $b \in \{0, 1\}^{q(n+1)}$ let $Q_j(b) \subseteq 2^U \times \mathbb{N}$ be the set of queries $\mathcal{A}$ makes to the j-th oracle given the input U, $\texttt{Ext}_{\texttt{adv}}$ as the j-th oracle for every $j \in [s]$, and b.

Claim 5.7. *Let $b \in \{0, 1\}^{q(n+1)}$. Then*

$$\texttt{cost}_{\mathcal{A}}(n + 1) \geq f_{\mathcal{L}, \beta}(n) \cdot \frac{\sum_{j \in [s]} \left| \texttt{coverage}_j(Q_j(b)) \right|}{(n + 1) \cdot \binom{n}{k^*} \cdot \max_{j \in [s]} c_j}.$$

Proof. For every $j \in [s]$ it holds that

$$\left|\text{coverage}_j(Q_j(b))\right| \leq \sum_{(X,\ell) \in Q_j(b)} \left|\text{coverage}_j(X,\ell)\right|$$

$$= \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} \left|\text{coverage}_j(X,\ell)\right|$$

$$= \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} \sum_{y=k^*-\ell}^{k^*} \binom{|X|}{y} \cdot \binom{n+1-|X|}{k^*-y} \tag{23}$$

$$\leq (n+1) \cdot \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} \sum_{y=k^*-\ell}^{k^*} \binom{|X|}{y} \cdot \binom{n-|X|}{k^*-y}.$$

The second equality follows from a simple counting argument and the definition of coverage_j in (22). The second inequality holds since $\binom{m+1}{r} \leq (m+1) \cdot \binom{m}{r}$ if $m \geq r \geq 0$. Observe that $n-|X| \geq \ell \geq k^*-y$ for every $(X,\ell) \in Q_j(b)$ such that $\text{coverage}_j(X,\ell) \neq \emptyset$, and every $k^* \leq y \leq k^* - \ell$ using Claim 5.6.

Plugging the formula for hyper from (2) into (23) we obtain

$$\left|\text{coverage}_j(Q_j(b))\right| \leq (n+1) \cdot \binom{n}{k^*} \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} \text{hyper}(n,k^*,|X|,k^*-\ell)$$

$$\leq (n+1) \cdot \binom{n}{k^*} \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} \text{hyper}\left(n,k^*,\lfloor \beta k^* - \alpha_j \ell \rfloor, k^*-\ell\right)$$

$$\leq (n+1) \cdot \binom{n}{k^*} \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} \text{hyper}\left(n,k^*,\lfloor \beta k^* - \alpha_j \ell \rfloor, k^* - \frac{\beta k^* - \lfloor \beta k^* - \alpha_j \ell \rfloor}{\alpha_j}\right) \tag{24}$$

$$= (n+1) \cdot \binom{n}{k^*} \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} (c_j)^{\frac{\beta k^* - \lfloor \beta k^* - \alpha_j \ell \rfloor}{\alpha_j}} \cdot \frac{1}{G_j\left(k^*, \lfloor \beta k^* - \alpha_j \ell \rfloor\right)}.$$

The second inequality holds since $|X| \geq \lfloor \beta \cdot k^* - \alpha_j \ell \rfloor$ by Claim 5.6. The third inequality follows from $-\ell \geq -\frac{\beta k^* - \lfloor \beta k^* - \alpha_j \cdot \ell \rfloor}{\alpha_j}$. The last equality follows from the definition of G_j in (20).

By Claim 5.6 it holds that $M^*_{\alpha_j,\beta} \cdot k^* \leq \lfloor \beta \cdot k^* - \alpha_j \cdot \ell \rfloor \leq \beta \cdot k^*$ for every $(X,\ell) \in 2^U \times \mathbb{N}$ for which $\text{coverage}_j(X,\ell) \neq \emptyset$. Hence,

$$G_j(k^*, \lfloor \beta \cdot k^* - \alpha_j \cdot \ell \rfloor) \geq G_j(k^*, t^*_j(k^*)) \geq G_{j^*}(k^*, t^*) = f_{\mathcal{L},\beta}(n) \tag{25}$$

for every $(X,\ell) \in 2^U \times \mathbb{N}$ such that $\text{coverage}_j(X,\ell) \neq \emptyset$. Combining (24) with (25) we have

$$\left|\text{coverage}_j(Q_j(b))\right| \leq (n+1) \cdot \binom{n}{k^*} \sum_{\substack{(X,\ell) \in Q_j(b) \text{ s.t.} \\ \text{coverage}_j(X,\ell) \neq \emptyset}} (c_j)^{\frac{\beta k^* - \lfloor \beta k^* - \alpha_j \ell \rfloor}{\alpha_j}} \cdot \frac{1}{f_{\mathcal{L},\beta}(n)}$$

$$\leq (n+1) \cdot \binom{n}{k^*} \sum_{(X,\ell) \in Q_j(b)} (c_j)^{\ell+1} \cdot \frac{1}{f_{\mathcal{L},\beta}(n)}.$$

Since the last inequality holds for every $j \in [s]$ we get

$$\sum_{j \in [s]} \left|\text{coverage}_j(Q_j(b))\right| \leq (n+1) \cdot \binom{n}{k^*} \cdot \left(\max_{j \in [s]} c_j\right) \cdot \sum_{j \in [s]} \sum_{(X,\ell) \in Q_j(b)} (c_j)^{\ell} \cdot \frac{1}{f_{\mathcal{L},\beta}(n)}$$

24

and thus,

$$\text{cost}_{\mathcal{A}}(n+1) \ \geq \ \sum_{j\in[s]} \sum_{(X,\ell)\in Q_j(b)} (c_j)^\ell \ \geq \ f_{\mathcal{L},\beta}(n) \cdot \frac{\sum_{j\in[s]} \left|\text{coverage}_j(Q_j(b))\right|}{(\max_{j\in[s]} c_j) \cdot (n+1) \cdot \binom{n}{k*}}. \qquad \lrcorner$$

By Claim 5.7, in order to lower bound the cost of $\mathcal{A}$, we only need to provide a lower bound on $\sum_{j\in[s]} \left|\text{coverage}_j(Q_j(b))\right|$ for some $b \in \{0,1\}^{q(n+1)}$. For every $T \subseteq U$ such that $|T| = k^*$ and every $j \in [s]$ we define an α_j-extension oracle $\text{Ext}_{T,j}$ for U and $\mathcal{F}_T$ by

$$\text{Ext}_{T,j}(X,\ell) \coloneqq \begin{cases} T\setminus X & T\in\text{coverage}_j(X,\ell), \\ \text{Ext}_{\text{adv}}(X,\ell) & \text{otherwise.} \end{cases}$$

Claim 5.8. *For every $T \subseteq U$ such that $|T| = k^*$ and $j \in [s]$ it holds that $\text{Ext}_{T,j}$ is an α_j-extension oracle for U and the set system $\mathcal{F}_T$.*

Proof. Let $(X,\ell) \in 2^U \times \mathbb{N}$. If $T \in \text{coverage}_j(X,\ell)$ then $X\cup\text{Ext}_{T,j}(X,\ell) = X\cup(T\setminus X) = X\cup T \in \mathcal{F}_T$. Otherwise $X \cup \text{Ext}_{T,j}(X,\ell) = X \cup \text{Ext}_{\text{adv}}(X,\ell) \in \mathcal{F}_{\text{adv}} \subseteq \mathcal{F}_T$. That is, $X \cup \text{Ext}_{T,j}(X,\ell) \in \mathcal{F}_T$ in all cases.

Suppose X has an ℓ-extension S with respect to U and $\mathcal{F}_T$. To complete the proof we need to show that $|\text{Ext}_{T,j}(X,\ell)| \leq \alpha_j \cdot \ell$. We distinguish the following two cases.

- If $|X| + \alpha_j \cdot \ell \geq \lfloor\beta k^*\rfloor + 1$ then $\text{coverage}_j(X,\ell) = \emptyset$ by (22), and thus

$$|\text{Ext}_{T,j}(X,\ell)| \ = \ |\text{Ext}_{\text{adv}}(X,\ell)| \ = \ \max\left\{\lfloor\beta\cdot k^*\rfloor + 1 - |X|, 0\right\} \ \leq \ \max\{\alpha_j\cdot\ell, 0\} \ \leq \ \alpha_j\cdot\ell.$$

- Otherwise $|X| + \alpha_j \cdot \ell < \lfloor\beta k^*\rfloor + 1$ and we have that $|S \cup X| \leq |X| + \ell < \lfloor\beta k^*\rfloor + 1$. This means $S \cup X \notin \mathcal{F}_{\text{adv}}$. Since $S \cup X \in \mathcal{F}_T$, we conclude that $T \subseteq S \cup X$. So $T \setminus X \subseteq S$ and $|T \setminus X| \leq |S| \leq \ell$. Since $|T| = k^*$ we conclude that $T \in \text{coverage}_j(X,\ell)$. It follows that $\text{Ext}_{T,j}(X,\ell) = T \setminus X$, and

$$|\text{Ext}_{T,j}(X,\ell)| = |T \setminus X| \leq |S| \leq \ell \leq \alpha_j \cdot \ell. \qquad \lrcorner$$

Now, let $b^* \in \{0,1\}^{q(n+1)}$ be the bit-string for which $\sum_{j\in[s]} \left|\text{coverage}_j(Q_j(b^*))\right|$ is maximal.

Claim 5.9. *It holds that*

$$\sum_{j\in[s]} \left|\text{coverage}_j(Q_j(b^*))\right| \geq \frac{1}{2} \cdot \binom{n+1}{k^*}.$$

Proof. Consider the execution of $\mathcal{A}$ with the universe U, the oracles $\text{Ext}_{T,1}, \ldots, \text{Ext}_{T,s}$ and the bit-string b, where $T \subseteq U$ and $|T| = k^*$. Unless $T \in \bigcup_{j\in[s]} \text{coverage}_j(Q_j(b))$ the execution is identical to the execution of $\mathcal{A}$ with the universe U, the oracles $\text{Ext}_{\text{adv}}, \ldots, \text{Ext}_{\text{adv}}$ and b. Hence, $\mathcal{A}$ has to return a set $S \in \mathcal{F}_{\text{adv}}$ (otherwise it violates the correctness requirement for the latter execution), and thus $|S| \geq \lfloor\beta k^*\rfloor + 1$. It follows that

$$|\mathcal{A}(U, \text{Ext}_{T,1}, \ldots, \text{Ext}_{T,s}, b)| \leq \beta k^* \implies T \in \bigcup_{j\in[s]} \text{coverage}_j(Q_j(b)). \qquad (26)$$

We define two independent random variables. Let $R \subseteq U$ be a uniformly distributed random set

of cardinality k^*, and let $r \in \{0,1\}^{q(n+1)}$ be a uniformly distributed bit-string. Using (26) we get

$$
\Pr\left(|\mathcal{A}(U, \mathtt{Ext}_{R,1}, \ldots, \mathtt{Ext}_{R,s}, r)| \leq \beta k^*\right) \ \leq \ \Pr\left(R \in \bigcup_{j\in[s]} \mathtt{coverage}_j(Q_j(r))\right)
$$

$$
= \sum_{b\in\{0,1\}^{q(n+1)}} \Pr(r=b) \cdot \Pr\left(R \in \bigcup_{j\in[s]} \mathtt{coverage}_j(Q_j(r)) \,\middle|\, b=r\right)
$$

$$
= \sum_{b\in\{0,1\}^{q(n+1)}} \Pr(r=b) \cdot \Pr\left(R \in \bigcup_{j\in[s]} \mathtt{coverage}_j(Q_j(b))\right) \tag{27}
$$

$$
\leq \sum_{b\in\{0,1\}^{q(n+1)}} \Pr(r=b) \cdot \frac{\sum_{j\in[s]} \left|\mathtt{coverage}_j(Q_j(b))\right|}{\binom{n+1}{k^*}}
$$

$$
\leq \sum_{b\in\{0,1\}^{q(n+1)}} \Pr(r=b) \cdot \frac{\sum_{j\in[s]} \left|\mathtt{coverage}_j(Q_j(b^*))\right|}{\binom{n+1}{k^*}}
$$

$$
= \frac{\sum_{j\in[s]} \left|\mathtt{coverage}_j(Q_j(b^*))\right|}{\binom{n+1}{k^*}}.
$$

The second equality holds since r is independent of R. Furthermore,

$$
\Pr\left(|\mathcal{A}(U, \mathtt{Ext}_{R,1}, \ldots, \mathtt{Ext}_{R,s}, r)| \leq \beta k^*\right)
$$

$$
= \sum_{T\subseteq U \text{ s.t. } |T|=k^*} \Pr(R=T) \cdot \Pr\left(|\mathcal{A}(U, \mathtt{Ext}_{R,1}, \ldots, \mathtt{Ext}_{R,s}, r)| \leq \beta k^* \mid R=T\right)
$$

$$
= \sum_{T\subseteq U \text{ s.t. } |T|=k^*} \Pr(R=T) \cdot \Pr\left(|\mathcal{A}(U, \mathtt{Ext}_{T,1}, \ldots, \mathtt{Ext}_{T,s}, r)| \leq \beta k^*\right) \tag{28}
$$

$$
\geq \sum_{T\subseteq U \text{ s.t. } |T|=k^*} \Pr(R=T) \cdot \frac{1}{2} \ \geq \ \frac{1}{2}
$$

By (27) and (28) it holds that $\frac{\sum_{j\in[s]} \left|\mathtt{coverage}_j(Q_j(b^*))\right|}{\binom{n+1}{k^*}} \geq \frac{1}{2}$ and the claim immediately follows.   ⌟

By Claims 5.7 and 5.9 it holds that

$$
\mathtt{cost}_{\mathcal{A}}(n+1) \geq \ f_{\mathcal{L},\beta}(n) \cdot \frac{\sum_{j\in[s]} \left|\mathtt{coverage}_j(Q_j(b^*))\right|}{(n+1) \cdot \binom{n}{k^*} \cdot \max_{j\in[s]} c_j}
$$

$$
\geq \ f_{\mathcal{L},\beta}(n) \cdot \frac{\frac{1}{2} \cdot \binom{n+1}{k^*}}{(n+1) \cdot \binom{n}{k^*} \cdot \max_{j\in[s]} c_j}
$$

$$
\geq \ \frac{f_{\mathcal{L},\beta}(n)}{2 \cdot (n+1) \cdot \max_{j\in[s]} c_j}.
$$

$\square$

Proof of Lemma 2.8. Let $\mathcal{A}$ be a β-approximation algorithm for $\mathcal{L}$-SUB. By Lemmas 2.11 and 5.4 it holds that

$$
\mathtt{cost}_{\mathcal{A}}(n+1) \geq \frac{f_{\mathcal{L},\beta}(n)}{2 \cdot (n+1) \cdot \max_{(\alpha,c)\in\mathcal{L}} c} \geq n^{-\mathcal{O}(1)} \cdot (\mathtt{amls}(\mathcal{L},\beta))^n \geq n^{-\mathcal{O}(1)} \cdot (\mathtt{amls}(\mathcal{L},\beta))^{n+1}
$$

$$
\geq n^{-\mathcal{O}(1)} \cdot f_{\mathcal{L},\beta}(n+1)
$$

for every $n \geq 1$. $\square$

6 From Discrete to Continuous Optimization

Lemma 2.11 shows that $f_{\mathcal{L},\beta}(n) \approx (\mathtt{amls}(\mathcal{L},\beta))^n$ up to polynomial factors. While $f_{\mathcal{L},\beta}(n)$ (5) is defined via maximum and minimum operations over a discrete set of values, the value of $\mathtt{amls}(\mathcal{L},\beta)$ (11) is the outcome of continuous maximization and minimization. The proof utilizes basic estimation of binomial coefficient using entropy and bounded-difference properties of the entropy function.

The value of τ in the definition of $\mathtt{amls}$ (11) corresponds to $\frac{t}{n}$ in the formula of f (5). We note that the range of $\frac{t}{n}$ in (5) may differ from the range of τ in (11). Part of the proof is dedicated for showing this difference is insignificant.

We first prove that $f_{\mathcal{L},\beta}(n) \lesssim (\mathtt{amls}(\mathcal{L},\beta))^n$ in Lemma 6.6, and subsequently show that $f_{\mathcal{L},\beta}(n) \gtrsim (\mathtt{amls}(\mathcal{L},\beta))^n$ in Lemma 6.7. The proof of Lemma 6.6 is technically easier. This stems from the fact that restricting the range of t, extending the range of k and lower-bounding the value of $\mathtt{hyper}$ (as it appears in (5)) are trivial in this direction of the inequality, but not in the other. We also note that special cases of the inequality $f_{\mathcal{L},\beta}(n) \lesssim (\mathtt{amls}(\mathcal{L},\beta))^n$ for $\mathcal{L} = \{(\beta,c)\}$ implicitly appear in previous works on (Approximate) Monotone Local Search [26, 31] as part of the analysis of the algorithm. The opposite direction, $f_{\mathcal{L},\beta}(n) \gtrsim (\mathtt{amls}(\mathcal{L},\beta))^n$, is central for the correctness of the lower bounds in Lemma 2.8 and theorem 2.10, but has no algorithmic implications. As such, this direction of the inequality was irrelevant to the previous works which only provided algorithmic results.

Recall $\mathcal{H}(x) = -x \ln(x) - (1-x) \ln(1-x)$. With slight abuse of notation we define $0 \cdot \mathcal{H}\left(\frac{a}{0}\right) = 0$. Our proofs utilize the following bound on binomial coefficients (see, e.g., [19, Example 11.1.3]):

$$\frac{1}{n+1} \cdot \exp\left(n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right) \leq \binom{n}{k} \leq \exp\left(n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right) \tag{29}$$

for all $n, k \in \mathbb{N}$ such that $0 \leq k \leq n$. Furthermore, we utilize the following technical lemma which follows from [26].

Lemma 6.1. *For all $0 \leq b \leq a \leq n$, $d > 1$ and $\varepsilon, \delta \in [-d, d]$ such that $0 \leq b + \delta \leq a + \varepsilon$, we have*

$$\left| a \cdot \mathcal{H}\left(\frac{b}{a}\right) - (a + \varepsilon) \cdot \mathcal{H}\left(\frac{b+\delta}{a+\varepsilon}\right) \right| = \mathcal{O}(d \cdot \log(n)).$$

For every $\alpha, \beta \geq 1$ we define $x_{\alpha,\beta}(k,t) = \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}$. We utilize the following technical lemmas as part of the proofs of Lemmas 6.6 and 6.7.

Lemma 6.2. *Let $\alpha, \beta \geq 1$, $n \in \mathbb{N}$, $k \in \left[0, \frac{n}{\beta}\right]$, $t \geq 0$ and $y \geq x_{\alpha,\beta}(k,t)$. Then $k - y \leq n - t$.*

Proof. We have

$$n - t \geq \beta k - t \geq \frac{\beta k - t}{\alpha} = k - k + \frac{\beta k}{\alpha} - \frac{t}{\alpha} = k - x_{\alpha,\beta}(k,t) \geq k - y. \qquad \square$$

Lemma 6.3. *Let $\alpha, \beta \geq 1$, $n \in \mathbb{N}$, $k \in \left[0, \frac{n}{\beta}\right]$, and $t \in \left[M^*_{\alpha,\beta} k, \beta k\right]$. Then $x_{\alpha,\beta}(k,t) \leq \min\{k, t\}$.*

Proof. Since $t \geq 0$ we have

$$x_{\alpha,\beta}(k,t) = \left(1 - \frac{\beta}{\alpha}\right)k + \frac{t}{\alpha} \leq \left(1 - \frac{\beta}{\alpha}\right)k \leq k.$$

For the second part, we consider the following two cases.

- If $\beta \geq \alpha$ then

$$x_{\alpha,\beta}(k,t) = \left(1 - \frac{\beta}{\alpha}\right)k + \frac{t}{\alpha} \leq \frac{t}{\alpha} \leq t.$$

- Otherwise $\beta < \alpha$ and we have $t \geq M^*_{\alpha,\beta} \cdot k = \frac{\alpha - \beta}{\alpha - 1} \cdot k$. Thus,

$$x_{\alpha,\beta}(k,t) = \left(1 - \frac{\beta}{\alpha}\right)k + \frac{t}{\alpha} = \frac{\alpha - 1}{\alpha} \cdot \frac{\alpha - \beta}{\alpha - 1} \cdot k + \frac{t}{\alpha} \leq \frac{\alpha - 1}{\alpha} \cdot t + \frac{t}{\alpha} = t.$$

In both cases $x_{\alpha,\beta}(k,t) \leq t$ which completes the proof. $\qquad\square$

Lemma 6.4. *Let* $\alpha, \beta \geq 1$, $n \in \mathbb{N}$, $k \in \left[0, \frac{n}{\beta}\right]$, *and* $t \in \left[M^*_{\alpha,\beta}k, \beta k\right]$. *Then* $x_{\alpha,\beta}(k,t) \geq \frac{kt}{n}$ *if and only if* $t \geq M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n$.

Proof. We consider the following two cases.

- If $\alpha < \beta$ it holds that

$$x_{\alpha,\beta}(k,t) \geq \frac{kt}{n} \quad \Longleftrightarrow \quad \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha} \geq \frac{kt}{n} \quad \Longleftrightarrow \quad \left(1 - \frac{\beta}{\alpha}\right) \cdot k \geq t \cdot \left(\frac{k}{n} - \frac{1}{\alpha}\right).$$

  Since $k \leq \frac{n}{\beta}$ we conclude that $\frac{k}{n} - \frac{1}{\alpha} < 0$. So

$$x_{\alpha,\beta}(k,t) \geq \frac{kt}{n} \quad \Longleftrightarrow \quad t \geq \frac{\left(1 - \frac{\beta}{\alpha}\right)}{\frac{k}{n} - \frac{1}{\alpha}} \cdot k = \frac{\beta - \alpha}{1 - \alpha \cdot \frac{k}{n}} \cdot \frac{k}{n} \cdot n = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n.$$

- Otherwise $\alpha \geq \beta$ and we have $M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n = M^*_{\alpha,\beta} \cdot k$. Thus, we need to prove that $x_{\alpha,\beta}(k,t) \geq \frac{kt}{n}$ holds unconditionally. Indeed,

$$
\begin{aligned}
x_{\alpha,\beta}(k,t) &= \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha} \\
&= \left(1 - \frac{\beta}{\alpha}\right) \cdot k + t\left(-\frac{k}{n} + \frac{1}{\alpha}\right) + \frac{kt}{n} \\
&\geq \left(1 - \frac{\beta}{\alpha}\right) \cdot k + t\left(-\frac{1}{\beta} + \frac{1}{\alpha}\right) + \frac{kt}{n} \\
&\geq \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \beta \cdot k\left(-\frac{1}{\beta} + \frac{1}{\alpha}\right) + \frac{kt}{n} = \frac{kt}{n},
\end{aligned}
$$

  where the first inequality follows from $k \leq \frac{n}{\beta}$, and the second inequality holds since $t \leq \beta k$ and $-\frac{1}{\beta} + \frac{1}{\alpha} \leq 0$. $\qquad\square$

Finally, we use the following relation between $M^*_{\alpha,\beta}$ and $M_{\alpha,\beta}$.

Lemma 6.5. *Let* $\alpha, \beta \geq 1$, $n \in \mathbb{N}$, $k \in \left[0, \frac{n}{\beta}\right]$. *Then* $M^*_{\alpha,\beta} \cdot k \leq M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n \leq \beta k$.

Proof. We first show that $M^*_{\alpha,\beta} \cdot k \leq M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n$. If $\alpha \leq \beta$ it holds that $M^*_{\alpha,\beta} = 0$. Since $M_{\alpha,\beta}\left(\frac{k}{n}\right) \geq 0$ it follows that $M^*_{\alpha,\beta} \cdot k = 0 \leq M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n$. Otherwise $\alpha > \beta$ and we have

$$M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n = \frac{\alpha - \beta}{\alpha - 1} \cdot \frac{k}{n} \cdot n = M^*_{\alpha,\beta} \cdot k,$$

where the last equality follows from the definition of $M^*_{\alpha,\beta}$ (4).

Next, we show that $M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n \leq \beta k$. If $\alpha < \beta$ it holds that

$$M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n = \frac{\beta - \alpha}{1 - \alpha \cdot \frac{k}{n}} \cdot \frac{k}{n} \cdot n \leq \frac{\beta - \alpha}{1 - \alpha \cdot \frac{1}{\beta}} \cdot k = \beta \cdot k,$$

where the inequality follows from $k \leq \frac{n}{\beta}$. Otherwise $\beta < \alpha$ and we have

$$M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n = \frac{\alpha - \beta}{\alpha - 1} \cdot \frac{k}{n} \cdot n \leq \frac{\alpha - 1}{\alpha - 1} \cdot \frac{k}{n} \cdot n = k \leq \beta k.$$

Finally, if $\alpha = \beta$, we have $M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n = 0 \cdot n \leq \beta k$. $\qquad\square$

The next lemma show the second inequality of Lemma 2.11.

Lemma 6.6. *For every $\beta \geq 1$ and specification list $\mathcal{L}$ it holds that*

$$f_{\mathcal{L},\beta}(n) \leq n^{\mathcal{O}(1)} \cdot (\mathtt{amls}(\mathcal{L},\beta))^n.$$

Proof. Let $n \geq 1$. For every $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$ it holds that

$$\min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot c\right)}{\mathtt{hyper}(n, k, t, x_{\alpha,\beta}(k,t))} = \min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot c\right)}{\sum_{y=\lceil x_{\alpha,\beta}(k,t)\rceil}^{\min\{k,t\}} \frac{\binom{t}{y} \cdot \binom{n-t}{k-y}}{\binom{n}{k}}}$$

$$\leq \min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot c\right)}{\left(\frac{\binom{t}{\max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}} \cdot \binom{n-t}{k-\max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}}}{\binom{n}{k}}\right)} \tag{30}$$

$$\leq (n+1)^2 \cdot \min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k] \cap \mathbb{N}} \exp\left(\frac{\beta k - t}{\alpha} \cdot c - t \cdot \mathcal{H}\left(\frac{\max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}}{t}\right)\right.$$

$$\left. - (n-t) \cdot \mathcal{H}\left(\frac{k - \max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}}{n-t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right)$$

The first equality follows from the definition of $\mathtt{hyper}$ (2). The first inequality follows from selecting $y = \max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}$. Note that the resulting expression is well defined by Lemmas 6.2 and 6.3. The last inequality follows from (29).

We can use Lemma 6.1 to avoid the rounding of the values of $x_{\alpha,\beta}$ as well as extending the range of t in (30). That is, for every $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$ it holds that

$$\min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot c\right)}{\mathtt{hyper}(n, k, t, x_{\alpha,\beta}(k,t))}$$

$$\leq n^{\mathcal{O}(1)} \cdot \min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k] \cap \mathbb{N}} \exp\left(\frac{\beta k - t}{\alpha} \cdot c - t \cdot \mathcal{H}\left(\frac{\max\{x_{\alpha,\beta}(k,t),\, 0\}}{t}\right)\right.$$

$$\left. - (n-t) \cdot \mathcal{H}\left(\frac{k - \max\{x_{\alpha,\beta}(k,t),\, 0\}}{n-t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right)$$

$$\leq n^{\mathcal{O}(1)} \cdot \min_{t \in [M_{\alpha,\beta}^* \cdot k, \beta k]} \exp\left(\frac{\beta k - t}{\alpha} \cdot c - t \cdot \mathcal{H}\left(\frac{\max\{x_{\alpha,\beta}(k,t),\, 0\}}{t}\right)\right.$$

$$\left. - (n-t) \cdot \mathcal{H}\left(\frac{k - \max\{x_{\alpha,\beta}(k,t),\, 0\}}{n-t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right) \tag{31}$$

$$\leq n^{\mathcal{O}(1)} \cdot \min_{t \in [M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n, \beta k]} \exp\left(\frac{\beta k - t}{\alpha} \cdot c - t \cdot \mathcal{H}\left(\frac{\max\{x_{\alpha,\beta}(k,t),\, 0\}}{t}\right)\right.$$

$$\left. - (n-t) \cdot \mathcal{H}\left(\frac{k - \max\{x_{\alpha,\beta}(k,t),\, 0\}}{n-t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right)$$

$$\leq n^{\mathcal{O}(1)} \cdot \min_{t \in [M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n, \beta k]} \exp\left(\frac{\beta k - t}{\alpha} \cdot c - t \cdot \mathcal{H}\left(\frac{x_{\alpha,\beta}(k,t)}{t}\right)\right.$$

$$\left. - (n-t) \cdot \mathcal{H}\left(\frac{k - x_{\alpha,\beta}(k,t)}{n-t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right)$$

Observe that in the third expression the range of t is not restricted to integers. The range of t was further changed in the forth expression using Lemma 6.5. The last inequality follow from Lemma 6.4 (trivially, $\frac{kt}{n} \geq 0$).

Observe that
$$\frac{x_{\alpha,\beta}(k,t)}{t} = \left(1 - \frac{\beta}{\alpha}\right)\frac{k}{t} + \frac{1}{\alpha} = \gamma_{\alpha,\beta}\left(\frac{k}{n}, \frac{t}{n}\right) \tag{32}$$

unless $t = 0$, and

$$\frac{k - x_{\alpha,\beta}(k,t)}{n-t} = \frac{k - \left(1 - \frac{\beta}{\alpha}\right)k - \frac{t}{\alpha}}{n-t} = \frac{\frac{\beta}{\alpha} \cdot k - \frac{t}{\alpha}}{n-t} = \delta_{\alpha,\beta}\left(\frac{k}{n}, \frac{t}{n}\right) \tag{33}$$

unless $t = 1$.

Using (32) and (33) we can simplify the expression in (31) and obtain

$$\min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot c\right)}{\mathtt{hyper}(n,k,t,x_{\alpha,\beta}(k,t))}$$

$$\leq n^{\mathcal{O}(1)} \cdot \min_{t \in \left[M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n, \beta k\right]} \exp\left(\frac{\beta\frac{k}{n} - \frac{t}{n}}{\alpha} \cdot c - \frac{t}{n} \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}\left(\frac{k}{n}, \frac{t}{n}\right)\right)\right.$$
$$\left. - \left(1 - \frac{t}{n}\right) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}\left(\frac{k}{n}, \frac{t}{n}\right)\right) + \mathcal{H}\left(\frac{k}{n}\right)\right)^n \tag{34}$$

$$= n^{\mathcal{O}(1)} \cdot \min_{\tau \in \left[M_{\alpha,\beta}\left(\frac{k}{n}\right), \beta\frac{k}{n}\right]} \left(g_{\alpha,\beta,c}\left(\frac{k}{n}, \tau\right)\right)^n.$$

By incorporating (34) into the formula of $f_{\mathcal{L},\beta}$ (5) we get

$$f_{\mathcal{L},\beta}(n) = \max_{k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}} \min_{(\alpha,c) \in \mathcal{L}} \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta \cdot k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}\left(n,k,t,\left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}\right)}$$

$$\leq n^{\mathcal{O}(1)} \cdot \max_{k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}} \min_{(\alpha,c) \in \mathcal{L}} \min_{\tau \in \left[M_{\alpha,\beta}\left(\frac{k}{n}\right), \beta\frac{k}{n}\right]} \exp\left(g_{\alpha,\beta,c}\left(\frac{k}{n}, \tau\right)\right)^n$$

$$\leq n^{\mathcal{O}(1)} \cdot \max_{k \in \left[0, \frac{n}{\beta}\right]} \min_{(\alpha,c) \in \mathcal{L}} \min_{\tau \in \left[M_{\alpha,\beta}\left(\frac{k}{n}\right), \beta\frac{k}{n}\right]} \exp\left(g_{\alpha,\beta,c}\left(\frac{k}{n}, \tau\right)\right)^n$$

$$= n^{\mathcal{O}(1)} \cdot \max_{\kappa \in \left[0, \frac{1}{\beta}\right]} \min_{(\alpha,c) \in \mathcal{L}} \min_{\tau \in \left[M_{\alpha,\beta}(\kappa), \beta\kappa\right]} \exp\left(g_{\alpha,\beta,c}\left(\kappa, \tau\right)\right)^n$$

$$= n^{\mathcal{O}(1)} \cdot \left(\mathtt{amls}(\mathcal{L}, \beta)\right)^n.$$

The second inequality simply extended the range of values k can takes, and the second equality substituted k with $\kappa \cdot n$. $\qquad \square$

Lemma 6.7. *For every $\beta \geq 1$ and specification list $\mathcal{L}$ it holds that*
$$f_{\mathcal{L},\beta}(n) \geq n^{\mathcal{O}(1)} \cdot \left(\mathtt{amls}(\mathcal{L}, \beta)\right)^n.$$

Proof. Let $n \in \mathbb{N}$. For every $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$ it holds that

$$\min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}(n,k,t,x_{\alpha,\beta}(k,t))} = \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\sum_{y=\lceil x_{\alpha,\beta}(k,t)\rceil}^{\min\{k,t\}} \frac{\binom{t}{y} \cdot \binom{n-t}{k-y}}{\binom{n}{k}}}$$

$$\geq \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{n \cdot \max_{y \in \left[\max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}, \ \min\{k,t\}\right] \cap \mathbb{N}} \frac{\binom{t}{y} \cdot \binom{n-t}{k-y}}{\binom{n}{k}}} \tag{35}$$

$$= \frac{1}{n} \cdot \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, \beta k\right] \cap \mathbb{N}} \min_{y \in \left[\max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\}, \ \min\{k,t\}\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\frac{\binom{t}{y} \cdot \binom{n-t}{k-y}}{\binom{n}{k}}}.$$

As in the proof of Lemma 6.6, we use (29) to estimate the binomial coefficients in (35). Thus, for all $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$ we have

$$
\min_{t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathrm{hyper}(n, k, t, x_{\alpha,\beta}(k, t))}
$$

$$
\geq\; n^{-\mathcal{O}(1)} \cdot \min_{t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right] \cap \mathbb{N}} \;\; \min_{y \in \left[\max\{\lceil x_{\alpha,\beta}(k,t)\rceil, 0\},\; \min\{k, t\}\right] \cap \mathbb{N}}
$$

$$
\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c - t \cdot \mathcal{H}\left(\frac{y}{t}\right) - (n - t) \cdot \mathcal{H}\left(\frac{k - y}{n - t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right) \tag{36}
$$

$$
\geq\; n^{-\mathcal{O}(1)} \cdot \min_{t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right] \cap \mathbb{N}} \;\; \min_{y \in \left[\max\{x_{\alpha,\beta}(k,t), 0\},\; \min\{k, t\}\right]}
$$

$$
\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c - t \cdot \mathcal{H}\left(\frac{y}{t}\right) - (n - t) \cdot \mathcal{H}\left(\frac{k - y}{n - t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)\right).
$$

For every $k \in \left[0, \frac{n}{\beta}\right]$ and $t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right]$ we define

$$
h_{k,t}(y) = -t \cdot \mathcal{H}\left(\frac{y}{t}\right) - (n - t) \cdot \mathcal{H}\left(\frac{k - y}{n - t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right). \tag{37}
$$

We can use $h_{k,t}$ to rewrite (36) as

$$
\min_{t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathrm{hyper}(n, k, t, x_{\alpha,\beta}(k, t))}
$$

$$
\geq\; n^{-\mathcal{O}(1)} \cdot \min_{t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right] \cap \mathbb{N}} \;\; \min_{y \in \left[\max\{x_{\alpha,\beta}(k,t), 0\},\; \min\{k, t\}\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + h_{k,t}(y)\right) \tag{38}
$$

for all $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$.

Claim 6.8. *For all $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$, $(\alpha, c) \in \mathcal{L}$ and $t \in \left[M_{\alpha,\beta}^* \cdot k, \beta k\right]$ it holds that*

$$
\min_{y \in \left[\max\{x_{\alpha,\beta}(k,t), 0\},\; \min\{k, t\}\right]} h_{k,t}(y) = \begin{cases} h_{k,t}(x_{\alpha,\beta}(k, t)) & \text{if } x_{\alpha,\beta}(k, t) \geq \frac{kt}{n}, \\ 0 & \text{otherwise.} \end{cases}
$$

Furthermore, $h_{k,t}\left(\frac{kt}{n}\right) = 0$.

Proof. We have $\mathcal{H}'(x) = \ln \frac{1 - x}{x}$ where $\mathcal{H}'$ is the first derivative of $\mathcal{H}$. So the first derivative of $h_{k,t}$ is

$$
h_{k,t}'(y) = -t \cdot \frac{1}{t} \cdot \ln\left(\frac{1 - \frac{y}{t}}{\frac{y}{t}}\right) - (n - t) \cdot \frac{-1}{n - t} \cdot \ln\left(\frac{1 - \frac{k - y}{n - t}}{\frac{k - y}{n - t}}\right)
$$

$$
= -\ln\left(\frac{t}{y} - 1\right) + \ln\left(\frac{n - t}{k - y} - 1\right).
$$

It can be easily observed that $h_{k,t}'$ is a monotonically increasing function and thus, $h_{k,t}$ is convex. Furthermore,

$$
h_{k,t}'\left(\frac{kt}{n}\right) = -\ln\left(t \cdot \frac{n}{kt} - 1\right) + \ln\left(\frac{n - t}{k - \frac{kt}{n}} - 1\right) = -\ln\left(\frac{n}{k} - 1\right) + \ln\left(\frac{n}{k} - 1\right) = 0
$$

and

$$
h_{k,t}\left(\frac{kt}{n}\right) = -t \cdot \mathcal{H}\left(\frac{\frac{kt}{n}}{t}\right) - (n - t) \cdot \mathcal{H}\left(\frac{k - \frac{kt}{n}}{n - t}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right)
$$

31

$$= -t \cdot \mathcal{H}\left(\frac{k}{n}\right) - (n-t) \cdot \mathcal{H}\left(\frac{k}{n}\right) + n \cdot \mathcal{H}\left(\frac{k}{n}\right) = 0.$$

So overall, $h_{k,t}(y)$ is a convex function with a global minimum of value 0 at $y = \frac{kt}{n}$. Since $k, t \leq n$ it also holds that $\frac{kt}{n} \leq \min\{k, t\}$. Hence,

$$\min_{y \in \left[\max\{x_{\alpha,\beta}(k,t),0\}, \ \min\{k,t\}\right]} h_{k,t}(y) = \begin{cases} h_{k,t}(x_{\alpha,\beta}(k,t)) & \text{if } x_{\alpha,\beta}(k,t) \geq \frac{kt}{n}, \\ 0 & \text{otherwise.} \end{cases}$$

⌋

By Claim 6.8 and (38), for all $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$, we have

$$\min_{t \in [M^*_{\alpha,\beta} \cdot k, \beta k] \cap \mathbb{N}} \frac{\exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)}{\mathtt{hyper}(n, k, t, x_{\alpha,\beta}(k,t))}$$

$$\geq \ n^{-\mathcal{O}(1)} \cdot \min_{t \in [M^*_{\alpha,\beta} \cdot k, \beta k] \cap \mathbb{N}} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + \begin{cases} h_{k,t}(x_{\alpha,\beta}(k,t)) & \text{if } x_{\alpha,\beta}(k,t) \geq \frac{kt}{n} \\ 0 & \text{otherwise} \end{cases}\right) \quad (39)$$

$$\geq \ n^{-\mathcal{O}(1)} \cdot \min_{t \in [M^*_{\alpha,\beta} \cdot k, \beta k]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + \begin{cases} h_{k,t}(x_{\alpha,\beta}(k,t)) & \text{if } x_{\alpha,\beta}(k,t) \geq \frac{kt}{n} \\ 0 & \text{otherwise} \end{cases}\right).$$

Observe the range of t in the last expression is not restricted to integral values.

Claim 6.9. *For all $k \in \left[0, \frac{n}{\beta}\right] \cap \mathbb{N}$ and $(\alpha, c) \in \mathcal{L}$ it holds that*

$$\min_{t \in [M^*_{\alpha,\beta} \cdot k, \beta k]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + \begin{cases} h_{k,t}(x_{\alpha,\beta}(k,t)) & \text{if } x_{\alpha,\beta}(k,t) \geq \frac{kt}{n} \\ 0 & \text{otherwise} \end{cases}\right)$$

$$= \min_{t \in \left[M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n, \beta k\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + h_{k,t}(x_{\alpha,\beta}(k,t))\right).$$

Proof. First suppose $\alpha \geq \beta$. Then $M^*_{\alpha,\beta} \cdot k = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n$. Furthermore, by Lemma 6.4, it holds that $x_{\alpha,\beta}(k,t) \geq \frac{kt}{n}$ for all $t \in \left[M^*_{\alpha,\beta} \cdot k, \beta k\right]$, and the statement of the claim immediately follows.

We are left to handle the case $\alpha < \beta$. By Lemma 6.4 we have

$$\min_{t \in \left[M^*_{\alpha,\beta} \cdot k, M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + \begin{cases} h_{k,t}(x_{\alpha,\beta}(k,t)) & \text{if } x_{\alpha,\beta}(k,t) \geq \frac{kt}{n} \\ 0 & \text{otherwise} \end{cases}\right)$$

$$= \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + \begin{cases} h_{k,t}(x_{\alpha,\beta}(k,t)) & \text{if } t = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n \\ 0 & \text{otherwise} \end{cases}\right)$$

$$= \min_{t \in \left[M^*_{\alpha,\beta} \cdot k, M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)$$

$$= \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c\right)\Big|_{t = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n}$$

$$= \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + h_{k,t}(x_{\alpha,\beta}(k,t))\right)\Big|_{t = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n},$$

where the second and forth equalities follow from $x_{\alpha,\beta}(k,t) = \frac{kt}{n}$ for $t = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n$ and $h_{k,t}\left(\frac{kt}{n}\right) = 0$ (Claim 6.8). The statement of the claim follows from the last series of equalities. ⌋

By (39) and Claim 6.9 it holds that

$$\min_{t\in[M^*_{\alpha,\beta}\cdot k,\beta k]\cap\mathbb{N}}\frac{\exp\left(\frac{\beta k-t}{\alpha}\cdot\ln c\right)}{\texttt{hyper}(n,k,t,x_{\alpha,\beta}(k,t))} \tag{40}$$
$$\geq\; n^{-\mathcal{O}(1)}\cdot\min_{t\in\left[M_{\alpha,\beta}\left(\frac{k}{n}\right)\cdot n,\beta k\right]}\exp\left(\frac{\beta k-t}{\alpha}\cdot\ln c+h_{k,t}(x_{\alpha,\beta}(k,t))\right)$$

for all $k\in\left[0,\frac{n}{\beta}\right]\cap\mathbb{N}$ and $(\alpha,c)\in\mathcal{L}$. By the definition of $f_{\mathcal{L},\beta}$ (5) we have

$$f_{\mathcal{L},\beta}(n)=\max_{k\in\left[0,\frac{n}{\beta}\right]\cap\mathbb{N}}\;\min_{(\alpha,c)\in\mathcal{L}}\;\min_{t\in[M^*_{\alpha,\beta}\cdot k,\beta\cdot k]\cap\mathbb{N}}\frac{\exp\left(\frac{\beta k-t}{\alpha}\cdot\ln c\right)}{\texttt{hyper}\left(n,k,t,(1-\frac{\beta}{\alpha})\cdot k+\frac{t}{\alpha}\right)} \tag{41}$$
$$\geq\; n^{-\mathcal{O}(1)}\max_{k\in\left[0,\frac{n}{\beta}\right]\cap\mathbb{N}}\;\min_{(\alpha,c)\in\mathcal{L}}\;\min_{t\in\left[M_{\alpha,\beta}\left(\frac{k}{n}\right)\cdot n,\beta k\right]}\exp\left(\frac{\beta k-t}{\alpha}\cdot\ln c+h_{k,t}(x_{\alpha,\beta}(k,t))\right)$$

where the inequality is by (40). To complete the proof we need to change the range of k in (41) to a continuous range. The following claims are used to this end.

Claim 6.10. *Let $\beta,\alpha\geq 1$ and $\kappa\in\left[0,\frac{1}{\beta}\right]$. Then*

$$0\leq M'_{\alpha,\beta}(\kappa)\leq\begin{cases}\frac{\beta^2}{\beta-\alpha} & \text{if }\beta>\alpha\\ 0 & \text{if }\alpha=\beta\,,\\ \frac{\alpha-\beta}{\alpha-1} & \text{if }\beta<\alpha\end{cases}$$

where $M'_{\alpha,\beta}(\kappa)$ is the derivative of $M_{\alpha,\beta}(\kappa)$.

Proof. Consider the following cases.

- If $\beta>\alpha$ it holds that

$$M'_{\alpha,\beta}(\kappa)=\frac{(1-\alpha\kappa)\cdot(\beta-\alpha)+\alpha(\beta-\alpha)\cdot\kappa}{(1-\alpha\kappa)^2}=\frac{\beta-\alpha}{(1-\alpha\kappa)^2}.$$

  Observe the value is well define since $\alpha\cdot\kappa\leq\alpha\cdot\frac{1}{\beta}<1$. Thus, $M'_{\alpha,\beta}(\kappa)=\frac{\beta-\alpha}{(1-\alpha\kappa)^2}\geq 0$. Similarly, since $\kappa\leq\frac{1}{\beta}$,

$$M'_{\alpha,\beta}(\kappa)=\frac{\beta-\alpha}{(1-\alpha\kappa)^2}\leq\frac{\beta-\alpha}{\left(1-\alpha\cdot\frac{1}{\beta}\right)^2}=\frac{\beta^2}{\beta-\alpha}.$$

- If $\alpha=\beta$ it holds that $M'_{\alpha,\beta}(\kappa)=0$.

- If $\beta<\alpha$ it holds that $M'_{\alpha,\beta}(\kappa)=\frac{\alpha-\beta}{\alpha-1}>0$. ⌟

Claim 6.11. *Let $k\in\left[0,\frac{n}{\beta}\right]$ and $k'=\lfloor k\rfloor$. Then*

$$\min_{(\alpha,c)\in\mathcal{L}}\;\min_{t\in\left[M_{\alpha,\beta}\left(\frac{k}{n}\right)\cdot n,\beta k\right]}\exp\left(\frac{\beta k-t}{\alpha}\cdot\ln c+h_{k,t}(x_{\alpha,\beta}(k,t))\right)$$
$$\leq\; n^{\mathcal{O}(1)}\cdot\min_{(\alpha,c)\in\mathcal{L}}\;\min_{t\in\left[M_{\alpha,\beta}\left(\frac{k'}{n}\right)\cdot n,\beta k'\right]}\exp\left(\frac{\beta k'-t}{\alpha}\cdot\ln c+h_{k',t}(x_{\alpha,\beta}(k',t))\right).$$

33

Proof. Pick $(\alpha', c') \in \mathcal{L}$ and $t' \in \left[M_{\alpha',\beta}\left(\frac{k'}{n}\right) \cdot n, \beta k' \right]$ such that

$$\exp\left(\frac{\beta k' - t'}{\alpha'} \ln c' + h_{k',t'}(x_{\alpha',\beta}(k',t')) \right)$$

$$= \min_{(\alpha,c)\in\mathcal{L}} \min_{t\in\left[M_{\alpha,\beta}\left(\frac{k'}{n}\right)\cdot n, \beta k'\right]} \exp\left(\frac{\beta k' - t}{\alpha} \ln c + h_{k',t}(x_{\alpha,\beta}(k',t)) \right).$$

We set $t'' := \max\{t', M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n\}$. Since $t' \le \beta k' \le \beta k$ and $M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n \le \beta k$ (Lemma 6.5) it follows that $t'' \le \beta k$. In order to bound $|t' - t''|$ we consider the following cases.

- If $t' = t''$ we have $|t' - t''| = 0$.

- If $t'' = M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n$ it holds that $M_{\alpha,\beta}\left(\frac{k'}{n}\right) \cdot n \le t' \le M_{\alpha,\beta}\left(\frac{k}{n}\right) \cdot n = t''$. Then

$$\left| t'' - t' \right| \le n \cdot \left| M_{\alpha,\beta}\left(\frac{k'}{n}\right) - M_{\alpha,\beta}\left(\frac{k}{n}\right) \right| \le n \cdot \mathcal{O}(1) \cdot \frac{k' - k}{n} = \mathcal{O}(1),$$

  where the second inequality follows from Claim 6.10.

So overall $|t'' - t'| = \mathcal{O}(1)$. Hence,

$$\min_{(\alpha,c)\in\mathcal{L}} \min_{t\in\left[M_{\alpha,\beta}\left(\frac{k}{n}\right)\cdot n, \beta k\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + h_{k,t}(x_{\alpha,\beta}(k,t)) \right)$$

$$\le \exp\left(\frac{\beta k - t''}{\alpha'} \cdot \ln c' + h_{k,t''}(x_{\alpha',\beta}(k,t'')) \right)$$

$$= \exp\left(\frac{\beta k - t''}{\alpha'} \cdot \ln c' - t'' \cdot \mathcal{H}\left(\frac{x_{\alpha',\beta}(k,t'')}{t''} \right) - (n - t'') \cdot \mathcal{H}\left(\frac{k - x_{\alpha',\beta}(k,t'')}{n - t''} \right) + n \cdot \mathcal{H}\left(\frac{k}{n} \right) \right)$$

$$= \exp\left(\frac{\beta k - t''}{\alpha'} \cdot \ln c' - t'' \cdot \mathcal{H}\left(\frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t''}{\alpha}}{t''} \right) - (n - t'') \cdot \mathcal{H}\left(\frac{\frac{\beta}{\alpha} \cdot k - \frac{t''}{\alpha}}{n - t''} \right) + n \cdot \mathcal{H}\left(\frac{k}{n} \right) \right)$$

$$\le n^{\mathcal{O}(1)} \cdot \exp\left(\frac{\beta k' - t'}{\alpha'} \cdot \ln c' - t' \cdot \mathcal{H}\left(\frac{\left(1 - \frac{\beta}{\alpha}\right) k' + \frac{t'}{\alpha}}{t'} \right) - (n - t') \cdot \mathcal{H}\left(\frac{\frac{\beta}{\alpha} k' - \frac{t'}{\alpha}}{n - t'} \right) + n \cdot \mathcal{H}\left(\frac{k'}{n} \right) \right)$$

$$= n^{\mathcal{O}(1)} \cdot \exp\left(\frac{\beta k' - t'}{\alpha'} \cdot \ln c' + h_{k',t'}(x_{\alpha',\beta}(k',t')) \right)$$

$$= n^{\mathcal{O}(1)} \cdot \min_{(\alpha,c)\in\mathcal{L}} \min_{t\in\left[M_{\alpha,\beta}\left(\frac{k'}{n}\right)\cdot n, \beta k'\right]} \exp\left(\frac{\beta k' - t}{\alpha} \ln c + h_{k',t}(x_{\alpha,\beta}(k',t)) \right).$$

The second inequality follows from Lemma 6.1. ⌟

By (41) and Claim 6.11 it holds that

$$f_{\mathcal{L},\beta}(n) \ge n^{-\mathcal{O}(1)} \max_{k\in\left[0,\frac{n}{\beta}\right]\cap\mathbb{N}} \min_{(\alpha,c)\in\mathcal{L}} \min_{t\in\left[M_{\alpha,\beta}\left(\frac{k}{n}\right)\cdot k, \beta k\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + h_{k,t}(x_{\alpha,\beta}(k,t)) \right)$$

$$\ge n^{-\mathcal{O}(1)} \max_{k\in\left[0,\frac{n}{\beta}\right]} \min_{(\alpha,c)\in\mathcal{L}} \min_{t\in\left[M_{\alpha,\beta}\left(\frac{k}{n}\right)\cdot n, \beta k\right]} \exp\left(\frac{\beta k - t}{\alpha} \cdot \ln c + h_{k,t}(x_{\alpha,\beta}(k,t)) \right)$$

$$= n^{-\mathcal{O}(1)} \max_{\kappa\in\left[0,\frac{1}{\beta}\right]} \min_{(\alpha,c)\in\mathcal{L}} \min_{\tau\in\left[M_{\alpha,\beta}(\kappa), \beta\kappa\right]} \exp\left(\frac{\beta\kappa - \tau}{\alpha} \cdot \ln c + \frac{1}{n} \cdot h_{\kappa\cdot n,\tau\cdot n}(x_{\alpha,\beta}(\kappa \cdot n, \tau \cdot n)) \right)^n$$

$$= n^{-\mathcal{O}(1)} \max_{\kappa\in\left[0,\frac{1}{\beta}\right]} \min_{(\alpha,c)\in\mathcal{L}} \min_{\tau\in\left[M_{\alpha,\beta}(\kappa), \beta\kappa\right]} \exp\left(g_{\alpha,\beta,c}(\kappa, \tau) \right)^n$$

$$= n^{-\mathcal{O}(1)} \left(\mathtt{amls}(\mathcal{L}, \beta) \right)^n.$$

The first equality simply replaces k and t with $\kappa \cdot n$ and $\tau \cdot n$. The second equality follows from

$$
\frac{\beta\kappa - \tau}{\alpha} \cdot \ln c + \frac{1}{n} \cdot h_{\kappa \cdot n, \tau \cdot n}(x_{\alpha,\beta}(\kappa \cdot n, \tau \cdot n))
$$

$$
= \frac{\beta\kappa - \tau}{\alpha} \cdot \ln c - \tau \cdot \mathcal{H}\left(\frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa + \frac{\tau}{\alpha}}{\tau}\right) - (1 - \tau) \cdot \mathcal{H}\left(\frac{\frac{\beta}{\alpha} \cdot \kappa - \frac{\tau}{\alpha}}{1 - \tau}\right) + \mathcal{H}(\kappa)
$$

$$
= g_{\alpha,\beta,c}(\kappa, \tau).
$$

□

Proof of Lemma 2.11. The lemma follows immediately from Lemmas 6.6 and 6.7. □

7 Evaluating the Running Time: Convexity and Concavity

In this section we prove Lemmas 2.13 and 2.15 which provide the mathematical properties required for the evaluation of `amls`, as well as Lemma 2.19. In Section 7.1 we state basic properties of the functions δ, γ and g which we need later on. In Section 7.2 we show that $g_{\alpha,\beta,c}(\kappa, \tau)$ is a convex function of the variable τ in the interval $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$, for all $\kappa \in \left(0, \frac{1}{\beta}\right)$. This fact is used to show the value $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ that minimizes $g_{\alpha,\beta,c}(\kappa, \tau)$ actually belongs to $[M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. In Section 7.3 we prove that $g^*(\kappa)$ is a concave function of κ in the interval $\kappa \in \left[0, \frac{1}{\beta}\right]$. The proof uses properties of the Hessian which are shown in Section 7.4, and relies on technical computations from Section 7.5 and Section 7.6. Section 7.3 also contains the proof of Lemma 2.19 which follows from the technical lemmas proved in the same section.

Recall the definitions of the functions used in the definition of `amls`:

$$
\delta_{\alpha,\beta}(\kappa, \tau) = \begin{cases} \frac{\frac{\beta}{\alpha}\kappa - \frac{\tau}{\alpha}}{1 - \tau} = \frac{\frac{\beta}{\alpha}\kappa - \frac{1}{\alpha}}{1 - \tau} + \frac{1}{\alpha} & \text{if } \tau \neq 1 \\ \frac{1}{\alpha} & \text{if } \tau = 1 \end{cases}
$$

$$
\gamma_{\alpha,\beta}(\kappa, \tau) = \begin{cases} \left(1 - \frac{\beta}{\alpha}\right)\frac{\kappa}{\tau} + \frac{1}{\alpha} & \text{if } \tau \neq 0 \\ \frac{1}{\alpha} & \text{if } \tau = 0 \end{cases}
$$

$$
g_{\alpha,\beta,c}(\kappa, \tau) = \frac{\beta\kappa - \tau}{\alpha} \ln c - \tau \cdot \mathcal{H}(\gamma_{\alpha,\beta}(\kappa, \tau)) - (1 - \tau) \cdot \mathcal{H}(\delta_{\alpha,\beta}(\kappa, \tau)) + \mathcal{H}(\kappa)
$$

$$
M_{\alpha,\beta}(\kappa) = \begin{cases} \frac{\beta - \alpha}{1 - \alpha \cdot \kappa} \cdot \kappa & \text{if } \alpha < \beta \\ 0 & \text{if } \alpha = \beta \\ \frac{\alpha - \beta}{\alpha - 1} \cdot \kappa & \text{if } \alpha > \beta \end{cases}
$$

With a slight abuse of notation, we sometimes omit the subscript (α, β, c) from $g_{\alpha,\beta,c}$ or (α, β) from $\delta_{\alpha,\beta}$ and $\gamma_{\alpha,\beta}$, whenever it is clear from the context.

7.1 Basic Properties

We start by discussing basic properties of the functions γ, δ and g. We commonly rely on monotonicity properties of δ and γ, as well as their possible range of values.

Lemma 7.1. *For all $\alpha \geq 1$, $\beta > 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$, it holds that $\delta_{\alpha,\beta}(\kappa, \tau)$ is strictly decreasing with τ in the range $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$. Furthermore, $0 \leq \delta_{\alpha,\beta}(\kappa, \tau) < \frac{1}{\alpha}$ for all $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$.*

Proof. Since $\kappa < \frac{1}{\beta}$ and $\frac{\beta}{\alpha} \cdot \kappa - \frac{1}{\alpha} < 0$, the term $\frac{\frac{\beta}{\alpha} \cdot \kappa - \frac{1}{\alpha}}{1 - \tau}$ is a strictly decreasing function of τ. Therefore $\delta(\kappa, \tau)$ is strictly decreasing with τ because $\delta(\kappa, \tau) = \frac{\frac{\beta}{\alpha} \cdot \kappa - \frac{1}{\alpha}}{1 - \tau} + \frac{1}{\alpha}$. Therefore, for all $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$, we have $0 = \delta(\kappa, \beta \cdot \kappa) \leq \delta(\kappa, \tau)$ and

$$
\delta(\kappa, \tau) \leq \delta(\kappa, M_{\alpha,\beta}(\kappa)) = \frac{\frac{\beta}{\alpha} \cdot \kappa - \frac{1}{\alpha}}{1 - M_{\alpha,\beta}(\kappa)} + \frac{1}{\alpha} < \frac{1}{\alpha},
$$

where the last step holds because $M_{\alpha,\beta}(\kappa) \leq \beta \cdot \kappa < 1$ and $\frac{\beta}{\alpha} \cdot \kappa - \frac{1}{\alpha} < 0$. $\hspace{1cm}\square$

While the function δ is decreasing regardless of the values of α and β, the direction of monotonicity of γ does depend on the values of α and β.

Lemma 7.2. *For all $\alpha \geq 1$, $\beta > 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$, the function $\gamma_{\alpha,\beta}(\kappa, \tau)$ satisfies the following properties depending on the values of α and β:*

- *If $\alpha > \beta$, the function $\gamma_{\alpha,\beta}(\kappa, \tau)$ is strictly decreasing with $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$, and $\frac{1}{\beta} \leq \gamma_{\alpha,\beta}(\kappa, \tau) \leq 1$.*

- *If $\alpha = \beta$, it holds that $\gamma_{\alpha,\beta}(\kappa, \tau) = \frac{1}{\alpha}$ for all $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$.*

- *If $\alpha < \beta$, the function $\gamma_{\alpha,\beta}(\kappa, \tau)$ is strictly increasing with $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$, and $\kappa \leq \gamma_{\alpha,\beta}(\kappa, \tau) \leq \frac{1}{\beta}$.*

Proof. Let us fix a $\kappa \in \left(0, \frac{1}{\beta}\right)$ and consider the different cases where $\alpha > \beta$, $\alpha = \beta$ and $\alpha < \beta$.

- If $\alpha > \beta$, the term $\left(1 - \frac{\beta}{\alpha}\right)$ is strictly positive, therefore $\left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{\kappa}{\tau}$ is a strictly decreasing function of τ. It follows that $\gamma(\kappa, \tau) = \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{\kappa}{\tau} + \frac{1}{\alpha}$ is also a strictly decreasing function of τ. Moreover, for all $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$, we have

$$\gamma(\kappa, \tau) \geq \gamma(\kappa, \beta \cdot \kappa) = \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{\kappa}{\beta \cdot \kappa} + \frac{1}{\alpha} = \frac{1}{\beta} \qquad \text{and}$$

$$\gamma(\kappa, \tau) \leq \gamma\left(\kappa, M_{\alpha,\beta}(\kappa)\right) = \gamma\left(\kappa, \frac{\alpha - \beta}{\alpha - 1} \cdot \kappa\right) = \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{\alpha - 1}{\alpha - \beta} + \frac{1}{\alpha} = 1.$$

- If $\beta = \alpha$, it is easy to see that $\gamma(\kappa, \tau) = \frac{1}{\alpha}$ for all $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ because $\left(1 - \frac{\beta}{\alpha}\right) = 0$.

- If $\alpha < \beta$, the term $\left(1 - \frac{\beta}{\alpha}\right)$ is strictly negative and $\left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{\kappa}{\tau}$ is a strictly increasing function of τ. Therefore $\gamma(\kappa, \tau) = \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{\kappa}{\tau} + \frac{1}{\alpha}$ is also a strictly increasing function of τ. Moreover, for all $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ it holds that

$$\gamma(\kappa, \tau) \leq \gamma(\kappa, \beta \cdot \kappa) = \frac{1}{\beta} \qquad \text{and}$$

$$\gamma(\kappa, \tau) \geq \gamma\left(\kappa, M_{\alpha,\beta}(\kappa)\right) = \gamma\left(\kappa, \frac{(\beta - \alpha)}{1 - \alpha \cdot \kappa} \cdot \kappa\right) = \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{1 - \alpha \cdot \kappa}{\beta - \alpha} + \frac{1}{\alpha}$$
$$= \frac{\alpha \kappa - 1 + 1}{\alpha}$$
$$= \kappa.$$

$\hspace{1cm}\square$

The next lemma is also used a few times.

Lemma 7.3. *Let $\beta > \alpha > 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$. Then*

$$\delta_{\alpha,\beta}\left(\kappa, M_{\alpha,\beta}(\kappa)\right) = \gamma_{\alpha,\beta}(\kappa, M_{\alpha,\beta}(\kappa)) = \kappa.$$

Proof. By simple calculation we have

$$\delta(\kappa, M_{\alpha,\beta}(\kappa)) = \frac{1}{\alpha} \cdot \left(\frac{\beta \cdot \kappa - 1}{1 - \frac{(\beta - \alpha) \cdot \kappa}{1 - \alpha \cdot \kappa}} + 1\right)$$

$$= \frac{1}{\alpha} \cdot \left(\frac{\beta \cdot \kappa - 1}{\frac{1 - \alpha \cdot \kappa - \beta \cdot \kappa + \alpha \cdot \kappa}{1 - \alpha \cdot \kappa}} + 1 \right)$$

$$= \frac{1}{\alpha} \cdot \left((1 - \alpha \cdot \kappa) \cdot \frac{\beta \cdot \kappa - 1}{1 - \beta \cdot \kappa} + 1 \right)$$

$$= \frac{1}{\alpha} \cdot (\alpha \cdot \kappa)$$

$$= \kappa.$$

Similarly,

$$\gamma(\kappa, M_{\alpha,\beta}(\kappa)) = \left(1 - \frac{\beta}{\alpha} \right) \cdot \frac{\kappa}{\frac{(\beta - \alpha) \cdot \kappa}{1 - \alpha \cdot \kappa}} + \frac{1}{\alpha}$$

$$= \frac{\alpha - \beta}{\alpha} \cdot \frac{\kappa \cdot (1 - \alpha \cdot \kappa)}{(\beta - \alpha) \cdot \kappa} + \frac{1}{\alpha}$$

$$= \frac{\alpha \cdot \kappa - 1}{\alpha} + \frac{1}{\alpha}$$

$$= \kappa.$$

$\square$

Finally, we show g is non-negative.

Lemma 7.4. *For all $\alpha, \beta, c \geq 1$, $\kappa \in \left[0, \frac{1}{\beta} \right]$ and $\tau \in [M_{\alpha,\beta}(\kappa), \beta\kappa]$ it holds that $g_{\alpha,\beta,c}(\kappa, \tau) \geq 0$.*

Proof. It holds that

$$
\begin{aligned}
g_{\alpha,\beta,c}(\kappa, \tau) &= \frac{\beta\kappa - \tau}{\alpha} \cdot \ln c - \tau \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\kappa, \tau) \right) - (1 - \tau) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\kappa, \tau) \right) + \mathcal{H}(\kappa) \\
&\geq 0 - \tau \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\kappa, \tau) \right) - (1 - \tau) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\kappa, \tau) \right) + \mathcal{H}(\kappa) \\
&\geq -\mathcal{H}\left(\tau \cdot \gamma_{\alpha,\beta}(\kappa, \tau) + (1 - \tau) \cdot \delta_{\alpha,\beta}(\kappa, \tau) \right) + \mathcal{H}(\kappa) \\
&= -\mathcal{H}(\kappa) + \mathcal{H}(\kappa) = 0,
\end{aligned}
$$

where the first inequality holds since $\tau \leq \beta \cdot \kappa$ and $c \geq 1$. The second inequality holds as $\mathcal{H}$ is concave, and the second equality follows from $\tau \cdot \gamma(\kappa, \tau) + (1 - \tau) \cdot \delta(\kappa, \tau) = \kappa$. $\square$

7.2 Convexity

In this section we fix some value for κ and analyze the function $g_{\alpha,\beta,c}(\kappa, \tau)$ as a function of τ. For all $\kappa \in \left[0, \frac{1}{\beta} \right]$, we define $g_{\alpha,\beta,c}^{[\kappa]} \colon (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa) \to \mathbb{R}$ via

$$g_{\alpha,\beta,c}^{[\kappa]}(\tau) := g_{\alpha,\beta,c}(\kappa, \tau).$$

Recall that $\mathcal{D}\left(a \| b \right) = a \ln \frac{a}{b} + (1 - a) \ln \frac{1-a}{1-b}$ is the Kullback-Leibler divergence between two Bernoulli distributions with parameters a and b.

Let $(\partial_\tau g_{\alpha,\beta,c})(\kappa_0, \tau_0)$ and $(\partial_{\tau,\tau}^2 g_{\alpha,\beta,c})(\kappa_0, \tau_0)$ denote the first and second order partial derivatives of the function $g_{\alpha,\beta,c}(\kappa, \tau)$ with respect to the variable τ, evaluated at (κ_0, τ_0). Lemmas 7.5 and 7.6 provide formulas for $(\partial_\tau g_{\alpha,\beta,c})(\kappa_0, \tau_0)$ and $(\partial_{\tau,\tau}^2 g_{\alpha,\beta,c})(\kappa_0, \tau_0)$. The lemmas follow from a simple calculation and we defer the proofs to Section 7.5.

Lemma 7.5. *For all $\alpha, c \geq 1$ and $\beta > 1$ it holds that*

$$(\partial_\tau g_{\alpha,\beta,c})(\kappa, \tau) = -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma(\kappa, \tau) \right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta(\kappa, \tau) \right).$$

We use the following functions to simplify the formula for $(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa,\tau)$:

$$\Gamma_{\alpha,\beta}(\kappa,\tau) = \frac{1}{\tau} \cdot \frac{1}{\gamma_{\alpha,\beta}(\kappa,\tau) \cdot (1 - \gamma_{\alpha,\beta}(\kappa,\tau))} \tag{42}$$

$$\Delta_{\alpha,\beta}(\kappa,\tau) = \frac{1}{(1-\tau)} \cdot \frac{1}{\delta_{\alpha,\beta}(\kappa,\tau) \cdot (1 - \delta_{\alpha,\beta}(\kappa,\tau))} \tag{43}$$

Lemma 7.6. *For all $\alpha, c \geq 1$ and $\beta > 1$, the second order partial derivative of $g(\kappa,\tau)$ by τ, i.e., $(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa,\tau)$ is given by*

$$(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa,\tau) = \left(\gamma(\kappa,\tau) - \frac{1}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa,\tau) + \left(\delta(\kappa,\tau) - \frac{1}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa,\tau).$$

We sometimes omit the subscript α, β, c from $g^{[\kappa]}_{\alpha,\beta,c}$, $(\partial_\tau g_{\alpha,\beta,c})$ and $(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})$ unless it causes confusion.

Lemma 7.7. *Let $\alpha, c \geq 1$, $\beta > 1$ and $0 < \kappa < \frac{1}{\beta}$. The function $g^{[\kappa]}_{\alpha,\beta,c}(\tau)$ is strictly convex in the open interval $(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. In particular, the second order partial derivative of $g_{\alpha,\beta,c}(\kappa,\tau)$ with respect to τ is strictly positive, i.e., $(\partial^2_{\tau,\tau} g)(\kappa,\tau) > 0$ for all $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$.*

Proof. Let $\alpha, c \geq 1$, $\beta > 1$, by Lemma 7.6 and (43) and (42) we have

$$(\partial^2_{\tau,\tau} g)(\kappa,\tau) = \frac{1}{\tau} \cdot \frac{\left(\gamma(\kappa,\tau) - \frac{1}{\alpha}\right)^2}{\gamma(\kappa,\tau) \cdot \left(1 - \gamma(\kappa,\tau)\right)} + \frac{1}{1-\tau} \cdot \frac{\left(\delta(\kappa,\tau) - \frac{1}{\alpha}\right)^2}{\delta(\kappa,\tau) \cdot \left(1 - \delta(\kappa,\tau)\right)}.$$

By Lemmas 7.1 and 7.2, we have that $0 < \delta(\kappa,\tau) < \frac{1}{\alpha} < 1$ and $0 < \gamma(\kappa,\tau) < 1$, since $\beta > 1$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa) \subseteq (0,1)$. Therefore, it follows that $(\partial^2_{\tau,\tau} g)(\kappa,\tau) > 0$. Thus $g^{[\kappa]}(\tau)$ is a strictly convex function. $\qquad\square$

Since $g^{[\kappa]}_{\alpha,\beta,c}(\tau)$ is a continuous function of the variable τ, it attains its minimum over the closed interval $[M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$. We show that up to some corner cases, the optimal value of τ lies within the open interval $(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. We use the following definition to easily exclude the corner cases.

Definition 7.8. *We say $\alpha, \beta, c \geq 1$ are simple if $\beta > 1$ and none of the following conditions hold:*

- $\alpha = \beta$, *or*

- $c = 1$ *and* $\alpha < \beta$.

Lemma 7.9. *For all $\alpha \geq 1$, $c \geq 1$, $\beta > 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$, there is a unique value of $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$, denoted by $\tau^*_{\alpha,\beta,c}(\kappa)$, that minimizes $g_{\alpha,\beta,c}(\kappa,\tau)$. Furthermore, it holds that $\tau^*_{\alpha,\beta,c}(\kappa) < \beta\kappa$, and if α, β, c are simple then $M_{\alpha,\beta}(\kappa) < \tau^*_{\alpha,\beta,c}(\kappa) < \beta\kappa$ and $(\partial_\tau g)\left(\kappa, \tau^*_{\alpha,\beta,c}(\kappa)\right) = 0$.*

Proof. Since the function $g^{[\kappa]}(\tau)$ is strictly convex by Lemma 7.7, the value

$$\tau^*_{\alpha,\beta,c}(\kappa) = \underset{\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]}{\operatorname{argmin}} g_{\alpha,\beta,c}(\kappa,\tau)$$

is uniquely defined. Also note that $(\partial_\tau g)(\kappa,\tau)$ is an increasing function of τ by Lemma 7.7.

By Lemma 7.5 we have

$$\lim_{\tau \to \beta \cdot \kappa} (\partial_\tau g)(\kappa,\tau) = \lim_{\tau \to \beta \cdot \kappa} \left(-\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma(\kappa,\tau)\right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta(\kappa,\tau)\right) \right)$$

$$= -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha} \middle\| \frac{1}{\beta}\right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \lim_{\tau \to \beta \cdot \kappa} \delta(\kappa,\tau)\right) = \infty, \tag{44}$$

which follows from the fact that $\mathcal{D}\left(\frac{1}{\alpha}\big\|x\right)$ is a continuous function of x and $\lim_{\tau \to \beta \cdot \kappa} \delta(\kappa, \tau) = 0$. Thus, there is $\varepsilon > 0$ such that $(\partial_\tau g)(\kappa, \tau) > 0$ for all $\tau \in (\beta\kappa - \varepsilon, \beta\kappa)$, and hence $g(\kappa, \tau)$ is strictly increasing in $(\beta\kappa - \varepsilon, \beta\kappa)$, and by continuity in $(\beta\kappa - \varepsilon, \beta\kappa]$. This implies that $\tau^*_{\alpha,\beta,c}(\kappa) < \beta\kappa$ by its definition.

This above completes the proof for general values of α, β, c, and thus we can assume α, β, c are simple from this point onward. Using Lemma 7.5 once more we get

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} (\partial_\tau g)(\kappa, \tau) = -\frac{\ln(c)}{\alpha} - \lim_{\tau \to M_{\alpha,\beta}(\kappa)} \mathcal{D}\left(\frac{1}{\alpha}\big\|\gamma(\kappa, \tau)\right) + \lim_{\tau \to M_{\alpha,\beta}(\kappa)} \mathcal{D}\left(\frac{1}{\alpha}\big\|\delta(\kappa, \tau)\right). \tag{45}$$

Consider the following cases.

- If $\alpha > \beta > 1$ and $c \geq 1$, we have

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \gamma(\kappa, \tau) = \gamma(\kappa, M_{\alpha,\beta}(\kappa)) = \gamma\left(\kappa, \frac{\alpha - \beta}{\alpha - 1} \cdot \kappa\right) = 1,$$

  therefore $\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \mathcal{D}\left(\frac{1}{\alpha}\big\|\gamma(\kappa, \tau)\right) = \infty$. Similarly, by Lemma 7.1, we have

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \delta(\kappa, \tau) = \delta(\kappa, M_{\alpha,\beta}(\kappa)) < \frac{1}{\alpha} < 1,$$

  and

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \delta(\kappa, \tau) = \delta(\kappa, M_{\alpha,\beta}(\kappa)) > 0, \tag{46}$$

  since $M_{\alpha,\beta}(\kappa) > 0$. Therefore it holds that $\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \mathcal{D}\left(\frac{1}{\alpha}\big\|\delta(\kappa, \tau)\right) < \infty$ and by (45) we get

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} (\partial_\tau g)(\kappa, \tau) = \lim_{\tau \to \beta \cdot \kappa} \left(-\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha}\big\|\gamma(\kappa, \tau)\right) + \mathcal{D}\left(\frac{1}{\alpha}\big\|\delta(\kappa, \tau)\right)\right) = -\infty.$$

- If $\beta > \alpha \geq 1$, $c > 1$, by Lemma 7.3 we have

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \delta(\kappa, \tau) = \delta(\kappa, M_{\alpha,\beta}(\kappa)) = \kappa$$

  and

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} \gamma(\kappa, \tau) = \gamma(\kappa, M_{\alpha,\beta}(\kappa)) = \kappa.$$

  Thus, by (45), we have

$$\lim_{\tau \to M_{\alpha,\beta}(\kappa)} (\partial_\tau g)(\kappa, \tau) = -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha}\big\|\kappa\right) + \mathcal{D}\left(\frac{1}{\alpha}\big\|\kappa\right) = -\frac{\ln(c)}{\alpha} < 0,$$

  since $c > 1$.

So in both cases $\lim_{\tau \to M_{\alpha,\beta}(\kappa)} (\partial_\tau g)(\kappa, \tau) < 0$. Thus, by (44) there is $\tilde{\tau} \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$ such that $(\partial_\tau g)(\kappa, \tilde{\tau}) = 0$, and since $g_{\alpha,\beta,c}(\kappa, \tau)$ is convex as a function of τ this implies $\tau^*_{\alpha,\beta,c}(\kappa) = \tilde{\tau}$. Hence $(\partial_\tau g)(\kappa, \tau^*_{\alpha,\beta,c}(\kappa)) = 0$ and $\tau^*_{\alpha,\beta,c}(\kappa) \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. $\qquad \square$

Lemma 2.13. *Let $\alpha, c \geq 1$, $\beta > 1$ and $0 < \kappa < \frac{1}{\beta}$. The function $h(\tau) = g_{\alpha,\beta,c}(\kappa, \tau)$ is convex in the domain $[M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ and $\min_{\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]} g_{\alpha,\beta,c}(\kappa, \tau) < h(\beta\kappa) = g_{\alpha,\beta,c}(\kappa, \beta\kappa)$.*

Proof. The first part of the claim follows from the fact that $g^{[\kappa]}(\tau)$ is a continuous function on the closed interval $[M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ and convex on the open interval $(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$ by Lemma 7.7. The second part of the claim simply follows from Lemmas 7.7 and 7.9. $\qquad \square$

7.3 Concavity

In this section we prove Lemma 2.15, that is, we show $g_{\alpha,\beta,c}^*(\kappa)$ is concave. The proof relies on properties of the Hessian of $g_{\alpha,\beta,c}$ when α, β, c are simple (see Definition 7.8). The excluded corner cases, in which α, β, c are not simple, are handled separately.

As in previous sections, we use g and g^* instead of $g_{\alpha,\beta,c}$ and $g_{\alpha,\beta,c}^*$ when the values of α, β and c are known by context. Recall that the Hessian matrix of g at (κ, τ), denoted $H_g(\kappa, \tau)$, is defined by

$$H_g(\kappa, \tau) = \begin{pmatrix} \frac{\partial^2 g(\kappa,\tau)}{\partial \kappa^2} & \frac{\partial^2 g(\kappa,\tau)}{\partial \kappa \partial \tau} \\ \frac{\partial^2 g(\kappa,\tau)}{\partial \kappa \partial \tau} & \frac{\partial^2 g(\kappa,\tau)}{\partial \tau^2} \end{pmatrix}.$$

For every (κ, τ) in the domain of g, $|H_g(\kappa, \tau)|$ denotes the determinant of the Hessian of the function g evaluated at (κ, τ). Specifically, we have

$$|H_g(\kappa, \tau)| = \frac{\partial^2 g(\kappa,\tau)}{\partial \kappa^2} \cdot \frac{\partial^2 g(\kappa,\tau)}{\partial \tau^2} - \left(\frac{\partial^2 g(\kappa,\tau)}{\partial \kappa \partial \tau}\right)^2.$$

Our proof is motivated by the second partial derivative test for multivariate functions, which uses the Hessian to classify critical points to maximum, minimum and saddle points. Technically, we use the Hessian directly and do explicitly rely on the second derivative test.

Recall that $\tau_{\alpha,\beta,c}^*(\kappa)$ is the unique value of the $\tau \in [M_{\alpha,\beta}(\kappa), \beta \cdot \kappa]$ that minimizes $g_{\alpha,\beta,c}^{[\kappa]}(\tau)$.

Lemma 7.10. *For all simple $\alpha, \beta, c \geq 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$, the determinant of the Hessian of g at $(\kappa, \tau^*(\kappa))$ is negative, i.e., $|H_g(\kappa, \tau^*(\kappa))| < 0$.*

The proof of Lemma 7.10 is given in Section 7.4. We also use the next theorem from [22] (see also [44]) to show that $\tau_{\alpha,\beta,c}^*$ is continuously differentiable and to calculate its derivative.

Theorem 7.11 (Implicit Function Theorem for $\mathbb{R}^2$, [22, Theorem 4]). *Let $\mathcal{G}(x, y)$ be a real-valued continuously differentiable function defined in a neighbourhood of $(x_0, y_0) \in \mathbb{R}^2$. Suppose that $\mathcal{G}(x, y)$ satisfies the two conditions*

$$\mathcal{G}(x_0, y_0) = 0,$$

$$\left.\frac{\partial \mathcal{G}(x, y)}{\partial y}\right|_{(x,y)=(x_0,y_0)} > 0.$$

Then there exist open intervals $U \subseteq \mathbb{R}$ and $V \subseteq \mathbb{R}$, with $x_0 \in U, y_0 \in V$, and a function $G\colon U \to V$ satisfying

$$\mathcal{G}(x, G(x)) = 0, \qquad \text{for all } x \in U.$$

Furthermore, this function G is continuously differentiable with

$$G'(x_0) = \left.\frac{\partial G(x)}{\partial x}\right|_{x=x_0} = -\frac{\left.\frac{\partial \mathcal{G}(x,y)}{\partial x}\right|_{(x,y)=(x_0,y_0)}}{\left.\frac{\partial \mathcal{G}(x,y)}{\partial y}\right|_{(x,y)=(x_0,y_0)}}.$$

We use Theorem 7.11 in the proof of the following lemma.

Lemma 7.12. *For every simple $\alpha, \beta, c \geq 1$, the function $\tau_{\alpha,\beta,c}^*(\kappa)$ is continuously differentiable on $\left(0, \frac{1}{\beta}\right)$. Moreover, for all $\kappa_0 \in \left(0, \frac{1}{\beta}\right)$ it holds that*

$$\left.\frac{\partial \tau_{\alpha,\beta,c}^*(\kappa)}{\partial \kappa}\right|_{\kappa=\kappa_0} = -\frac{(\partial_{\kappa,\tau}^2 g_{\alpha,\beta,c})(\kappa_0, \tau_{\alpha,\beta,c}^*(\kappa_0))}{(\partial_{\tau,\tau}^2 g_{\alpha,\beta,c})(\kappa_0, \tau_{\alpha,\beta,c}^*(\kappa_0))}.$$

Proof. Let $\kappa_0 \in \left(0, \frac{1}{\beta}\right)$ and $\tau_0 = \tau^*(\kappa_0)$. Consider a function $\mathcal{G}$ defined on a neighborhood E of (κ_0, τ_0) by

$$\mathcal{G}(\kappa, \tau) = (\partial_\tau g_{\alpha,\beta,c})(\kappa, \tau),$$

for every $(\kappa, \tau) \in E$. Lemma 7.6 implies that the partial derivative $\frac{\partial g(\kappa, \tau)}{\partial \tau}$ of $g(\kappa, \tau)$ is continuously differentiable. By Lemma 7.9 we have that $\mathcal{G}(\kappa_0, \tau_0) = 0$. Furthermore, by Lemma 7.7 we also have

$$\frac{\partial \mathcal{G}}{\partial \tau}(\kappa_0, \tau_0) = \left.\frac{\partial^2 g(\kappa, \tau)}{\partial \tau^2}\right|_{(\kappa,\tau)=(\kappa_0,\tau_0)} > 0,$$

therefore Theorem 7.11 implies that exists open intervals U, V with $\kappa_0 \in U, \tau_0 \in V$ and a continuously differentiable function $G \colon U \to V$ such that

$$\mathcal{G}\big(\kappa, G(\overline{\kappa})\big) = \left.\frac{\partial g(\kappa, \tau)}{\partial \tau}\right|_{(\kappa,\tau)=(\overline{\kappa},G(\overline{\kappa}))} = 0 \qquad \text{for all } \overline{\kappa} \in U.$$

By Lemma 7.9 it holds that $\tau_0 \in (M_{\alpha,\beta}(\kappa_0), \beta\kappa_0)$. So there is an environment $U' \subseteq U$ of κ_0 such that $G(\kappa) \in (M_{\alpha,\beta}(\kappa), \beta\kappa)$ for all $\kappa \in U'$. By Lemma 7.7 it also holds that $g^{[\kappa]}$ is strictly convex for every $\kappa \in U'$. Thus, by the definition of τ^*, we have $G(\kappa) = \tau^*(\kappa)$ for every $\kappa \in U'$.

This implies that $\tau^*(\kappa)$ is continuously differentiable in a neighborhood of κ_0. Since this holds for all $\kappa_0 \in \left(0, \frac{1}{\beta}\right)$, it follows that $\tau^*(\kappa)$ is continuously differentiable on $\left(0, \frac{1}{\beta}\right)$. Moreover, Theorem 7.11 further implies

$$G'(\kappa_0) = \left.\frac{\partial \tau^*(k)}{\partial \kappa}\right|_{\kappa=\kappa_0} = -\frac{(\partial^2_{\kappa,\tau} g_{\alpha,\beta,c})(\kappa_0, \tau_0)}{(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa_0, \tau_0)}. \qquad \square$$

Lemma 7.13. *For every simple $\alpha, \beta, c \geq 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$ It holds that*

$$\left.\frac{\partial^2 g^*_{\alpha,\beta,c}(\kappa)}{\partial \kappa^2}\right|_{\kappa=\kappa_0} < 0.$$

Proof. Let $\kappa_0 \in \left(0, \frac{1}{\beta}\right)$ and $\tau_0 = \tau^*_{\alpha,\beta,c}(\kappa_0)$. Using the chain rule for differentiation we get

$$
\begin{aligned}
\left.\frac{\partial g^*(\kappa)}{\partial \kappa}\right|_{\kappa=\kappa_0} &= \left.\frac{\partial g(\kappa, \tau)}{\partial \kappa}\right|_{(\kappa,\tau)=(\kappa_0,\tau_0)} + \left.\frac{\partial g(\kappa, \tau)}{\partial \tau}\right|_{(\kappa,\tau)=(\kappa_0,\tau_0)} \cdot \left.\frac{\partial \tau^*(\kappa)}{\partial \kappa}\right|_{\kappa=\kappa_0} \\
&= \left.\frac{\partial g(\kappa, \tau)}{\partial \kappa}\right|_{(\kappa,\tau)=(\kappa_0,\tau_0)} \\
&= (\partial_\kappa g)(\kappa_0, \tau^*(\kappa_0))
\end{aligned}
\tag{47}
$$

where the second equality follows from $\left.\frac{\partial g(\kappa, \tau)}{\partial \tau}\right|_{(\kappa,\tau)=(\kappa_0,\tau_0)} = 0$ by Lemma 7.9. By (47) and using Lemma 7.12 we get

$$
\begin{aligned}
\left.\frac{\partial^2 g^*(\kappa)}{\partial^2 \kappa}\right|_{\kappa=\kappa_0} &= (\partial^2_{\kappa,\kappa} g)(\kappa_0, \tau_0) + \left.\frac{\partial \tau^*(\kappa)}{\partial \kappa}\right|_{\kappa=\kappa_0} \cdot (\partial^2_{\kappa,\tau} g)(\kappa_0, \tau_0) \\
&= (\partial^2_{\kappa,\kappa} g)(\kappa_0, \tau_0) - \frac{(\partial^2_{\kappa,\tau} g)(\kappa_0, \tau_0)}{(\partial^2_{\tau,\tau} g)(\kappa_0, \tau_0)} \cdot (\partial^2_{\kappa,\tau} g)(\kappa_0, \tau_0) \\
&= \frac{1}{(\partial^2_{\tau,\tau} g)(\kappa_0, \tau_0)} \cdot \left((\partial^2_{\kappa,\kappa} g)(\kappa_0, \tau_0) \cdot (\partial^2_{\tau,\tau} g)(\kappa_0, \tau_0) - \left((\partial^2_{\kappa,\tau} g)(\kappa_0, \tau_0)\right)^2\right)
\end{aligned}
$$

41

$$= \frac{1}{(\partial^2_{\tau,\tau} g)(\kappa_0, \tau_0)} \cdot |H_g(\kappa_0, \tau_0)|$$
$$< 0,$$

where the last inequality follows from Lemmas 7.7 and 7.10. $\qquad \square$

It can also be easily shown using standard calculus arguments that $g^{[\kappa]}_{\alpha,\beta,c}(\kappa)$ is continuous in the closed interval $\left[0, \frac{1}{\beta}\right]$ (see, e.g., [61]). Thus, the following corollary is an immediate consequence of Lemma 7.13.

Corollary 7.14. *For every simple $\alpha, \beta, c \geq 1$ it holds that $g^*_{\alpha,\beta,c}$ is concave on the interval $\left[0, \frac{1}{\beta}\right]$.*

To complete the proof Lemma 2.15 we need to handle the corner cases excluded from Corollary 7.14.

Lemma 7.15. *For all $\beta > \alpha \geq 1$ and $c = 1$, it holds that $g^*_{\alpha,\beta,c}(\kappa)$ is concave in the interval $\left[0, \frac{1}{\beta}\right]$.*

Proof. For every $\kappa \in \left[0, \frac{1}{\beta}\right]$, by the definition of τ^*, it holds that

$$
\begin{aligned}
g_{\alpha,\beta,1}\left(\kappa, \tau^*_{\alpha,\beta,1}(\kappa)\right) &\leq g\left(\kappa, M_{\alpha,\beta}(\kappa)\right) \\
&= \frac{\beta \cdot \kappa - M_{\alpha,\beta}(\kappa)}{\alpha} \cdot \ln(1) - M_{\alpha,\beta}(\kappa) \cdot \mathcal{H}\left(\gamma\left(\kappa, M_{\alpha,\beta}(\kappa)\right)\right) \\
&\quad - (1 - M_{\alpha,\beta}(\kappa)) \cdot \mathcal{H}\left(\delta\left(\kappa, M_{\alpha,\beta}(\kappa)\right)\right) + \mathcal{H}(\kappa) \\
&= \mathcal{H}(\kappa) \cdot \left(-M_{\alpha,\beta}(\kappa) - \left(1 - M_{\alpha,\beta}(\kappa)\right) + 1\right) \\
&= 0,
\end{aligned}
$$

where the second equality follows from Lemma 7.3. By Lemma 7.4 we also have $g_{\alpha,\beta,1}\left(\kappa, \tau^*_{\alpha,\beta,1}(\kappa)\right) \geq 0$. So $g^*_{\alpha,\beta,1}(\kappa) = g_{\alpha,\beta,1}\left(\kappa, \tau^*_{\alpha,\beta,1}(\kappa)\right) = 0$. Thus, the function is trivially concave. $\qquad \square$

Another easy to handle corner case occurs when $\alpha = \beta$ and $c = 1$.

Lemma 7.16. *Let $\alpha > 1$ and $c = 1$, then $g^*_{\alpha,\alpha,c}(\kappa) = 0$ for all $\kappa \in \left[0, \frac{1}{\beta}\right]$.*

Proof. Let $\kappa \in \left[0, \frac{1}{\beta}\right]$. By Lemma 7.4, for every $\tau \in [M_{\alpha,\alpha}(\kappa), \alpha \cdot \kappa] = [0, \alpha\kappa]$, it holds that $g_{\alpha,\alpha,c}(\kappa, \tau) \geq 0$. Furthermore,

$$
\begin{aligned}
g_{\alpha,\alpha,c}(\kappa, 0) &= \left(\kappa - \frac{0}{\alpha}\right) \cdot \ln c - 0 \cdot \mathcal{H}\left(\gamma_{\alpha,\alpha}(\kappa, 0)\right) - (1 - 0) \cdot \mathcal{H}\left(\delta_{\alpha,\alpha}(\kappa, 0)\right) + \mathcal{H}(\kappa) \\
&= -\left(\kappa - \frac{0}{\alpha}\right) \cdot 0 - \mathcal{H}(\kappa) + \mathcal{H}(\kappa) = 0.
\end{aligned}
$$

So we have
$$g^*_{\alpha,\alpha,c}(\kappa) = \min_{\tau \in [0, \alpha \cdot \kappa]} g_{\alpha,\alpha,c}(\kappa, \tau) = 0.$$

$\qquad \square$

We are left to handle the case in which $\alpha = \beta > 1$ and $c > 1$. The analysis for this case is based on ideas from [26]. The analysis is also used as part of the proof of Lemma 2.19. We first provide an explicit formula for $g^*(\kappa)$ in this case.

Lemma 7.17. *Let $\alpha > 1$ and $c > 1$. Then*

$$
g^*_{\alpha,\alpha,c}(\kappa) = \kappa \cdot \ln c - \begin{cases} 0 & \kappa < \delta^* \\ \mathcal{D}\left(\kappa \| \delta^*\right) & \kappa \geq \delta^* \end{cases}.
$$

for all $\kappa \in \left(0, \frac{1}{\alpha}\right)$ where $\delta^* \in \left(0, \frac{1}{\alpha}\right)$ is the unique value which satisfies $\mathcal{D}\left(\frac{1}{\alpha} \| \delta^*\right) = \frac{\ln c}{\alpha}$.

In the proof of Lemma 7.17 we use the following identity.

Lemma 7.18. *For all $a, b \in (0, 1)$ it holds that*

$$\mathcal{H}(a) - (a - b) \cdot \ln\left(\frac{1-a}{a}\right) = \mathcal{D}\left(b\|a\right) + \mathcal{H}(b).$$

Proof. By expanding the term $\mathcal{H}(a)$ we get

$$
\begin{aligned}
\mathcal{H}(a) - (a - b) \cdot \ln\left(\frac{1-a}{a}\right) &= -a \ln a - (1-a) \ln(1-a) - (a-b) \ln(1-a) + (a-b) \ln a \\
&= -b \ln a - (1-b) \ln(1-a) \\
&= b \ln\left(\frac{b}{a}\right) + (1-b) \ln\left(\frac{1-b}{1-a}\right) - b \ln b - (1-b) \ln(1-b) \\
&= \mathcal{D}\left(b\|a\right) + \mathcal{H}(b).
\end{aligned}
$$

$\square$

Proof of Lemma 7.17. Define $\tilde{\tau}(\kappa) = \frac{\kappa - \delta^*}{\frac{1}{\alpha} - \delta^*}$. It can be easily verified that $\delta_{\alpha,\alpha}(\kappa, \tilde{\tau}(\kappa)) = \delta^*$ for all $\kappa \in \left[0, \frac{1}{\alpha}\right]$. By Lemma 7.5 we have

$$
\begin{aligned}
(\partial_\tau g_{\alpha,\alpha,c})(\kappa, \tilde{\tau}(\kappa)) &= -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha}\Big\|\gamma_{\alpha,\alpha}(\kappa, \tilde{\tau}(\kappa))\right) + \mathcal{D}\left(\frac{1}{\alpha}\Big\|\delta_{\alpha,\alpha}(\kappa, \tilde{\tau}(\kappa))\right) \\
&= -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha}\Big\|\frac{1}{\alpha}\right) + \mathcal{D}\left(\frac{1}{\alpha}\Big\|\delta^*\right) \\
&= -\frac{\ln(c)}{\alpha} + -\frac{\ln(c)}{\alpha} = 0,
\end{aligned}
$$

where the third equality follows from the definition of δ^*. By Lemma 2.13 it holds that $g_{\alpha,\alpha,c}(\kappa, \tau)$ is convex as a function of τ, and we can conclude that the function has a global minimum at $\tilde{\tau}(\kappa)$.

It also holds that

$$\tilde{\tau}(\kappa) = \frac{\kappa - \delta^*}{\frac{1}{\alpha} - \delta^*} = \frac{\alpha \cdot \kappa \cdot \left(\frac{1}{\alpha} - \delta^*\right) + \alpha \kappa \delta^* - \delta^*}{\frac{1}{\alpha} - \delta^*} \leq \alpha \kappa$$

for all $\kappa \leq \frac{1}{\alpha}$.

Thus, for every $\delta^* \leq \kappa \leq \frac{1}{\beta}$ it holds that $\tilde{\tau}(\kappa) \geq 0$, and hence,

$$
\begin{aligned}
g^*_{\alpha,\alpha,c}(\kappa) &= \min_{0 \leq \tau \leq \alpha\kappa} g_{\alpha,\alpha,c}(\kappa, \tau) \\
&= g_{\alpha,\alpha,c}(\kappa, \tilde{\tau}(\kappa)) \\
&= \left(\kappa - \frac{\tilde{\tau}(\kappa)}{\alpha}\right) \cdot \ln c - \tilde{\tau}(\kappa) \cdot \mathcal{H}\left(\frac{1}{\alpha}\right) - (1 - \tilde{\tau}(\kappa)) \cdot \mathcal{H}\left(\delta_{\alpha,\alpha}(\kappa, \tilde{\tau}(\kappa))\right) + \mathcal{H}(\kappa) \\
&= \kappa \cdot \ln c - \tilde{\tau}(\kappa) \cdot \mathcal{D}\left(\frac{1}{\alpha}\Big\|\delta^*\right) - \tilde{\tau}(\kappa) \cdot \mathcal{H}\left(\frac{1}{\alpha}\right) - (1 - \tilde{\tau}(\kappa)) \cdot \mathcal{H}(\delta^*) + \mathcal{H}(\kappa) \\
&= \kappa \cdot \ln c - \tilde{\tau}(\kappa) \cdot \left(\mathcal{H}(\delta^*) - \left(\delta^* - \frac{1}{\alpha}\right) \cdot \ln\left(\frac{1 - \delta^*}{\delta^*}\right)\right) - (1 - \tilde{\tau}(\kappa)) \cdot \mathcal{H}(\delta^*) + \mathcal{H}(\kappa) \\
&= \kappa \cdot \ln c + \tilde{\tau}(\kappa) \cdot \left(\delta^* - \frac{1}{\alpha}\right) \cdot \ln\left(\frac{1 - \delta^*}{\delta^*}\right) - \mathcal{H}(\delta^*) + \mathcal{H}(\kappa)
\end{aligned}
$$

The forth equality holds as $\frac{\ln c}{\alpha} = \mathcal{D}\left(\frac{1}{\alpha}\|\delta^*\right)$, and the fifth equality follows from Lemma 7.18. By the definition of $\tilde{\tau}(\kappa)$ we have $\tilde{\tau}(\kappa)\left(\delta - \frac{1}{\alpha}\right) = (\delta^* - \kappa)$, thus for every $\delta^* \leq \kappa < \frac{1}{\beta}$ we have

$$
\begin{aligned}
g^*_{\alpha,\alpha,c}(\kappa) &= \kappa \cdot \ln(c) + \tilde{\tau}(\kappa) \cdot \left(\delta^* - \frac{1}{\alpha}\right) \cdot \ln\left(\frac{1 - \delta^*}{\delta^*}\right) - \mathcal{H}(\delta^*) + \mathcal{H}(\kappa) \\
&= \kappa \cdot \ln(c) + (\delta^* - \kappa^*) \cdot \ln\left(\frac{1 - \delta^*}{\delta^*}\right) - \mathcal{H}(\delta^*) + \mathcal{H}(\kappa) \\
&= \kappa \cdot \ln(c) - \mathcal{D}\left(\kappa\|\delta^*\right),
\end{aligned}
\tag{48}
$$

where the last equality follows from Lemma 7.18.

Also, for every $0 \leq \kappa < \delta^*$ it holds that $\tilde{\tau}(\kappa) < 0$. Hence, since $g_{\alpha,\alpha,c}(\kappa,\tau)$ is convex as a function of τ (Lemma 7.7), it holds that

$$
\begin{aligned}
g^*_{\alpha,\alpha,c}(\kappa) &= \min_{0 \leq \tau \leq \alpha\kappa} g_{\alpha,\alpha,c}(\kappa,\tau) \\
&= g_{\alpha,\alpha,c}(\kappa,0) \\
&= \left(\kappa - \frac{0}{\alpha}\right) \cdot \ln c - 0 \cdot \mathcal{H}\left(\frac{1}{\alpha}\right) - (1-0) \cdot \mathcal{H}\left(\delta_{\alpha,\alpha}(\kappa,0)\right) + \mathcal{H}(\kappa) \quad (49) \\
&= \kappa \cdot \ln c - \mathcal{H}(\kappa) + \mathcal{H}(\kappa) \\
&= \kappa \cdot \ln c.
\end{aligned}
$$

By (48) and (49) it holds that

$$
g^*_{\alpha,\alpha,c}(\kappa) = \kappa \cdot \ln c - \begin{cases} 0 & \kappa < \delta^* \\ \mathcal{D}(\kappa\|\delta^*) & \kappa \geq \delta^* \end{cases}.
$$

$\square$

Observe the function

$$
\zeta(x) = \begin{cases} 0 & x < \delta^* \\ \mathcal{D}(x\|\delta^*) & x \geq \delta^* \end{cases}
$$

is convex. Thus the following is a corollary of Lemma 7.17 and the continuity of $g^*_{\alpha,\alpha,c}(\kappa)$.

Corollary 7.19. *For every $\alpha > 1$ and $c > 1$ it holds that $g^*_{\alpha,\alpha,c}(\kappa)$ is concave on $\left[0,\frac{1}{\alpha}\right]$.*

We can now proceed to the proof of Lemma 2.15.

Lemma 2.15. *For all $\alpha, c \geq 1$ and $\beta > 1$, it holds that $g^*_{\alpha,\beta,c}(\kappa)$ is concave in the interval $\left[0,\frac{1}{\beta}\right]$.*

Proof. The lemma follows immediately from Corollary 7.14, Lemmas 7.15 and 7.16, and Corollary 7.19.

$\square$

We also use the formula in Lemma 7.17 to prove Lemma 2.19.

Lemma 2.19. *For every $\beta, c > 1$ it holds that $\mathtt{best}(\beta,c,\beta) = \mathtt{amls}_{\alpha=\beta}(\beta,c)$.*

Proof. Let $\beta, c > 1$ and let $\delta^* \in \left(0,\frac{1}{\beta}\right)$ be the unique value such that $\mathcal{D}\left(\frac{1}{\beta}\middle\|\delta^*\right) = \frac{\ln c}{\beta}$. Then by Lemma 7.17 it holds that

$$
\max_{0 \leq \kappa \leq \frac{1}{\beta}} g^*_{\beta,\beta,c}(\kappa) = \max_{0 \leq \kappa \leq \frac{1}{\beta}} \left(\kappa \cdot \ln c - \begin{cases} 0 & \kappa < \delta^* \\ \mathcal{D}(\kappa\|\delta^*) & \kappa \geq \delta^* \end{cases}\right) = \max_{\delta^* \leq \kappa \leq \frac{1}{\beta}} \left(\kappa \cdot \ln c - \mathcal{D}(\kappa\|\delta^*)\right).
$$

Define $h(\kappa) = \kappa \ln c - \mathcal{D}(\kappa\|\delta^*)$. Then, by Corollary 2.12 and (11), we get

$$
\mathtt{best}(\beta,c,\beta) = \mathtt{amls}(\beta,c,\beta) = \exp\left(\max_{0 \leq \kappa \leq \frac{1}{\beta}} g^*_{\beta,\beta,c}(\kappa)\right) = \exp\left(\max_{\delta^* \leq \kappa \leq \frac{1}{\beta}} h(\kappa)\right). \quad (50)
$$

Let h' be the derivative of h, and observe that $\frac{\partial \mathcal{D}(a\|b)}{\partial a} = \ln\left(\frac{a}{1-a} \cdot \frac{1-b}{b}\right)$. So

$$
h'(\kappa) = \ln c - \ln\left(\frac{\kappa}{1-\kappa} \cdot \frac{1-\delta^*}{\delta^*}\right). \quad (51)
$$

Since h is concave it can be trivially deduced that $h'(\kappa)$ is increasing in $\kappa \in [\delta^*, \kappa^*]$. Furthermore,

$$
h'(\delta^*) = \ln(c) - \ln\left(\frac{\delta^*}{1-\delta^*} \cdot \frac{1-\delta^*}{\delta^*}\right) = \ln(c) > 1 \quad (52)
$$

44

and

$$
\begin{aligned}
h'\left(\frac{1}{\beta}\right) &= \ln(c) - \ln\left(\frac{\frac{1}{\beta}}{1 - \frac{1}{\beta}} \cdot \frac{1 - \delta^*}{\delta^*}\right) \\
&= \beta \cdot \mathcal{D}\left(\frac{1}{\beta}\middle\| \delta^*\right) - \ln\left(\frac{\frac{1}{\beta}}{1 - \frac{1}{\beta}} \cdot \frac{1 - \delta^*}{\delta^*}\right) \\
&= \beta \cdot \frac{1}{\beta} \cdot \ln\left(\frac{\left(\frac{1}{\beta}\right)}{\delta^*}\right) + \beta \cdot \left(1 - \frac{1}{\beta}\right) \cdot \ln\left(\frac{1 - \frac{1}{\beta}}{1 - \delta^*}\right) - \ln\left(\frac{\frac{1}{\beta}}{1 - \frac{1}{\beta}} \cdot \frac{1 - \delta^*}{\delta}\right) \\
&= \beta \cdot \ln\left(\frac{1 - \frac{1}{\beta}}{1 - \delta^*}\right) < 0,
\end{aligned}
\tag{53}
$$

where the second equality uses $\frac{\ln c}{\beta} = \mathcal{D}\left(\frac{1}{\beta}\middle\| \delta^*\right)$ and the last inequality holds since $\delta < \frac{1}{\beta}$. By (52) and (53), there is $k^* \in \left(\delta^*, \frac{1}{\beta}\right)$ such that $h'(\kappa^*) = 0$. Furthermore, by (51) and simple algebraic manipulation, we get $\kappa^* = \frac{c \cdot \delta^*}{1 + \delta^*(c-1)}$. Therefore,

$$
\max_{\delta^* \leq \kappa \leq \frac{1}{\beta}} h(\kappa) = h(\kappa^*).
\tag{54}
$$

It also holds that

$$
\begin{aligned}
\mathcal{D}(\kappa^* \| \delta^*) &= \kappa^* \cdot \ln\left(\frac{\kappa^*}{\delta^*}\right) + (1 - \kappa^*) \cdot \ln\left(\frac{1 - \kappa^*}{1 - \delta^*}\right) \\
&= \kappa^* \cdot \left(\frac{\kappa^*}{1 - \kappa^*} \cdot \frac{1 - \delta^*}{\delta^*}\right) + \ln\left(\frac{1 - \kappa^*}{1 - \delta^*}\right) \\
&= \kappa^* \ln(c) + \ln\left(\frac{1 - \kappa^*}{1 - \delta^*}\right),
\end{aligned}
$$

where the last equality follows from $h'(\kappa^*) = 0$ and (51). Thus,

$$
h(\kappa^*) = \kappa^* \ln(c) - \mathcal{D}(\kappa^* \| \delta^*) = -\ln\left(\frac{1 - \kappa^*}{1 - \delta^*}\right) = -\ln\left(\frac{1 - \frac{c \cdot \delta^*}{1 + \delta^*(c-1)}}{1 - \delta^*}\right) = \ln\left(1 + (c-1)\delta^*\right).
$$

By the above equitation, (50) and (54) we have $\texttt{best}(\beta, c, \beta) = 1 + (c-1) \cdot \delta^*$. Thus, $1 \leq \texttt{best}(\beta, c, \beta) \leq 1 + \frac{c-1}{\beta}$ and

$$
\mathcal{D}\left(\frac{1}{\beta}\middle\| \frac{\texttt{best}(\beta, c, \beta) - 1}{c - 1}\right) = \mathcal{D}\left(\frac{1}{\beta}\middle\| \delta^*\right) = \frac{\ln(c)}{\beta}.
$$

So $\texttt{best}(\beta, c, \beta) = \texttt{amls}_{\alpha=\beta}(\beta, c)$ by the definition of $\texttt{amls}_{\alpha=\beta}$. $\qquad\square$

7.4 The Determinant of the Hessian is Negative

In this section we prove Lemma 7.10, that is, we show the determinant of the Hessian of g is negative. To do so we first obtain an explicit formula for the Hessian. Recall Γ and Δ are defined in (42) and (43).

Lemma 7.20. *Let $\alpha \geq 1$, $\beta > 1$ such that $\alpha \neq \beta$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. Then*

$$
|H_g(\kappa, \tau)| = \frac{\Gamma_{\alpha,\beta}(\kappa, \tau) \cdot \Delta_{\alpha,\beta}(\kappa, \tau)}{\alpha^2 \cdot (1 - \kappa)} \cdot \frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma_{\alpha,\beta}(\kappa, \tau) - \delta_{\alpha,\beta}(\kappa, \tau))}{\gamma_{\alpha,\beta}(\kappa, \tau) - \frac{1}{\alpha}} \cdot
$$
$$
\left(A_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa, \tau)\right) + \delta_{\alpha,\beta}(\kappa, \tau) \cdot B_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa, \tau)\right)\right),
$$

where

$$
\begin{aligned}
A_{\alpha,\beta}(x) &:= -2 + x\left(1 + \alpha + \beta\right) - \alpha \cdot \beta \cdot x^2 \qquad \text{and} \\
B_{\alpha,\beta}(x) &:= x \cdot \alpha \cdot (\beta - 2) + 1 + \alpha - \beta.
\end{aligned}
$$

The formula in Lemma 7.20 is derived from a technical computation of $|H_g(\kappa, \tau)|$ followed by re-arrangement of the terms. We defer the proof of Lemma 7.20 to Section 7.6. We use the notation $A_{\alpha,\beta}$ and $B_{\alpha,\beta}$ (or just A and B) to refer to the functions defined in Lemma 7.20.

As before, we often omit the subscripts α, β from functions (e.g., $\delta(\kappa, \tau)$ instead of $\delta_{\alpha,\beta}(\kappa, \tau)$) when α, β are known by context. Furthermore, we often also omit the function parameters (κ, τ) when known by context (e.g., δ instead of $\delta_{\alpha,\beta}(\kappa, \tau)$).

By Lemmas 7.1 and 7.2, for every $\alpha \geq 1$, $\beta > 1$ such that $\alpha \neq \beta$ $\kappa \in (0, \frac{1}{\beta})$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$ it holds that $\delta_{\alpha,\beta}(\kappa, \tau), \gamma_{\alpha,\beta}(\kappa, \tau) \in (0, 1)$. So $\Gamma_{\alpha,\beta}(\kappa, \tau)$, $\Delta_{\alpha,\beta}(\kappa, \tau) > 0$. Also, by Lemma 7.2, if $\beta < \alpha$ it holds that $\left(1 - \frac{\beta}{\alpha}\right) > 0$ and $\gamma - \frac{1}{\alpha} > 0$, and if $\beta > \alpha$ it holds that $\left(1 - \frac{\beta}{\alpha}\right) < 0$ and $\gamma - \frac{1}{\alpha} < 0$. Hence,

$$\frac{\left(1 - \frac{\beta}{\alpha}\right)}{\gamma_{\alpha,\beta}(\kappa, \tau) - \frac{1}{\alpha}} > 0$$

in both cases. Following the above argument and Lemma 7.20 we attain the next corollary.

Corollary 7.21. *Let $\alpha \geq 1$, $\beta > 1$ such that $\alpha \neq \beta$, $\kappa \in (0, \frac{1}{\beta})$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. Then $|H_g(\kappa, \tau)| < 0$ if and only if*

$$(A_{\alpha,\beta}(\gamma_{\alpha,\beta}(\kappa, \tau)) + \delta_{\alpha,\beta}(\kappa, \tau) \cdot B_{\alpha,\beta}(\gamma_{\alpha,\beta}(\kappa, \tau))) \cdot (\gamma_{\alpha,\beta}(\kappa, \tau) - \delta_{\alpha,\beta}(\kappa, \tau)) < 0.$$

The following lemma allows us to determine the sign of $|H_g(\kappa, \tau)|$ using an even simpler expression.

Lemma 7.22. *Let $\alpha \geq 1$, $\beta > 1$ such that $\alpha \neq \beta$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. Then $\gamma_{\alpha,\beta}(\kappa, \tau) > \delta_{\alpha,\beta}(\kappa, \tau)$.*

Proof. Consider the following cases.

- If $\alpha > \beta > 1$, by Lemmas 7.1 and 7.2, we immediately have that $\delta(\kappa, \tau) < \frac{1}{\alpha} < \frac{1}{\beta} \leq \gamma(\kappa, \tau)$.

- If $\beta > \alpha \geq 1$, then by Lemma 7.2 it holds that $\gamma > \kappa$. Furthermore, by Lemma 7.1

$$\delta(\kappa, \tau) \leq \delta(\kappa, M_{\alpha,\beta}(\kappa, \tau)) = \frac{\frac{\beta}{\alpha}\kappa - \frac{1}{\alpha}}{1 - M_{\alpha,\beta}(\kappa)} + \frac{1}{\alpha} = \frac{\frac{\beta}{\alpha}\kappa - \frac{1}{\alpha}}{1 - \frac{\beta - \alpha}{1 - \alpha\kappa} \cdot \kappa} + \frac{1}{\alpha} = \frac{\frac{\beta}{\alpha}\kappa - \frac{1}{\alpha}}{\frac{1 - \alpha\kappa - (\beta - \alpha)\kappa}{1 - \alpha\kappa}} + \frac{1}{\alpha}$$

$$= \frac{\frac{1}{\alpha}(\beta\kappa - 1)}{1 - \beta\kappa} \cdot (1 - \alpha\kappa) + \frac{1}{\alpha} = \kappa.$$

Thus, $\delta \leq \kappa < \gamma$. $\square$

By Corollary 7.23 and Lemma 7.22 we obtain the following.

Corollary 7.23. *Let $\alpha \geq 1$, $\beta > 1$ such that $\alpha \neq \beta$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. Then $|H_g(\kappa, \tau)| < 0$ if and only if*

$$A_{\alpha,\beta}(\gamma_{\alpha,\beta}(\kappa, \tau)) + \delta_{\alpha,\beta}(\kappa, \tau) \cdot B_{\alpha,\beta}(\gamma_{\alpha,\beta}(\kappa, \tau)) < 0.$$

We proceed to analyze the functions $A_{\alpha,\beta}$ and $B_{\alpha,\beta}$ towards our goal of showing that $A_{\alpha,\beta}(\gamma) + \delta \cdot B_{\alpha,\beta}(\gamma)$ is negative.

Lemma 7.24. *For every $\alpha, \beta \geq 1$ such that $\alpha \neq \beta$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in \left(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa\right)$ it holds that $A_{\alpha,\beta}(\gamma(\kappa, \tau)) < 0$.*

Proof. We can re-write $A_{\alpha,\beta}$ (as defined in Lemma 7.20) as

$$A_{\alpha,\beta}(x) = -\alpha \cdot \beta \cdot \left(x - \frac{1 + \alpha + \beta}{2 \cdot \alpha \cdot \beta}\right)^2 - 2 + \frac{(\alpha + \beta + 1)^2}{4 \cdot \alpha \cdot \beta}.$$

Note that $A_{\alpha,\beta}(x)$ is a quadratic polynomial in x, which reaches its maximum value at $x_0 := \frac{1 + \alpha + \beta}{2 \cdot \alpha \cdot \beta}$. Consider the following cases.

- If $x_0 \le 1$, then for every $x \in (0,1)$ we have

$$A_{\alpha,\beta}(x) \ \le \ A_{\alpha,\beta}(x_0) \ = \ -\alpha \cdot \beta(x_0 - x_0)^2 - 2 + x_0^2 \ = \ -2 + x_0^2 \ < \ 0,$$

where the last inequality holds as $0 \le x_0 \le 1$.

- If $x_0 > 1$, then $A_{\alpha,\beta}(x)$ is monotonically increasing in $[0,1]$. Thus, for every $x \in (0,1)$, it holds that

$$A_{\alpha,\beta}(x) \ < \ A_{\alpha,\beta}(1) \ = \ -2 + (1 + \alpha + \beta) - \alpha \cdot \beta \ = \ -(\alpha - 1)(\beta - 1) \ \le 0.$$

So overall $A_{\alpha,\beta}(x) < 0$ for all $x \in (0,1)$. By Lemma 7.2 it holds that $\gamma_{\alpha,\beta}(\kappa,\tau) \in (0,1)$. So $A_{\alpha,\beta}(\gamma_{\alpha,\beta}(\kappa,\tau)) < 0$. $\qquad\square$

The tools attained so far suffice to show that $A(\gamma) + \delta \cdot B(\gamma)$ is negative in case $\alpha < \beta$.

Lemma 7.25. *For all $\beta > \alpha \ge 1$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in \left(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa\right)$ it holds that*

$$A\Big(\gamma(\kappa,\tau)\Big) + \delta(\kappa,\tau) \cdot B\Big(\gamma(\kappa,\tau)\Big) < 0.$$

Proof. Consider the following cases.

- If $B(\gamma(\kappa,\tau)) \le 0$, then $\delta \ge 0$ by Lemma 7.1. So

$$A\Big(\gamma(\kappa,\tau)\Big) + \delta(\kappa,\tau) \cdot B\Big(\gamma(\kappa,\tau)\Big) \ \le \ A\Big(\gamma(\kappa,\tau)\Big) < 0,$$

where the last inequality follows from Lemma 7.24.

- If $B(\gamma(\kappa,\tau)) > 0$, we can use $\gamma_{\alpha,\beta}(\kappa,\tau) > \delta_{\alpha,\beta}(\kappa,\tau)$ (Lemma 7.22) to obtain

$$\begin{aligned} A(\gamma) + \delta \cdot B(\gamma) &< A(\gamma) + \gamma \cdot B(\gamma) \\ &= -2 + \gamma \left(1 + \beta + \alpha\right) - \beta \cdot \alpha \cdot \gamma^2 + \alpha \cdot (\beta - 2) \cdot \gamma^2 + \gamma \cdot (1 + \alpha - \beta) \\ &= \gamma^2 \cdot (-2\alpha) + \gamma \cdot (2(\alpha + 1)) - 2 \\ &= -2 \cdot (1 - \gamma) \cdot (1 - \alpha \cdot \gamma) \ \le \ 0, \end{aligned}$$

where the last inequality follows from $\gamma \le \frac{1}{\alpha}$ by Lemma 7.2. $\qquad\square$

Lemma 7.25 suffices to show Lemma 7.10 for the case $\beta > \alpha \ge 1$.

Lemma 7.26. *For all $\beta > \alpha \ge 1$, $c > 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$ it holds that $|H_g(\kappa, \tau^*(\kappa))| < 0$.*

Proof. By Lemma 7.9 it holds that $\tau^*(\kappa) \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$ and $(\partial_\tau g)(\kappa, \tau^*(\kappa)) = 0$. By Lemma 7.25 we have

$$A\Big(\gamma(\kappa, \tau^*(\kappa))\Big) + \delta(\kappa, \tau^*(\kappa)) \cdot B\Big(\gamma(\kappa, \tau^*(\kappa))\Big) < 0.$$

Thus, by Corollary 7.23, we get $|H_g(\kappa, \tau^*(\kappa))| < 0$. $\qquad\square$

Note that Lemma 7.25 implies a stronger claim than the one stated in Lemma 7.26: for all $\beta > \alpha \ge 1$ and (κ, τ) the determinant Hessian evaluated at (κ, τ) is negative. This property, however, does not hold if $\alpha > \beta > 1$, i.e., in this case the determinant of the Hessian may be positive for some values of (κ, τ). We use the following lemma to restrict the possible values κ and τ may take.

Lemma 7.27. *For all $\alpha > \beta > 1$, $c \ge 1$, $\kappa \in (0, \frac{1}{\beta})$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$ such that $(\partial_\tau g)(\kappa, \tau) = 0$, it holds that $\mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma_{\alpha,\beta}(\kappa,\tau)\right) \le \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta_{\alpha,\beta}(\kappa,\tau)\right)$.*

Proof. By Lemma 7.5, the condition $(\partial_\tau g)(\kappa, \tau) = 0$ is equivalent to

$$-\mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma(\kappa,\tau)\right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta(\kappa,\tau)\right) = \frac{\ln(c)}{\alpha} \ge 0$$

since $c \ge 1$. Therefore, $\mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma_{\alpha,\beta}(\kappa,\tau)\right) \le \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta_{\alpha,\beta}(\kappa,\tau)\right)$. $\qquad\square$

The next lemmas enables us to further simplify the criteria in Corollary 7.23.

Lemma 7.28. *Let $\alpha > \beta > 1$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in \left(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa\right)$, then $B_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa,\tau)\right) > 0$.*

Proof. Consider the following cases.

- If $\beta \geq 2$, it holds that

$$B(\gamma) = \gamma \cdot \alpha \cdot (\beta - 2) + 1 + \alpha - \beta \geq 0 \cdot \alpha(\beta - 2) + 1 + \alpha - \beta = 1 + \alpha - \beta > 0,$$

  where the first inequality follows from $\gamma(\kappa,\tau) \geq 0$ (Lemma 7.2) and the last inequality uses $\beta < \alpha$.

- If $\beta < 2$, since $\gamma < 1$ (Lemma 7.2) we have that

$$B(\gamma) = \gamma \cdot (\beta - 2) \cdot \alpha + 1 + \alpha - \beta > 1 \cdot \alpha (\beta - 2) + 1 + \alpha - \beta = (\beta - 1) \cdot (\alpha - 1) \geq 0.$$

$\square$

For every $\alpha > \beta > 1$ we define $C_{\alpha,\beta} : (0,1) \to \mathbb{R}$ via

$$C_{\alpha,\beta}(x) := -\frac{A_{\alpha,\beta}(x)}{B_{\alpha,\beta}(x)} \tag{55}$$

for all $x \in (0,1)$. Lemma 7.28 and corollary 7.23 imply the following.

Corollary 7.29. *Let $\alpha > \beta > 1$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in \left(M_{\alpha,\beta}(\kappa), \beta \cdot \kappa\right)$. Then $|H_g(\kappa,\tau)| < 0$ if and only if $C(\gamma_{\alpha,\beta}(\kappa,\tau)) > \delta_{\alpha,\beta}(\kappa,\tau)$.*

The following lemma utilizes Lemma 7.27 to show that the condition in Corollary 7.29 holds on critical points of $g_{\alpha,\beta,c}$.

Lemma 7.30. *Let $\alpha > \beta > 1$, $c \geq 1$, $\kappa \in (0, \frac{1}{\beta})$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$, such that $(\partial_\tau g)(\kappa,\tau) = 0$. Then $C_{\alpha,\beta}\left(\gamma(\kappa,\tau)\right) > \delta_{\alpha,\beta}(\kappa,\tau)$.*

Proof. The next claim allows us to eliminate the dependency on β.

Claim 7.31. $C_{\alpha,\beta}\left(\gamma(\kappa,\tau)\right) > \frac{\gamma_{\alpha,\beta}(\kappa,\tau)}{2 \cdot \alpha \cdot \gamma_{\alpha,\beta}(\kappa,\tau) - 1}$.

Proof. For every $\tilde{\beta} \in (1, \alpha)$ can rewrite

$$A_{\alpha,\tilde{\beta}}(x) = -2 + x\left(1 + \alpha + \tilde{\beta}\right) - \alpha\tilde{\beta}x^2 = -2 + x(1+\alpha) + \tilde{\beta}x \cdot (1 - \alpha x) = a(\alpha, x) - \tilde{\beta}x\left(\alpha x - 1\right), \tag{56}$$

where $a(\alpha, x) := -2 + x\left(1 + \alpha\right)$. Similarly, for every $\tilde{\beta} \in (1, \alpha)$ it holds that

$$B_{\alpha,\tilde{\beta}}(x) = x \cdot \alpha \cdot \left(\tilde{\beta} - 2\right) + 1 + \alpha - \tilde{\beta} = \tilde{\beta}\left(\alpha x - 1\right) - 2 \cdot x \cdot \alpha + 1 + \alpha = \tilde{\beta}(\alpha x - 1) + b(\alpha, x), \tag{57}$$

where $b(\alpha, x) := -2 \cdot x \cdot \alpha + 1 + \alpha$.

Using (56) and (57), for every $\tilde{\beta} \in (1, \alpha)$ and $x \in \left(\frac{1}{\alpha}, 1\right)$ it holds that

$$\begin{aligned}
A_{\alpha,\tilde{\beta}}(x) + x \cdot B_{\alpha,\tilde{\beta}}(x) &= a(\alpha, x) + x \cdot b(\alpha, x) \\
&= -2 + x(1 + \alpha) - 2 \cdot x^2 \cdot \alpha + x + \alpha x \\
&= 2 \cdot \left(-1 + x(1 + \alpha) - x^2 \cdot \alpha\right) > 0.
\end{aligned} \tag{58}$$

The last inequality holds as $\zeta(x) = -1 + x(1 + \alpha) - \alpha x^2$ is concave and $\zeta\left(\frac{1}{\alpha}\right) = \zeta(1) = 0$. Observe the sum $A_{\alpha,\tilde{\beta}}(x) + x \cdot B_{\alpha,\tilde{\beta}}(x)$ does not depend on $\tilde{\beta}$.

By (58) we have

$$C_{\alpha,\tilde{\beta}}(x) = -\frac{A_{\alpha,\tilde{\beta}(x)}}{B_{\alpha,\tilde{\beta}}(x)} = -\frac{A_{\alpha,\tilde{\beta}}(x) + x \cdot B_{\alpha,\tilde{\beta}}(x) - x \cdot B_{\alpha,\tilde{\beta}}(x)}{B_{\alpha,\tilde{\beta}}(x)} = x - \frac{a(\alpha, x) + x \cdot b(\alpha, x)}{B_{\alpha,\tilde{\beta}}(x)}. \tag{59}$$

For a fixed $x > \frac{1}{\alpha}$, by (57), we have $B_{\alpha,\tilde{\beta}}(x)$ is increasing as a function of $\tilde{\beta}$. Hence, by (58) and (59), the expression $C_{\alpha,\tilde{\beta}}(x)$ is increasing as a function of $\tilde{\beta}$. By Lemma 7.2 it holds that $\frac{1}{\alpha} < \frac{1}{\beta} < \gamma(\kappa,\tau) < 1$. Using the monotonicity property of $C_{\alpha,\tilde{\beta}}(x)$ we get

$$
\begin{aligned}
C_{\alpha,\beta}(\gamma) \;>\; & C_{\alpha,\frac{1}{\gamma}}(\gamma) \\
=\; & -\frac{A_{\alpha,\frac{1}{\gamma}}(\gamma)}{B_{\alpha,\frac{1}{\gamma}}(\gamma)} \\
=\; & -\frac{-2 + \gamma(1+\alpha) + \frac{1}{\gamma}\cdot\gamma\cdot(1-\alpha\cdot\gamma)}{\frac{1}{\gamma}(\alpha\gamma-1) - 2\cdot\gamma\cdot\alpha + 1 + \alpha} \\
=\; & -\frac{-1+\gamma}{2\alpha - 2\gamma\alpha + 1 - \frac{1}{\gamma}} \\
=\; & \frac{\gamma}{2\cdot\alpha\cdot\gamma - 1}
\end{aligned}
$$

where the last equality follows from a re-arrangement of terms. ⌟

We combine Claim 7.31 with the following inequality.

Claim 7.32. *For every $x \in \left[\frac{1}{\alpha}, 1\right)$ it holds that $\mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{x}{2\alpha x-1}\right) \leq \mathcal{D}\left(\frac{1}{\alpha}\middle\|x\right)$.*

Proof. Define $h(x) = \mathcal{D}\left(\frac{1}{\alpha}\middle\|x\right) - \mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{x}{2\alpha x-1}\right)$. The statement of the claim is equivalent to $h(x) \geq 0$ for all $x \in \left[\frac{1}{\alpha}, 0\right)$. Recall $\frac{\partial \mathcal{D}(a\|b)}{\partial b} = \frac{b-a}{b(1-b)}$. Therefore,

$$
\begin{aligned}
\frac{\partial}{\partial x}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{x}{2\alpha x-1}\right)\right) \;=\; & \frac{2\alpha x - 1 - x\cdot 2\alpha}{(2\alpha x - 1)^2}\cdot\frac{\frac{x}{2\alpha x-1} - \frac{1}{\alpha}}{\frac{x}{2\alpha x-1}\cdot\left(1 - \frac{x}{2\alpha x-1}\right)} \\
=\; & -\frac{\left(\frac{x - \frac{1}{\alpha}(2\alpha x-1)}{2\alpha x-1}\right)}{x(2\alpha x - 1 - x)} \\
=\; & \frac{x - \frac{1}{\alpha}}{x(2\alpha x - 1 - x)(2\alpha x - 1)}.
\end{aligned}
$$

Let h' be the derivative of h. Thus,

$$
h'(x) \;=\; \frac{x - \frac{1}{\alpha}}{x(1-x)} - \frac{x - \frac{1}{\alpha}}{x(2\alpha x - 1 - x)(2\alpha x - 1)} = \frac{x - \frac{1}{\alpha}}{x}\left(\frac{1}{1-x} - \frac{1}{(2\alpha x - 1 - x)(2\alpha x - 1)}\right).
$$

For every $x \in \left[\frac{1}{\alpha}, 1\right)$ it holds that $1-x,\ x-\frac{1}{\alpha},\ 2\alpha x - 1 - x,\ 2\alpha x - 1 \;\geq\; 0$. So

$$
\begin{aligned}
h'(x) \;\geq\; & 0 & \Longleftrightarrow \\
1 - x \;\leq\; & (2\cdot\alpha x - x - 1)\cdot(2\alpha x - 1) & \Longleftrightarrow \\
1 - x \;\leq\; & 4\alpha^2 x^2 - 2\alpha x - 2\alpha x^2 + x - 2\alpha x + 1 & \Longleftrightarrow \\
0 \;\leq\; & 2x\cdot(x\alpha - 1)(2\alpha - 1). &
\end{aligned}
$$

As the last condition is true for all $x \in \left[\frac{1}{\alpha}, 1\right)$, it follows that $h'(x) \geq 0$. So h is (weakly) increasing in $\left[\frac{1}{\alpha}, 1\right)$. Hence,

$$
h(x) \geq h\left(\frac{1}{\alpha}\right) = \mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{1}{\alpha}\right) - \mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{\frac{1}{\alpha}}{2\alpha\cdot\frac{1}{\alpha} - 1}\right) \;=\; \mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{1}{\alpha}\right) - \mathcal{D}\left(\frac{1}{\alpha}\middle\|\frac{1}{\alpha}\right) \;=\; 0
$$

for all $x \in \left[\frac{1}{\alpha}, 1\right)$. ⌟

To wrap-up the proof, consider the following cases.

- If $C_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa,\tau)\right) > \frac{1}{\alpha}$, then $\delta_{\alpha,\beta}(\kappa,\tau) < \frac{1}{\alpha}$ by Lemma 7.1. So $\delta_{\alpha,\beta}(\kappa,\tau) < C_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa,\tau)\right)$.

- If $C_{\alpha,\beta}(\gamma) \leq \frac{1}{\alpha}$, then $\mathcal{D}\left(\frac{1}{\alpha}\big\| x\right)$ is decreasing in the interval $\left[0, \frac{1}{\alpha}\right]$. Thus

$$\mathcal{D}\left(\frac{1}{\alpha}\bigg\| C_{\alpha,\beta}(\gamma)\right) \; < \; \mathcal{D}\left(\frac{1}{\alpha}\bigg\| \frac{\gamma}{2 \cdot \alpha \cdot \gamma - 1}\right) \; \leq \; \mathcal{D}\left(\frac{1}{\alpha}\bigg\| \gamma\right) \; \leq \; \mathcal{D}\left(\frac{1}{\alpha}\bigg\| \delta\right).$$

The first inequality follows from Claim 7.31, the second follows from Claim 7.32 (recall $\gamma \in \left(\frac{1}{\alpha}, 1\right)$ by Lemma 7.2) and the last inequality follows from Lemma 7.27. Since $C_{\alpha,\beta}(\gamma), \delta \leq \frac{1}{\alpha}$ (see Lemma 7.1), it follows that $C_{\alpha,\beta}(\gamma) < \delta$. $\qquad\square$

The next lemma follows from Lemma 7.30.

Lemma 7.33. *For all $\alpha > \beta > 1$, $c \geq 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$ it holds that $|H_g(\kappa, \tau^*(\kappa))| < 0$.*

Proof. By Lemma 7.9 it holds that $\tau^*(\kappa) \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$ and $(\partial_\tau g)(\kappa, \tau^*(\kappa)) = 0$. Thus, by Lemma 7.30, we have $C_{\alpha,\beta}\left(\gamma(\kappa, \tau^*(\kappa))\right) > \delta\left(\kappa, \tau^*(\kappa)\right)$. So $|H_g(\kappa, \tau^*(\kappa))| < 0$ by Corollary 7.29. $\qquad\square$

We can now proceed to the proof of Lemma 7.10.

Lemma 7.10. *For all simple $\alpha, \beta, c \geq 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$, the determinant of the Hessian of g at $(\kappa, \tau^*(\kappa))$ is negative, i.e., $|H_g(\kappa, \tau^*(\kappa))| < 0$.*

Proof. The lemma follows from Lemmas 7.26 and 7.33. $\qquad\square$

7.5 Partial Derivatives

In this section we calculate the partial derivatives of the function $g_{\alpha,\beta,c}(\kappa, \tau)$. For notational brevity, we use the following naming scheme for the partial derivatives:

$$(\partial_\kappa g_{\alpha,\beta,c})(\kappa_0, \tau_0) := \frac{\partial g_{\alpha,\beta,c}(\kappa, \tau)}{\partial \kappa}\bigg|_{(\kappa,\tau)=(\kappa_0,\tau_0)}$$

$$(\partial_\tau g_{\alpha,\beta,c})(\kappa_0, \tau_0) := \frac{\partial g_{\alpha,\beta,c}(\kappa, \tau)}{\partial \tau}\bigg|_{(\kappa,\tau)=(\kappa_0,\tau_0)}$$

$$(\partial^2_{\kappa,\tau} g_{\alpha,\beta,c})(\kappa_0, \tau_0) := \frac{\partial^2 g_{\alpha,\beta,c}(\kappa, \tau)}{\partial \kappa \partial \tau}\bigg|_{(\kappa,\tau)=(\kappa_0,\tau_0)}$$

$$(\partial^2_{\kappa,\kappa} g_{\alpha,\beta,c})(\kappa_0, \tau_0) := \frac{\partial^2 g_{\alpha,\beta,c}(\kappa, \tau)}{\partial \kappa^2}\bigg|_{(\kappa,\tau)=(\kappa_0,\tau_0)}$$

$$(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa_0, \tau_0) := \frac{\partial^2 g_{\alpha,\beta,c}(\kappa, \tau)}{\partial \tau^2}\bigg|_{(\kappa,\tau)=(\kappa_0,\tau_0)}$$

We sometimes omit the subscript (α, β, c) from $(\partial_\kappa g_{\alpha,\beta,c})$, $(\partial_\tau g_{\alpha,\beta,c})$, $(\partial^2_{\kappa,\kappa} g_{\alpha,\beta,c})$, $(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})$ and $(\partial^2_{\kappa,\tau} g_{\alpha,\beta,c})$.

Recall that $\mathcal{D}\left(a\|b\right) = a \ln \frac{a}{b} + (1-a) \ln \frac{1-a}{1-b}$ is the Kullback-Leibler divergence between two Bernoulli distributions with parameters a and b. In the next lemmas, we use algebraic properties of the KL divergence to calculate the partial derivatives of $g(\kappa, \tau)$.

It can be easily verified that the partial derivatives of $\mathcal{D}\left(a\|b\right)$ and $\mathcal{H}(x)$ are

$$\frac{\partial \mathcal{D}\left(a\|b\right)}{\partial b} = \frac{b-a}{b(1-b)} \tag{60}$$

$$\frac{\partial \mathcal{H}(x)}{\partial x} = \ln\left(\frac{1-x}{x}\right). \tag{61}$$

Moreover, the partial derivatives of $\gamma(\kappa, \tau)$ and $\delta(\kappa, \tau)$ are

$$\frac{\partial \gamma(\kappa, \tau)}{\partial \kappa} = \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{1}{\tau} \tag{62}$$

$$\frac{\partial \gamma(\kappa, \tau)}{\partial \tau} = -\left(1 - \frac{\beta}{\alpha}\right) \frac{\kappa}{\tau^2} = -\frac{\gamma(\kappa, \tau) - \frac{1}{\alpha}}{\tau} \tag{63}$$

$$\frac{\partial \delta(\kappa, \tau)}{\partial \kappa} = \frac{\beta}{\alpha \cdot (1 - \tau)} \tag{64}$$

$$\frac{\partial \delta(\kappa, \tau)}{\partial \tau} = \frac{(\beta \cdot \kappa - 1)}{\alpha \cdot (1 - \tau)^2} = \frac{\delta(\kappa, \tau) - \frac{1}{\alpha}}{1 - \tau} \tag{65}$$

For notational brevity, we sometimes omit the arguments (κ, τ) from $\gamma(\kappa, \tau)$ and $\delta(\kappa, \tau)$, and simply use γ and δ instead.

Lemma 7.5. *For all $\alpha, c \geq 1$ and $\beta > 1$ it holds that*

$$(\partial_\tau g_{\alpha,\beta,c})(\kappa, \tau) = -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma(\kappa, \tau)\right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta(\kappa, \tau)\right).$$

Proof. By (61) and standard derivation rules we have

$$(\partial_\tau g)(\kappa, \tau) = \frac{\partial}{\partial \tau}\left(\frac{\beta \kappa - \tau}{\alpha} \ln c - \tau \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\kappa, \tau)\right) - (1 - \tau) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\kappa, \tau)\right) + \mathcal{H}\left(\kappa\right)\right)$$

$$= -\frac{\ln(c)}{\alpha} - \mathcal{H}(\gamma) + \mathcal{H}(\delta) - \tau \cdot \ln\left(\frac{1 - \gamma}{\gamma}\right) \cdot \frac{\partial \gamma(\kappa, \tau)}{\partial \tau} - (1 - \tau) \cdot \ln\left(\frac{1 - \delta}{\delta}\right) \cdot \frac{\partial \delta(\kappa, \tau)}{\partial \tau}$$

$$= -\frac{\ln(c)}{\alpha} - \mathcal{H}(\gamma) + \mathcal{H}(\delta) + \left(\gamma - \frac{1}{\alpha}\right) \cdot \ln\left(\frac{1 - \gamma}{\gamma}\right) - \left(\delta - \frac{1}{\alpha}\right) \cdot \ln\left(\frac{1 - \delta}{\delta}\right)$$

$$= -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma\right) + \mathcal{H}\left(\frac{1}{\alpha}\right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta\right) - \mathcal{H}\left(\frac{1}{\alpha}\right)$$

$$= -\frac{\ln(c)}{\alpha} - \mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma\right) + \mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta\right)$$

where the third equality uses (63) and (65), and the forth follows from Lemma 7.18. $\qquad \square$

Lemma 7.34. *For all $\alpha, c \geq 1$ and $\beta > 1$ it holds that*

$$(\partial_\kappa g_{\alpha,\beta,c})(\kappa, \tau) = \frac{\beta}{\alpha} \cdot \ln(c) + (\beta - \alpha) \cdot \left(\mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma(\kappa, \tau)\right) + \ln\left(1 - \gamma(\kappa, \tau)\right)\right)$$

$$- \beta \cdot \left(\mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta(\kappa, \tau)\right) + \ln\left(1 - \delta(\kappa, \tau)\right)\right)$$

$$+ \alpha \cdot \left(\mathcal{D}\left(\frac{1}{\alpha} \middle\| \kappa\right) + \ln\left(1 - \kappa\right)\right).$$

The following identify is used in the proof of Lemma 7.34.

Lemma 7.35. *For all $x \in [0, 1]$ and $\alpha \geq 1$ it holds that*

$$\ln\left(\frac{1 - x}{x}\right) = \alpha \cdot \mathcal{D}\left(\frac{1}{\alpha} \middle\| x\right) - \ln\left(\frac{\frac{1}{\alpha}}{1 - \frac{1}{\alpha}}\right) - \alpha \cdot \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - x}\right)$$

Proof. By a sequence of algebraic manipulations we get

$$\frac{1}{\alpha} \cdot \ln\left(\frac{1 - x}{x}\right) = \frac{1}{\alpha} \cdot \ln\left(\frac{1}{x}\right) - \frac{1}{\alpha} \cdot \ln\left(\frac{1}{1 - x}\right)$$

$$= \frac{1}{\alpha} \cdot \ln\left(\frac{\frac{1}{\alpha}}{x}\right) - \frac{1}{\alpha} \cdot \ln\left(\frac{1}{\alpha}\right) - \frac{1}{\alpha} \cdot \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - x}\right) + \frac{1}{\alpha} \cdot \ln\left(1 - \frac{1}{\alpha}\right)$$

$$= \frac{1}{\alpha} \cdot \ln\left(\frac{\frac{1}{\alpha}}{x}\right) - \frac{1}{\alpha} \cdot \ln\left(\frac{1}{\alpha}\right) + \left(1 - \frac{1}{\alpha}\right) \cdot \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - x}\right) + \frac{1}{\alpha} \cdot \ln\left(1 - \frac{1}{\alpha}\right) - \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - x}\right)$$

$$= \mathcal{D}\left(\frac{1}{\alpha}\middle\|x\right) - \frac{1}{\alpha} \cdot \ln\left(\frac{1}{\alpha}\right) + \frac{1}{\alpha} \cdot \ln\left(1 - \frac{1}{\alpha}\right) - \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - x}\right)$$

$$= \mathcal{D}\left(\frac{1}{\alpha}\middle\|x\right) - \frac{1}{\alpha} \cdot \ln\left(\frac{\frac{1}{\alpha}}{1 - \frac{1}{\alpha}}\right) - \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - x}\right)$$

$\square$

Proof of Lemma 7.34. By (9) we have

$$(\partial_\kappa g)(\kappa, \tau) = \frac{\beta}{\alpha} \ln(c) + \underbrace{\frac{\partial}{\partial \kappa}\left(-\tau \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\kappa, \tau)\right)\right)}_{\Pi_1} + \underbrace{\frac{\partial}{\partial \kappa}\left(-(1 - \tau) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\kappa, \tau)\right)\right)}_{\Pi_2} + \underbrace{\frac{\partial}{\partial \kappa}\left(\mathcal{H}\left(\kappa\right)\right)}_{\Pi_3}.$$

It holds that

$$\Pi_1 = (-\tau) \cdot \ln\left(\frac{1 - \gamma(\kappa, \tau)}{\gamma(\kappa, \tau)}\right) \cdot \frac{\partial \gamma(\kappa, \tau)}{\partial \kappa}$$

$$= \frac{\beta - \alpha}{\alpha} \cdot \ln\left(\frac{1 - \gamma(\kappa, \tau)}{\gamma(\kappa, \tau)}\right)$$

$$= \frac{\beta - \alpha}{\alpha} \cdot \left(\alpha \cdot \mathcal{D}\left(\frac{1}{\alpha}\middle\|\gamma(\kappa, \tau)\right) - \ln\left(\frac{\frac{1}{\alpha}}{1 - \frac{1}{\alpha}}\right) - \alpha \cdot \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - \gamma(\kappa, \tau)}\right)\right),$$

where the second equality follows from (62) and the last equality follows from Lemma 7.35. By Lemma 7.35 and (64) we have

$$\Pi_2 = (-1 + \tau) \cdot \ln\left(\frac{1 - \delta(\kappa, \tau)}{\delta(\kappa, \tau)}\right) \cdot \frac{\partial \delta(\kappa, \tau)}{\partial \kappa}$$

$$= -\frac{\beta}{\alpha} \cdot \ln\left(\frac{1 - \delta(\kappa, \tau)}{\delta(\kappa, \tau)}\right)$$

$$= -\frac{\beta}{\alpha} \cdot \left(\alpha \cdot \mathcal{D}\left(\frac{1}{\alpha}\middle\|\delta(\kappa, \tau)\right) - \ln\left(\frac{\frac{1}{\alpha}}{1 - \frac{1}{\alpha}}\right) - \alpha \cdot \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - \delta(\kappa, \tau)}\right)\right),$$

and

$$\Pi_3 = \ln\left(\frac{1 - \kappa}{\kappa}\right) = \alpha \cdot \mathcal{D}\left(\frac{1}{\alpha}\middle\|\kappa\right) - \ln\left(\frac{\frac{1}{\alpha}}{1 - \frac{1}{\alpha}}\right) - \alpha \cdot \ln\left(\frac{1 - \frac{1}{\alpha}}{1 - \kappa}\right).$$

Overall, we get

$$(\partial_\kappa g)(\kappa, \tau) = \frac{\beta}{\alpha} \ln(c) + \Pi_1 + \Pi_2 + \Pi_3$$

$$= \frac{\beta}{\alpha} \cdot \ln(c) + (\beta - \alpha) \cdot \left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\gamma(\kappa, \tau)\right) + \ln\left(1 - \gamma(\kappa, \tau)\right)\right)$$

$$- \beta \cdot \left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\delta(\kappa, \tau)\right) + \ln\left(1 - \delta(\kappa, \tau)\right)\right)$$

$$+ \alpha \cdot \left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\kappa\right) + \ln\left(1 - \kappa\right)\right).$$

$\square$

Recall the functions $\Gamma_{\alpha,\beta}$ and $\Delta_{\alpha,\beta}$ from (42) and (43).

Lemma 7.36. *For all $\alpha, c \geq 1$ and $\beta > 1$ it holds that*

$$(\partial^2_{\kappa,\kappa} g_{\alpha,\beta,c})(\kappa, \tau) = \left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa, \tau) + \left(\frac{\beta}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa, \tau) - \frac{1}{\kappa \cdot (1 - \kappa)}.$$

Proof. Using Lemma 7.34 we have

$$(\partial^2_{\kappa,\kappa} g)(\kappa, \tau) = \underbrace{(\beta - \alpha) \cdot \frac{\partial}{\partial \kappa} \left(\mathcal{D}\left(\frac{1}{\alpha} \middle\| \gamma(\kappa, \tau)\right) + \ln\left(1 - \gamma(\kappa, \tau)\right) \right)}_{\Pi_1}$$

$$\underbrace{- \beta \cdot \frac{\partial}{\partial \kappa} \left(\mathcal{D}\left(\frac{1}{\alpha} \middle\| \delta(\kappa, \tau)\right) + \ln\left(1 - \delta(\kappa, \tau)\right) \right)}_{\Pi_2}$$

$$\underbrace{+ \alpha \cdot \frac{\partial}{\partial \kappa} \left(\mathcal{D}\left(\frac{1}{\alpha} \middle\| \kappa\right) + \ln\left(1 - \kappa\right) \right)}_{\Pi_3}.$$

By (60), (62) and (64) we obtain

$$\Pi_1 = (\beta - \alpha) \cdot \left(\frac{\gamma(\kappa, \tau) - \frac{1}{\alpha}}{\gamma(\kappa, \tau) \cdot (1 - \gamma(\kappa, \tau))} - \frac{1}{1 - \gamma(\kappa, \tau)} \right) \cdot \frac{\partial \gamma(\kappa, \tau)}{\partial \kappa}$$

$$= (\beta - \alpha) \cdot \left(\frac{\gamma(\kappa, \tau) - \frac{1}{\alpha}}{\gamma(\kappa, \tau) \cdot (1 - \gamma(\kappa, \tau))} - \frac{1}{1 - \gamma(\kappa, \tau)} \right) \cdot \left(1 - \frac{\beta}{\alpha}\right) \cdot \frac{1}{\tau}$$

$$= \left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \left(\frac{1}{\tau \cdot \gamma(\kappa, \tau) \cdot (1 - \gamma(\kappa, \tau))} \right)$$

$$= \left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa, \tau),$$

$$\Pi_2 = -\beta \cdot \left(\frac{\delta(\kappa, \tau) - \frac{1}{\alpha}}{\delta(\kappa, \tau) \cdot (1 - \delta(\kappa, \tau))} - \frac{1}{1 - \delta(\kappa, \tau)} \right) \cdot \frac{\partial \delta(\kappa, \tau)}{\partial \kappa}$$

$$= -\beta \cdot \left(\frac{\delta(\kappa, \tau) - \frac{1}{\alpha}}{\delta(\kappa, \tau) \cdot (1 - \delta(\kappa, \tau))} - \frac{1}{1 - \delta(\kappa, \tau)} \right) \cdot \left(\frac{\beta}{\alpha \cdot (1 - \tau)} \right)$$

$$= \left(\frac{\beta}{\alpha}\right)^2 \cdot \left(\frac{1}{(1 - \tau) \cdot \delta(\kappa, \tau) \cdot (1 - \delta(\kappa, \tau))} \right)$$

$$= \left(\frac{\beta}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa, \tau),$$

$$\Pi_3 = \alpha \cdot \left(\frac{\kappa - \frac{1}{\alpha}}{\kappa \cdot (1 - \kappa)} - \frac{1}{1 - \kappa} \right) = \frac{-1}{\kappa \cdot (1 - \kappa)}.$$

Finally, we have

$$(\partial^2_{\kappa,\kappa} g)(\kappa, \tau) = \Pi_1 + \Pi_2 + \Pi_3 = \left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa, \tau) + + \left(\frac{\beta}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa, \tau) - \frac{1}{\kappa \cdot (1 - \kappa)}.$$

$\square$

Lemma 7.37. *For all $\alpha, c \geq 1$ and $\beta > 1$ it holds that*

$$(\partial^2_{\kappa,\tau} g_{\alpha,\beta,c})(\kappa,\tau) = -\left(1 - \frac{\beta}{\alpha}\right) \cdot \left(\gamma(\kappa,\tau) - \frac{1}{\alpha}\right) \cdot \Gamma_{\alpha,\beta}(\kappa,\tau) + \frac{\beta}{\alpha} \cdot \left(\delta(\kappa,\tau) - \frac{1}{\alpha}\right) \cdot \Delta_{\alpha,\beta}(\kappa,\tau).$$

Proof. By lemma 7.34 we get

$$(\partial^2_{\kappa,\tau} g)(\kappa,\tau) = \underbrace{(\beta - \alpha) \cdot \frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\gamma(\kappa,\tau)\right) + \ln\left(1 - \gamma(\kappa,\tau)\right)\right)}_{\Pi_1}$$

$$\underbrace{-\beta \cdot \frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\delta(\kappa,\tau)\right) + \ln\left(1 - \delta(\kappa,\tau)\right)\right)}_{\Pi_2}.$$

By (60), (63) and (65) we have

$$\Pi_1 = (\beta - \alpha) \cdot \left(\frac{\gamma(\kappa,\tau) - \frac{1}{\alpha}}{\gamma(\kappa,\tau) \cdot \left(1 - \gamma(\kappa,\tau)\right)} - \frac{1}{1 - \gamma(\kappa,\tau)}\right) \cdot \frac{\partial \gamma(\kappa,\tau)}{\partial \tau}$$

$$= (\beta - \alpha) \cdot \left(\frac{-1}{\alpha \cdot \gamma(\kappa,\tau) \cdot \left(1 - \gamma(\kappa,\tau)\right)}\right) \cdot \left(\frac{-\gamma(\kappa,\tau) + \frac{1}{\alpha}}{\tau}\right)$$

$$= -\left(1 - \frac{\beta}{\alpha}\right) \cdot \left(\gamma(\kappa,\tau) - \frac{1}{\alpha}\right) \cdot \Gamma_{\alpha,\beta}(\kappa,\tau)$$

and

$$\Pi_2 = -\beta \cdot \left(\frac{\delta(\kappa,\tau) - \frac{1}{\alpha}}{\delta(\kappa,\tau) \cdot \left(1 - \delta(\kappa,\tau)\right)} - \frac{1}{1 - \delta(\kappa,\tau)}\right) \cdot \frac{\partial \delta(\kappa,\tau)}{\partial \tau}$$

$$= -\beta \cdot \left(\frac{-1}{\alpha \cdot \delta(\kappa,\tau) \cdot \left(1 - \delta(\kappa,\tau)\right)}\right) \cdot \left(\frac{\delta(\kappa,\tau) - \frac{1}{\alpha}}{1 - \tau}\right)$$

$$= \frac{\beta}{\alpha} \cdot \left(\delta(\kappa,\tau) - \frac{1}{\alpha}\right) \cdot \Delta_{\alpha,\beta}(\kappa,\tau).$$

So

$$(\partial^2_{\kappa,\tau} g)(\kappa,\tau) = \Pi_1 + \Pi_2 = -\left(1 - \frac{\beta}{\alpha}\right) \cdot \left(\gamma(\kappa,\tau) - \frac{1}{\alpha}\right) \cdot \Gamma_{\alpha,\beta}(\kappa,\tau) + \frac{\beta}{\alpha} \cdot \left(\delta(\kappa,\tau) - \frac{1}{\alpha}\right) \cdot \Delta_{\alpha,\beta}(\kappa,\tau).$$

$\square$

Lemma 7.6. *For all $\alpha, c \geq 1$ and $\beta > 1$, the second order partial derivative of $g(\kappa,\tau)$ by τ, i.e., $(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa,\tau)$ is given by*

$$(\partial^2_{\tau,\tau} g_{\alpha,\beta,c})(\kappa,\tau) = \left(\gamma(\kappa,\tau) - \frac{1}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa,\tau) + \left(\delta(\kappa,\tau) - \frac{1}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa,\tau).$$

Proof. By Lemma 7.5 we have

$$(\partial^2_{\tau,\tau} g) = \frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\delta(\kappa,\tau)\right)\right) - \frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\gamma(\kappa,\tau)\right)\right).$$

By (60) and (65) we have

$$\frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\delta(\kappa,\tau)\right)\right) = \frac{\delta - \frac{1}{\alpha}}{\delta \cdot (1 - \delta)} \cdot \frac{\partial \delta(\kappa,\tau)}{\partial \tau} = \frac{\delta - \frac{1}{\alpha}}{\delta \cdot (1 - \delta)} \cdot \frac{\delta - \frac{1}{\alpha}}{1 - \tau} = \left(\delta - \frac{1}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa,\tau).$$

54

Similarly, by (60) and (63) we have

$$\frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\gamma(\kappa,\tau)\right)\right) = \frac{\gamma - \frac{1}{\alpha}}{\gamma \cdot \left(1 - \gamma\right)} \cdot \frac{\partial \gamma(\kappa,\tau)}{\partial \tau} = \frac{\gamma - \frac{1}{\alpha}}{\gamma \cdot (1-\gamma)} \cdot \frac{-\left(\gamma - \frac{1}{\alpha}\right)}{\tau} = -\left(\gamma - \frac{1}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa,\tau).$$

Together

$$\begin{aligned}
(\partial^2_{\tau,\tau} g) &= \frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\delta(\kappa,\tau)\right)\right) - \frac{\partial}{\partial \tau}\left(\mathcal{D}\left(\frac{1}{\alpha}\middle\|\gamma(\kappa,\tau)\right)\right) \\
&= \left(\delta - \frac{1}{\alpha}\right)^2 \cdot \Delta_{\alpha,\beta}(\kappa,\tau) + \left(\gamma - \frac{1}{\alpha}\right)^2 \cdot \Gamma_{\alpha,\beta}(\kappa,\tau).
\end{aligned}$$

$\square$

7.6 A Formula for the Determinant of the Hessian

In this section we prove Lemma 7.20, that is, we provide a formula for $|H_g(\kappa,\tau)|$.

Lemma 7.20. *Let $\alpha \geq 1$, $\beta > 1$ such that $\alpha \neq \beta$, $\kappa \in \left(0, \frac{1}{\beta}\right)$ and $\tau \in (M_{\alpha,\beta}(\kappa), \beta \cdot \kappa)$. Then*

$$\begin{aligned}
|H_g(\kappa,\tau)| = \frac{\Gamma_{\alpha,\beta}(\kappa,\tau) \cdot \Delta_{\alpha,\beta}(\kappa,\tau)}{\alpha^2 \cdot (1-\kappa)} &\cdot \frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma_{\alpha,\beta}(\kappa,\tau) - \delta_{\alpha,\beta}(\kappa,\tau))}{\gamma_{\alpha,\beta}(\kappa,\tau) - \frac{1}{\alpha}} \cdot \\
&\left(A_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa,\tau)\right) + \delta_{\alpha,\beta}(\kappa,\tau) \cdot B_{\alpha,\beta}\left(\gamma_{\alpha,\beta}(\kappa,\tau)\right)\right),
\end{aligned}$$

where

$$\begin{aligned}
A_{\alpha,\beta}(x) &:= -2 + x\left(1 + \alpha + \beta\right) - \alpha \cdot \beta \cdot x^2 \qquad and \\
B_{\alpha,\beta}(x) &:= x \cdot \alpha \cdot (\beta - 2) + 1 + \alpha - \beta.
\end{aligned}$$

Proof. By Lemmas 7.6 and 7.36 we have

$$\begin{aligned}
&(\partial^2_{\kappa,\kappa} g)(\kappa,\tau) \cdot (\partial^2_{\tau,\tau} g)(\kappa,\tau) \\
&= \left(\left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \Gamma + \left(\frac{\beta}{\alpha}\right)^2 \cdot \Delta - \frac{1}{\kappa \cdot (1-\kappa)}\right) \cdot \left(\left(\gamma - \frac{1}{\alpha}\right)^2 \cdot \Gamma + \left(\delta - \frac{1}{\alpha}\right)^2 \cdot \Delta\right) \\
&= \left(\gamma - \frac{1}{\alpha}\right)^2 \cdot \left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \Gamma^2 + \left(\delta - \frac{1}{\alpha}\right)^2 \cdot \left(\frac{\beta}{\alpha}\right)^2 \cdot \Delta^2 - \frac{(\partial^2_{\tau,\tau} g)(\kappa,\tau)}{\kappa(1-\kappa)} \\
&\quad + \Delta \cdot \Gamma \left(\left(\gamma - \frac{1}{\alpha}\right)^2 \cdot \left(\frac{\beta}{\alpha}\right)^2 + \left(\delta - \frac{1}{\alpha}\right)^2 \cdot \left(1 - \frac{\beta}{\alpha}\right)^2\right).
\end{aligned}$$ (66)

Similarly, by Lemma 7.37 it holds that

$$\begin{aligned}
\left((\partial^2_{\kappa,\tau} g)(\kappa,\tau)\right)^2 &= \left(-\left(1 - \frac{\beta}{\alpha}\right) \cdot \left(\gamma - \frac{1}{\alpha}\right) \cdot \Gamma + \frac{\beta}{\alpha} \cdot \left(\delta - \frac{1}{\alpha}\right) \cdot \Delta\right)^2 \\
&= \left(\gamma - \frac{1}{\alpha}\right)^2 \cdot \left(1 - \frac{\beta}{\alpha}\right)^2 \cdot \Gamma^2 + \left(\delta - \frac{1}{\alpha}\right)^2 \cdot \left(\frac{\beta}{\alpha}\right)^2 \cdot \Delta^2 \\
&\quad - 2 \cdot \left(\gamma - \frac{1}{\alpha}\right) \cdot \left(1 - \frac{\beta}{\alpha}\right) \cdot \left(\delta - \frac{1}{\alpha}\right) \cdot \left(\frac{\beta}{\alpha}\right) \cdot \Gamma \cdot \Delta.
\end{aligned}$$ (67)

By (66) and (67) we have

$$
\begin{aligned}
|H_g(\kappa,\tau)| &= (\partial^2_{\kappa,\kappa}g)(\kappa,\tau)\cdot(\partial^2_{\tau,\tau}g)(\kappa,\tau) - \left((\partial^2_{\kappa,\tau}g)(\kappa,\tau)\right)^2 \\
&= -\frac{(\partial^2_{\tau,\tau}g)(\kappa,\tau)}{\kappa(1-\kappa)} + \Delta\cdot\Gamma\left(\left(\gamma-\frac{1}{\alpha}\right)^2\cdot\left(\frac{\beta}{\alpha}\right)^2 + \left(\delta-\frac{1}{\alpha}\right)^2\cdot\left(1-\frac{\beta}{\alpha}\right)^2\right) \\
&\quad + 2\cdot\left(\gamma-\frac{1}{\alpha}\right)\cdot\left(1-\frac{\beta}{\alpha}\right)\cdot\left(\delta-\frac{1}{\alpha}\right)\cdot\left(\frac{\beta}{\alpha}\right)\cdot\Gamma\cdot\Delta \\
&= -\frac{(\partial^2_{\tau,\tau}g)(\kappa,\tau)}{\kappa(1-\kappa)} + \Delta\cdot\Gamma\left(\left(\gamma-\frac{1}{\alpha}\right)\cdot\left(\frac{\beta}{\alpha}\right) + \left(\delta-\frac{1}{\alpha}\right)\cdot\left(1-\frac{\beta}{\alpha}\right)\right)^2 \\
&= \frac{\Delta\cdot\Gamma}{1-\kappa}\cdot\left(-\frac{(\partial^2_{\tau,\tau}g)}{\kappa\cdot\Delta\cdot\Gamma} + (1-\kappa)\cdot\left(\left(\gamma-\frac{1}{\alpha}\right)\cdot\left(\frac{\beta}{\alpha}\right) + \left(\delta-\frac{1}{\alpha}\right)\cdot\left(1-\frac{\beta}{\alpha}\right)\right)^2\right).
\end{aligned}
\tag{68}
$$

We define

$$
\psi_{\alpha,\beta}(\kappa,\tau) = \frac{\beta}{\alpha}\cdot\left(\gamma_{\alpha,\beta}(\kappa,\tau)-\frac{1}{\alpha}\right) + \left(1-\frac{\beta}{\alpha}\right)\cdot\left(\delta_{\alpha,\beta}(\kappa,\tau)-\frac{1}{\alpha}\right).
\tag{69}
$$

As before, we use the shorthand $\psi = \psi_{\alpha,\beta}(\kappa,\tau)$. Thus,

$$
|H_g(\kappa,\tau)| = \frac{\Delta\cdot\Gamma}{1-\kappa}\cdot\left(-\frac{(\partial^2_{\tau,\tau}g)}{\kappa\cdot\Delta\cdot\Gamma} + (1-\kappa)\cdot\psi^2\right).
\tag{70}
$$

We use the following algebraic identity to simplify (70).

Claim 7.38. $\kappa\cdot\psi = \delta\left(\gamma-\frac{1}{\alpha}\right)$.

Proof. The statement of the claim immediately follows from the following equation.

$$
\begin{aligned}
\kappa\cdot\psi - \delta\left(\gamma-\frac{1}{\alpha}\right) &= \kappa\cdot\left(1-\frac{\beta}{\alpha}\right)\cdot\left(\delta-\frac{1}{\alpha}\right) + \kappa\cdot\frac{\beta}{\alpha}\left(\gamma-\frac{1}{\alpha}\right) - \delta\cdot\left(\gamma-\frac{1}{\alpha}\right) \\
&= \left(\delta-\frac{1}{\alpha}\right)\cdot\left(\gamma-\frac{1}{\alpha}\right)\cdot\tau + \left(\gamma-\frac{1}{\alpha}\right)\cdot\left(\kappa\cdot\frac{\beta}{\alpha}-\delta\right) \\
&= \left(\gamma-\frac{1}{\alpha}\right)\cdot\left(\left(\delta-\frac{1}{\alpha}\right)\cdot\tau + \frac{\beta}{\alpha}\cdot\kappa-\delta\right) \\
&= \left(\gamma-\frac{1}{\alpha}\right)\cdot\left(-\delta\cdot(1-\tau)+\frac{\beta}{\alpha}\kappa-\frac{\tau}{\alpha}\right) = 0.
\end{aligned}
$$

The first equality follows from an expansion of ψ by (69). The second equality uses the identity $\left(\gamma-\frac{1}{\alpha}\right)\cdot\tau = \left(1-\frac{\beta}{\alpha}\right)\cdot\kappa$ by (8). The last equality holds as $\delta\cdot(1-\tau) = \frac{\beta}{\alpha}\cdot\kappa-\frac{\tau}{\alpha}$. ⌟

By Lemma 7.6 and the definition of Δ (43) and Γ (42) it holds that

$$
\begin{aligned}
\frac{(\partial^2_{\tau,\tau}g)}{\kappa\cdot\Delta\cdot\Gamma} &= \frac{1}{\kappa}\cdot\left(\gamma-\frac{1}{\alpha}\right)^2\cdot(1-\tau)\cdot\delta\cdot(1-\delta) + \frac{1}{\kappa}\cdot\left(\delta-\frac{1}{\alpha}\right)^2\cdot\tau\cdot(1-\gamma)\cdot\gamma \\
&= \frac{1}{\kappa}\cdot\left(\gamma-\frac{1}{\alpha}\right)^2\cdot\delta\cdot(1-\delta) + \frac{\tau}{\kappa}\left(\left(\delta-\frac{1}{\alpha}\right)^2\cdot(1-\gamma)\cdot\gamma - \left(\gamma-\frac{1}{\alpha}\right)^2\cdot\delta\cdot(1-\delta)\right) \\
&= \psi\cdot\left(\gamma-\frac{1}{\alpha}\right)\cdot(1-\delta) + \Pi_1,
\end{aligned}
\tag{71}
$$

where the last equality uses $\frac{1}{\kappa}\cdot\delta\left(\gamma-\frac{1}{\alpha}\right) = \psi$ by Claim 7.38 and

$$
\Pi_1 := \frac{\tau}{\kappa}\left(\left(\delta-\frac{1}{\alpha}\right)^2\cdot(1-\gamma)\cdot\gamma - \left(\gamma-\frac{1}{\alpha}\right)^2\cdot\delta\cdot(1-\delta)\right).
$$

Furthermore,

$$(1 - \kappa) \cdot \psi^2 = \psi^2 - \kappa \cdot \psi^2 = \psi^2 - \psi \cdot \delta \cdot \left(\gamma - \frac{1}{\alpha}\right), \tag{72}$$

where the second equality follows from Claim 7.38. By (70), (71) and (72) we have

$$
\begin{aligned}
|H_g(\kappa, \tau)| &= \frac{\Delta \cdot \Gamma}{1 - \kappa} \cdot \left(-\psi\left(\gamma - \frac{1}{\alpha}\right) \cdot (1 - \delta) - \Pi_1 + \psi^2 - \psi \cdot \delta \cdot \left(\gamma - \frac{1}{\alpha}\right)\right) \\
&= \frac{\Delta \cdot \Gamma}{1 - \kappa} \cdot \left(-\psi\left(\gamma - \frac{1}{\alpha}\right) + \psi^2 - \Pi_1\right) \\
&= \frac{\Delta \cdot \Gamma}{1 - \kappa} \cdot (\Pi_2 - \Pi_1),
\end{aligned}
\tag{73}
$$

where

$$\Pi_2 := -\psi\left(\gamma - \frac{1}{\alpha}\right) + \psi^2.$$

Observe that

$$
\begin{aligned}
&\gamma \cdot (1 - \gamma) \cdot \left(\delta - \frac{1}{\alpha}\right)^2 - \delta \cdot (1 - \delta) \cdot \left(\gamma - \frac{1}{\alpha}\right)^2 \\
&= \gamma \cdot (1 - \gamma) \cdot \left(\delta^2 - \frac{2}{\alpha}\delta + \frac{1}{\alpha^2}\right) - \delta \cdot (1 - \delta) \cdot \left(\gamma^2 - \frac{2}{\alpha}\gamma + \frac{1}{\alpha^2}\right) \\
&= \delta^2 \gamma - \frac{2}{\alpha}\delta\gamma + \frac{\gamma}{\alpha^2} - \delta^2 \cdot \gamma^2 + \frac{2}{\alpha}\delta \cdot \gamma^2 - \frac{\gamma^2}{\alpha^2} \\
&\quad - \gamma^2\delta + \frac{2}{\alpha}\gamma\delta - \frac{\delta}{\alpha^2} + \gamma^2 \cdot \delta^2 - \frac{2}{\alpha}\gamma \cdot \delta^2 + \frac{\delta^2}{\alpha^2} \\
&= (\delta - \gamma) \cdot \gamma\delta + (\gamma - \delta) \cdot \frac{1}{\alpha^2} + (\gamma - \delta) \cdot \frac{2}{\alpha}\gamma\delta - \frac{1}{\alpha^2}(\gamma + \delta)(\gamma - \delta) \\
&= (\gamma - \delta) \cdot \frac{1}{\alpha^2}\left(-\alpha^2 \cdot \gamma\delta + 1 + 2\alpha \cdot \gamma\delta - \gamma - \delta\right)
\end{aligned}
$$

and $\frac{\tau}{\kappa} = \frac{1 - \frac{\beta}{\alpha}}{\gamma - \frac{1}{\alpha}}$ by (8). Thus

$$
\begin{aligned}
\Pi_1 &= \frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot \left(-\alpha^2 \cdot \gamma\delta + 1 + 2\alpha \cdot \gamma\delta - \gamma - \delta\right) \\
&= \frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot \left(\delta \cdot \left(-\alpha^2 \cdot \gamma + 2\alpha\gamma - 1\right) + 1 - \gamma\right)
\end{aligned}
\tag{74}
$$

Additionally,

$$
\begin{aligned}
\Pi_2 &= -\psi\left(\gamma - \frac{1}{\alpha}\right) + \psi^2 \\
&= \psi\left(-\left(\gamma - \frac{1}{\alpha}\right) + \left(\delta - \frac{1}{\alpha}\right) \cdot \left(1 - \frac{\beta}{\alpha}\right) + \frac{\beta}{\alpha} \cdot \left(\gamma - \frac{1}{\alpha}\right)\right) \\
&= \psi\left(1 - \frac{\beta}{\alpha}\right) \cdot (\delta - \gamma).
\end{aligned}
$$

By further expanding the expression for ψ and dividing and multiplying by $\alpha^2 \cdot \left(\gamma - \frac{1}{\alpha}\right)$ we get

$$
\begin{aligned}
\Pi_2 \;=\; & -\frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot \left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2 \left(\left(\gamma - \frac{1}{\alpha}\right) \cdot \frac{\beta}{\alpha} + \left(\delta - \frac{1}{\alpha}\right) \cdot \left(1 - \frac{\beta}{\alpha}\right)\right) \\
\;=\; & -\frac{\left(1 - \frac{\beta}{\alpha}\right)(\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot \left(\left(\gamma - \frac{1}{\alpha}\right) \cdot \left(\left(\gamma - \frac{1}{\alpha}\right) \alpha\beta - (\alpha - \beta)\right) + \delta\alpha \left(\gamma - \frac{1}{\alpha}\right)(\alpha - \beta)\right) \\
\;=\; & -\frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot \left(\left(\gamma - \frac{1}{\alpha}\right) \cdot (\alpha \cdot \beta \cdot \gamma - \alpha) + \delta \cdot (\gamma\alpha \cdot (\alpha - \beta) - \alpha + \beta)\right) \\
\;=\; & -\frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot \left(\alpha\beta\gamma^2 - \gamma(\beta + \alpha) + 1 + \delta \cdot (\gamma\alpha \cdot (\alpha - \beta) - \alpha + \beta)\right).
\end{aligned}
\tag{75}
$$

Define $\xi = \frac{\Delta \cdot \Gamma}{1 - \kappa} \cdot \frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2}$. By (73), (74) and (75) we have

$$
\begin{aligned}
|H_g(\kappa, \tau)| \;=\; & \xi \cdot \Bigg(-\left(\alpha\beta\gamma^2 - \gamma(\beta + \alpha) + 1 + \delta \cdot (\gamma\alpha \cdot (\alpha - \beta) - \alpha + \beta)\right) \\
& \qquad - \delta \cdot \left(-\alpha^2 \cdot \gamma + 2\alpha\gamma - 1\right) - 1 + \gamma \Bigg) \\
\;=\; & \xi \cdot \left(-\alpha\beta\gamma^2 + \gamma(\alpha + \beta + 1) - 2 + \delta \left(\gamma\alpha \cdot (\beta - 2) + \alpha - \beta + 1\right)\right) \\
\;=\; & \xi \cdot (A_{\alpha,\beta}(\gamma) + \delta \cdot B_{\alpha,\beta}(\gamma)) \\
\;=\; & \frac{\Delta \cdot \Gamma}{1 - \kappa} \cdot \frac{\left(1 - \frac{\beta}{\alpha}\right) \cdot (\gamma - \delta)}{\left(\gamma - \frac{1}{\alpha}\right) \cdot \alpha^2} \cdot (A_{\alpha,\beta}(\gamma) + \delta \cdot B_{\alpha,\beta}(\gamma)),
\end{aligned}
$$

which completes the proof of the lemma. $\qquad\square$

8 Better than Brute Force

In this section we prove Theorem 2.18. We first use Lemma 2.13 to show that $\mathtt{amls}(\alpha, c, \beta) < \mathtt{brute}(\beta)$ for all $\alpha, c \geq 1$ and $\beta > 1$. Broadly speaking, the brute force algorithm presented in [26] works as follows. The algorithm iterates over k from 0 to $\frac{n}{\beta}$ (where n is the size of the universe U), and the analysis focuses on the iteration in which k is the minimum cardinality of a set in $\mathcal{F}$. For each value of k the algorithm samples random subsets of the universe U of size $\beta \cdot k$ and checks if each set is in the set system $\mathcal{F}$. The number of sampled sets is selected to be sufficiently large to ensure a constant success probability. It can be shown (though not formally used by our proofs) that the number of sampled sets of size βk should be $\approx \exp\left(n \cdot \xi_\beta\left(\frac{k}{n}\right)\right)$ where ξ_β is defined by

$$
\xi_\beta(\kappa) := -\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) \cdot \kappa + \mathcal{H}(\kappa).
\tag{76}
$$

for all $0 \leq \kappa \leq 1$. It can also be easily verified that

$$
\xi_\beta(\kappa) = g_{\alpha,\beta,c}(\kappa, \beta \cdot \kappa)
\tag{77}
$$

for all $\alpha, \beta, c \geq 1$ and $0 \leq \kappa \leq \frac{1}{\beta}$. We use the following property of ξ_β.

Lemma 8.1. *For all $\beta > 1$ it holds that $\max_{0 \leq \kappa \leq \frac{1}{\beta}} \xi_\beta(\kappa) = \ln\left(\mathtt{brute}(\beta)\right)$.*

Proof. The expression $-\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) \cdot \kappa$ is a linear function of κ, and $\mathcal{H}(\kappa)$ is a concave function of κ. So ξ_β is a concave function. Also,

$$
\xi_\beta(0) = -\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) \cdot 0 + \mathcal{H}(0) = 0
$$

and

$$\xi_\beta\left(\frac{1}{\beta}\right) = -\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) \cdot \frac{1}{\beta} + \mathcal{H}\left(\frac{1}{\beta}\right) = 0.$$

Thus, ξ_β has a maximum in $\left(0, \frac{1}{\beta}\right)$. Let $\kappa^* \in \left(0, \frac{1}{\beta}\right)$ be such a maximum and let ξ'_β be the derivative of ξ_β. Then $\xi'_\beta(\kappa^*) = 0$.

Using basic differentiation rules we have

$$\xi'_\beta(\kappa) = -\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) + \ln\left(\frac{1-\kappa}{\kappa}\right).$$

It follows that

$$\ln(1-\kappa^*) - \ln(\kappa^*) = \ln\left(\frac{1-\kappa^*}{\kappa^*}\right) = \beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right). \tag{78}$$

Hence,

$$\begin{aligned}
\max_{0 \leq \kappa \leq \frac{1}{\beta}} \xi_\beta(\kappa) &= \xi_\beta(\kappa^*) \\
&= -\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) \cdot \kappa^* + \mathcal{H}(\kappa^*) \\
&= -\kappa^* \cdot (\ln(1-\kappa^*) - \ln(\kappa^*)) - \kappa^* \cdot \ln(\kappa^*) - (1-\kappa^*)\ln(1-\kappa^*) \\
&= -\ln(1-\kappa^*) \\
&= \ln\left(\frac{1-\kappa^*+\kappa^*}{1-\kappa^*}\right) \\
&= \ln\left(1 + \frac{\kappa^*}{1-\kappa^*}\right) \\
&= \ln\left(1 + \exp\left(-\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right)\right)\right).
\end{aligned}$$

The third equality follows from (78) and the definition of $\mathcal{H}$. The forth, fifth and sixth equalities are simple re-arrangements of the terms. The seventh equality uses (78) once more. Recall that $\mathtt{brute}(\beta) = 1 + \exp\left(-\beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right)\right)$, so $\max_{0 \leq \kappa \leq \frac{1}{\beta}} \xi_\beta(\kappa) = \ln(\mathtt{brute}(\beta))$. $\qquad\square$

We use Lemma 8.1 in the proof of the following lemma.

Lemma 8.2. *Let $\alpha, c \geq 1$ and $\beta > 1$. Then $\mathtt{amls}(\alpha, c, \beta) < \mathtt{brute}(\beta)$.*

Proof. There exists $\kappa \in \left[0, \frac{1}{\beta}\right]$ such that $\exp\left(g^*_{\alpha,\beta,c}(\kappa)\right) = \mathtt{amls}(\alpha, c, \beta)$ (see (13)). Consider the following cases.

- If $\kappa = 0$ or $\kappa = \frac{1}{\beta}$, it can be easily verified that $g^*_{\alpha,\beta,c}(\kappa) = 0$. So

$$\mathtt{amls}(\alpha, c, \beta) = \exp\left(g^*_{\alpha,\beta,c}(\kappa)\right) = 1 < \mathtt{brute}(\beta).$$

- If $0 < \kappa < \frac{1}{\beta}$, then, by Lemma 2.13, we have

$$\begin{aligned}
\mathtt{amls}(\alpha, c, \beta) &= \exp\left(g^*_{\alpha,\beta,c}(\kappa)\right) \\
&< \exp\left(g_{\alpha,\beta,c}(\kappa, \beta \cdot \kappa)\right) \\
&= \exp\left(\xi_\beta(\kappa)\right) \\
&\leq \exp\left(\max_{0 \leq \kappa' \leq \frac{1}{\beta}} \xi_\beta(\kappa')\right) \\
&= \exp\left(\ln(\mathtt{brute}(\beta))\right) \\
&= \mathtt{brute}(\beta).
\end{aligned}$$

The second equality follows from (77) and the forth equality follows from Lemma 8.1. $\qquad\square$

The next lemma provides the missing ingredient towards the proof of Theorem 2.18.

Lemma 8.3. *Let $\alpha \geq 1$ and $\beta > 1$. Then $\lim_{c \to \infty} \mathtt{amls}(\alpha, c, \beta) = \mathtt{brute}(\beta)$.*

Proof. By Lemma 8.1 there is some $0 \leq \kappa \leq \frac{1}{\beta}$ such that $\xi_\beta(\kappa) = \ln(\mathtt{brute}(\beta))$. We define $L :=$ $\liminf_{c \to \infty} g^*_{\alpha, c, \beta}(\kappa)$. By Lemma 8.2, for every $c \geq 1$, it holds that $g^*_{\alpha, c, \beta}(\kappa) \leq \ln(\mathtt{amls}(\alpha, c, \beta)) \leq$ $\ln(\mathtt{brute}(\beta)))$. So $L \leq \ln(\mathtt{brute}(\beta)))$.

There exists a strictly increasing sequence $(c_i)_{i=1}^\infty$ such that $L = \lim_{i \to \infty} g_{\alpha, c_i, \beta}(\kappa)$. For every $i \in \mathbb{N}$ define $\tau_i := \mathrm{argmin}_{M^*_{\alpha, \beta}(\kappa) \leq \tau \leq \beta\kappa} g_{\alpha, \beta, c_i}(\kappa, \tau)$. Recall that the Bolzano-Weierstrass Theorem asserts that every bounded sequence has a convergent subsequence (see, e.g., in [47, Theorem 2.4.1]). By the Bolzano-Weierstrass Theorem, as $M^*_{\alpha, \beta}(\kappa) \leq \tau_i \leq \beta\kappa$ for all $i \in \mathbb{N}$, there exists a monotone sequence of indices $(i_j)_{j=0}^\infty$ such that $\tau_{i_j} \xrightarrow[j \to \infty]{} \tau^*$ for some $M^*_{\alpha, \beta}(\kappa) \leq \tau^* \leq \beta\kappa$. Thus, we have

$$
\begin{aligned}
L &= \lim_{j \to \infty} g^*_{\alpha, c_{i_j}, \beta}(\kappa) \\
&= \lim_{j \to \infty} g_{\alpha, c_{i_j}, \beta}(\kappa, \tau_{i_j}) \\
&= \lim_{j \to \infty} \left(\frac{\beta\kappa - \tau_{i_j}}{\alpha} \ln c_{i_j} - \tau_{i_j} \cdot \mathcal{H}\left(\gamma_{\alpha, \beta}(\kappa, \tau_{i_j})\right) - (1 - \tau_{i_j}) \cdot \mathcal{H}\left(\delta_{\alpha, \beta}(\kappa, \tau_{i_j})\right) + \mathcal{H}(\kappa) \right),
\end{aligned}
\tag{79}
$$

where second equality uses the definition of $g^*_{\alpha, \beta, c}$ (12) and the third equality uses the definition of $g_{\alpha, \beta, c}$ (9).

Assume towards contradiction that $\tau^* < \beta\kappa$. Then $\frac{\beta\kappa - \tau_{i_j}}{\alpha} \ln c_{i_j} \xrightarrow[j \to \infty]{} \infty$. Also, since the entropy function and τ_{i_j} are both bounded, it follows that the expression $-(1 - \tau_{i_j}) \cdot \mathcal{H}\left(\delta_{\alpha, \beta}(\kappa, \tau_{i_j})\right) + \mathcal{H}(\kappa)$ is bounded. Thus, by (79), we have $L = \infty$, contradicting $L \leq \ln(\mathtt{brute}(\beta))$. So $\tau^* = \beta\kappa$.

Using $\tau^* = \beta \cdot \kappa$ and $\tau_{i_j} \xrightarrow[j \to \infty]{} \tau^*$, we can simplify the limit in (79) and obtain

$$
\begin{aligned}
L &= \lim_{j \to \infty} \left(\frac{\beta\kappa - \tau_{i_j}}{\alpha} \ln c_{i_j} - \tau_{i_j} \cdot \mathcal{H}\left(\gamma_{\alpha, \beta}(\kappa, \tau_{i_j})\right) - (1 - \tau_{i_j}) \cdot \mathcal{H}\left(\delta_{\alpha, \beta}(\kappa, \tau_{i_j})\right) + \mathcal{H}(\kappa) \right). \\
&\geq -\beta \cdot \kappa \cdot \mathcal{H}\left(\gamma_{\alpha, \beta}(\kappa, \beta \cdot \kappa)\right) - (1 - \beta \cdot \kappa) \cdot \mathcal{H}\left(\delta_{\alpha, \beta}(\kappa, \beta \cdot \kappa)\right) + \mathcal{H}(\kappa) \\
&= -\kappa \cdot \beta \cdot \mathcal{H}\left(\frac{1}{\beta}\right) - (1 - \beta \cdot \kappa) \cdot \mathcal{H}(0) + \mathcal{H}(\kappa) \\
&= \xi_\beta(\kappa) = \ln(\mathtt{brute}(\beta)).
\end{aligned}
$$

Therefore,

$$
\liminf_{c \to \infty} \ln(\mathtt{amls}(\alpha, c, \beta)) \geq \liminf_{c \to \infty} g^*_{\alpha, \beta, c}(\kappa) \geq \ln(\mathtt{brute}(\beta)),
$$

where the first inequality follows from (13). Since ln is continuous the last inequality implies

$$
\liminf_{c \to \infty} \mathtt{amls}(\alpha, c, \beta) \geq \mathtt{brute}(\beta),
\tag{80}
$$

Also, by Lemma 8.2, we have

$$
\limsup_{c \to \infty} \mathtt{amls}(\alpha, c, \beta) \leq \limsup_{c \to \infty} \mathtt{brute}(\beta) = \mathtt{brute}(\beta).
\tag{81}
$$

Combining (80) and (81) we get $\lim_{c \to \infty} \mathtt{amls}(\alpha, c, \beta) = \mathtt{brute}(\beta)$. $\qquad \square$

We can now proceed to the proof of Theorem 2.18.

Proof of Theorem 2.18. Let $\beta > 1$ and let $\mathcal{L}$ be a specification list. Pick an arbitrary element $(\alpha^*, c^*) \in \mathcal{L}$ (recall that a specification list is always non-empty) and observe that $\alpha^*, c^* \geq 1$. Now, we have

$$
\begin{aligned}
\mathtt{best}(\mathcal{L}, \beta) &= \mathtt{amls}(\mathcal{L}, \beta) \\
&= \exp\left(\max_{0 \leq \kappa \leq \frac{1}{\beta}} \min_{(\alpha, c) \in \mathcal{L}} g^*_{\alpha, \beta, c}(\kappa) \right)
\end{aligned}
$$

$$\leq \quad \exp\left(\max_{0\leq\kappa\leq\frac{1}{\beta}} g^*_{\alpha^*,c^*,\beta}(\kappa)\right)$$
$$= \quad \mathtt{amls}(\alpha^*, c^*, \beta)$$
$$< \quad \mathtt{brute}(\beta),$$

where the first equality follows from Corollary 2.12, the second and third equalities follows from (13), and the last inequality follows from Lemma 8.2.

For the second part, let $\alpha \geq 1$. Then

$$\lim_{c\to\infty} \mathtt{best}(\alpha,c,\beta) \;=\; \lim_{c\to\infty} \mathtt{amls}(\alpha,c,\beta) \;=\; \mathtt{brute}(\beta),$$

where the first equality follows from Corollary 2.12, and the second equality follows from Lemma 8.3.
□

9 Monotonicity Properties

In this section we prove that $\mathtt{amls}(\alpha, c, \beta)$ is strictly monotone in α in the interval $[1, \beta]$, and use this result to prove Lemma 2.20.

Lemma 9.1. *For every $\beta \geq \alpha' > \alpha \geq 1$ and every $c > 1$ it holds that $\mathtt{amls}(\alpha, c, \beta) < \mathtt{amls}(\alpha', c, \beta)$.*

The proof of Lemma 9.1 is given towards the end of this section. We first use Lemma 9.1 to prove Lemma 2.20.

Proof of Lemma 2.20. Let $\beta > \alpha \geq 1$ and $c > 1$. Then

$$\mathtt{best}(\alpha,c,\beta) \;=\; \mathtt{amls}(\alpha,c,\beta) \;<\; \mathtt{amls}(\beta,c,\beta) \;=\; \mathtt{best}(\beta,c,\beta) \;=\; \mathtt{amls}_{\alpha=\beta}(\beta,c).$$

The first and second equalities follow from Corollary 2.12. The inequality holds by Lemma 9.1, and the last equality follows form Lemma 2.19.
□

In order to prove Lemma 9.1 we give an alternative formula for $g^*_{\alpha,\beta,c}$ (12) as a solution for a continuous optimization problem in two variables. Though this alternative formula uses a continuous optimization problem, it is inspired by an interpretation of the discrete analysis of Algorithms 1 and 2. The algorithm samples a t-element set $X \subseteq U$, and the analysis focuses on samples which satisfy $|X \cap \mathtt{OPT}| \geq y$, for a carefully selected y. Subsequently, the α-extension oracle is invoked with the query $(X, k - y)$. The algorithm optimally selects $y = \left(1 - \frac{\beta}{\alpha}\right) \cdot k + \frac{t}{\alpha}$. The analysis we present in this section leaves y as an additional parameter to be optimized. Though this seems to only yield a more involved formula, this formula turns out to be useful to determine the behavior of $\mathtt{amls}(\alpha, c, \beta)$ as α changes.

For every $1 \leq \alpha \leq \beta$ we define

$$X_{\alpha,\beta}(\kappa, \tau) \;:=\; \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa + \frac{\tau}{\alpha}. \tag{82}$$

Similarly, for every $1 \leq \alpha \leq \beta$ and $\kappa \in \left[0, \frac{1}{\beta}\right]$ we define a set

$$D_{\alpha,\beta}(\kappa) \;:=\; \left\{(\tau, y) \in \mathbb{R}^2 \mid 0 \leq \tau \leq \beta \cdot \kappa, \;\; \max\{0, X_{\alpha,\beta}(\kappa, \tau)\} \leq y \leq \min\{\kappa, \tau\}\right\}. \tag{83}$$

Finally, for every $1 \leq \alpha \leq \beta$ and $\kappa \in \left[0, \frac{1}{\beta}\right]$ we define the function

$$\tilde{g}^{[\kappa]}_c(\tau, y) := (\kappa - y) \cdot \ln(c) - \tau \cdot \mathcal{H}\left(\frac{y}{\tau}\right) - (1 - \tau) \cdot \mathcal{H}\left(\frac{\kappa - y}{1 - \tau}\right) + \mathcal{H}(\kappa). \tag{84}$$

It can be easily verified that

$$g^*_{\alpha,\beta,c}(\kappa, \tau) = \tilde{g}^{[\kappa]}_c\left(\tau, X_{\alpha,\beta}\left(\kappa, \tau\right)\right) \tag{85}$$

unless $\tau \in \{0, 1\}$.

The next lemma provides the alternative formula for $g^*_{\alpha,\beta,c}$.

Lemma 9.2. *Let $\beta \geq \alpha \geq 1$, $c > 1$ and $\kappa \in \left(0, \frac{1}{\beta}\right)$. Then*

$$g_{\alpha,\beta,c}^*(\kappa) = \min_{(\tau,y) \in D_{\alpha,\beta}(k)} \tilde{g}_c^{[\kappa]}(\tau, y). \tag{86}$$

Moreover, for all $(\tau, y) \in D_{\alpha,\beta}(\kappa)$ such that $g_{\alpha,\beta,c}^*(\kappa) = \tilde{g}_c^{[\kappa]}(\tau, y)$, we have $y = X_{\alpha,\beta}(\kappa, \tau)$.

Proof. The proof uses an alternative representation of the set $D_{\alpha,\beta}(\kappa)$. We define

$$E_{\alpha,\beta}(\kappa) := \left\{ (\tau, y) \in \mathbb{R}^2 \; \middle| \; 0 \leq y \leq \kappa, \; y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa \right) \right\}. \tag{87}$$

Claim 9.3. $E_{\alpha,\beta}(\kappa) = D_{\alpha,\beta}(\kappa)$.

Proof. Let $(\tau, y) \in D_{\alpha,\beta}(\kappa)$. Then

$$0 \leq \max\{0, X_{\alpha,\beta}(\kappa, \tau)\} \leq y \leq \min\{\kappa, \tau\} \leq \kappa.$$

Furthermore, $y \leq \min\{\kappa, \tau\} \leq \tau$ and

$$y \geq \max\{0, X_{\alpha,\beta}(\kappa, \tau)\} \geq X_{\alpha,\beta}(\kappa, \tau) = \left(1 - \frac{\beta}{\alpha}\right)\kappa + \frac{\tau}{\alpha}$$

and thus, $\tau \leq \alpha \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa \right)$. Overall, we have that $0 \leq y \leq \kappa$ and $t \leq \tau \leq \alpha \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa \right)$.
By (87) we have $(\tau, y) \in E_{\alpha,\beta}(\kappa)$, and we conclude that

$$D_{\alpha,\beta}(\kappa) \subseteq E_{\alpha,\beta}(\kappa). \tag{88}$$

Similarly, let $(\tau', y') \in E_{\alpha,\beta}(\kappa)$. It follows that $\tau' \geq y' \geq 0$. Furthermore,

$$\tau' \leq \alpha \left(y' - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa \right) \leq \alpha \cdot \left(\kappa - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa \right) = \beta \cdot \kappa,$$

where the second inequality follows from $y' \leq \kappa$. So together, we obtain that

$$0 \leq \tau' \leq \beta \cdot \kappa. \tag{89}$$

By re-arranging the inequality $\tau' \leq \alpha \left(y' - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa \right)$ we obtain

$$y' \geq \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa + \frac{\tau'}{\alpha} = X_{\alpha,\beta}(\kappa, \tau').$$

As $(\tau', y') \in E_{\alpha,\beta}(\kappa)$, it also holds that $y' \geq 0$. So $y' \geq \max\{0, X_{\alpha,\beta}(\kappa, \tau')\}$. Finally, since $(\tau', y') \in E_{\alpha,\beta}(\kappa)$ we have $y' \leq \kappa$ and $y' \leq \tau'$, and hence $y' \leq \min\{\kappa, \tau'\}$. So overall

$$\max\{0, X_{\alpha,\beta}(\kappa, \tau')\} \leq y' \leq \min\{\kappa, \tau'\}. \tag{90}$$

By (89) and (90) it holds that $(\tau', y') \in D_{\alpha,\beta}(\kappa)$. Thus

$$E_{\alpha,\beta}(\kappa) \subseteq D_{\alpha,\beta}(\kappa). \tag{91}$$

By (88) and (91) we have $D_{\alpha,\beta}(\kappa) = E_{\alpha,\beta}(\kappa)$. ⌟

Using Claim 9.3 we have

$$\min_{(\tau,y) \in D_{\alpha,\beta}(k)} \tilde{g}_c^{[\kappa]}(\tau, y) = \min_{(\tau,y) \in E_{\alpha,\beta}(k)} \tilde{g}_c^{[\kappa]}(\tau, y) = \min_{0 \leq y \leq \kappa} \; \min_{y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau, y). \tag{92}$$

In order to simplify (92) we use some analytical properties of $\tilde{g}_c^{[\kappa]}(\tau, y)$.

Claim 9.4. *It holds that* $\frac{\partial \tilde{g}_c^{[\kappa]}(\tau,y)}{\partial \tau} = \ln\left(1 - \frac{y}{\tau}\right) - \ln\left(1 - \frac{\kappa - y}{1-\tau}\right).$

Proof. For every $a \in \mathbb{R}$ define $q_a(x) = x \cdot \mathcal{H}\left(\frac{a}{x}\right)$. Using basic differentiation rules we have,

$$
\begin{aligned}
\frac{\partial q_a(x)}{\partial x} &= \mathcal{H}\left(\frac{a}{x}\right) + x \cdot \frac{-a}{x^2} \cdot \ln\left(\frac{1 - \frac{a}{x}}{\left(\frac{a}{x}\right)}\right) \\
&= -\frac{a}{x} \cdot \ln\left(\frac{a}{x}\right) - \left(1 - \frac{a}{x}\right) \cdot \ln\left(1 - \frac{a}{x}\right) - \frac{a}{x} \cdot \ln\left(1 - \frac{a}{x}\right) + \frac{a}{x} \cdot \ln\left(\frac{a}{x}\right) \\
&= -\ln\left(1 - \frac{a}{x}\right).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\frac{\partial \tilde{g}_c^{[\kappa]}(\tau,y)}{\partial \tau} &= \frac{\partial}{\partial \tau}\left((\kappa - y) \cdot \ln(c) - \tau \cdot \mathcal{H}\left(\frac{y}{\tau}\right) - (1 - \tau) \cdot \mathcal{H}\left(\frac{\kappa - y}{1 - \tau}\right) + \mathcal{H}(\kappa)\right) \\
&= -\frac{\partial q_y(\tau)}{\partial \tau} - \frac{\partial q_{\kappa - t}(1 - \tau)}{\partial \tau} \\
&= \ln\left(1 - \frac{y}{\tau}\right) - \ln\left(1 - \frac{\kappa - y}{1 - \tau}\right).
\end{aligned}
$$

⌋

We use Claim 9.4 to show the following.

Claim 9.5. *For every $0 \le y \le \kappa$, it holds that $\tilde{g}_c^{[\kappa]}(\tau,y)$ is strictly convex as a function of τ, and has a minimum at $\tau = \frac{y}{\kappa}$.*

Proof. For every $0 \le y \le \kappa$ the expression $1 - \frac{y}{\tau}$ is increasing with τ and the expression $1 - \frac{\kappa - y}{1 - \tau}$ is decreasing with τ. Furthermore, $1 - \frac{y}{\tau}$ is *strictly* increasing, unless $y = 0$, and in this case $1 - \frac{\kappa - y}{1 - \tau}$ is *strictly* decreasing. It follows that $\frac{\partial \tilde{g}_c^{[\kappa]}(\tau,y)}{\partial \tau} = \ln\left(1 - \frac{y}{\tau}\right) - \ln\left(1 - \frac{\kappa - y}{1 - \tau}\right)$ is strictly increasing with τ for every $0 \le y \le \kappa$. We conclude that $\tilde{g}_c^{[\kappa]}(\tau,y)$ is strictly convex as a function of τ (for a fixed $y \in [0, \kappa]$). Furthermore,

$$
\left.\frac{\partial \tilde{g}_c^{[\kappa]}(\tau,y)}{\partial \tau}\right|_{\tau = \frac{y}{\kappa}} = \ln\left(1 - \frac{y}{\frac{y}{\kappa}}\right) - \ln\left(1 - \frac{\kappa - y}{1 - \frac{y}{\kappa}}\right) = \ln(1 - \kappa) - \ln(1 - \kappa) = 0,
$$

so the minimum of $\tilde{g}_c^{[\kappa]}(\tau,y)$, as a function of τ (for a fixed y), is at $\tau = \frac{y}{\kappa}$. ⌋

For every $0 \le y \le \kappa$, the value of $\tilde{g}_c^{[\kappa]}(\tau,y)$ at its minimum is

$$
\begin{aligned}
\tilde{g}_c^{[\kappa]}\left(\frac{y}{\kappa}, y\right) &= (\kappa - y) \cdot \ln(c) - \frac{y}{\kappa} \cdot \mathcal{H}\left(\frac{y}{\frac{y}{\kappa}}\right) - \left(1 - \frac{y}{\kappa}\right) \cdot \mathcal{H}\left(\frac{\kappa - y}{1 - \frac{y}{\kappa}}\right) + \mathcal{H}(\kappa) \\
&= (\kappa - y) \cdot \ln(c) - \frac{y}{\kappa} \cdot \mathcal{H}(\kappa) - \left(1 - \frac{y}{\kappa}\right) \cdot \mathcal{H}(\kappa) + \mathcal{H}(\kappa) \\
&= (\kappa - y) \cdot \ln(c).
\end{aligned}
\tag{93}
$$

By the above, for every $0 \le y \le \kappa$ such that $\frac{y}{\kappa} \le \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)$ it holds that

$$
\min_{y \le \tau \le \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau,y) = \tilde{g}_c^{[\kappa]}\left(\frac{y}{\kappa}, y\right) = (\kappa - y) \cdot \ln(c).
\tag{94}
$$

63

The first equality holds since $\tilde{g}_c^{[\kappa]}(\tau, y)$ is convex with a minimum at $\frac{y}{\kappa} \geq y$, as a function of τ (Claim 9.5). The second equality follows from (93). Also, observe that

$$
\begin{aligned}
\frac{y}{\kappa} &\leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right) \\
\iff \quad y\left(\frac{1}{\kappa} - \alpha\right) &\leq -(\alpha - \beta) \cdot \kappa \\
\iff \quad y &\leq \frac{\beta - \alpha}{1 - \alpha\kappa} \cdot \kappa^2 = M_{\alpha,\beta}(\kappa) \cdot \kappa,
\end{aligned}
\tag{95}
$$

where the second transition holds as $\frac{1}{\kappa} > \beta \geq \alpha$. Recall that $M_{\alpha,\beta}(\kappa)$ is defined in (10).

By (94) and (95) we have

$$
\begin{aligned}
&\min_{0 \leq y \leq M_{\alpha,\beta}(\kappa) \cdot \kappa} \quad \min_{y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau, y) \\
&= \min_{0 \leq y \leq M_{\alpha,\beta}(\kappa) \cdot \kappa} (\kappa - y) \ln(c) \\
&= (\kappa - M_{\alpha,\beta}(\kappa) \cdot \kappa) \cdot \ln c \\
&= \tilde{g}_c^{[\kappa]}\left(\frac{M_{\alpha,\beta}(\kappa) \cdot \kappa}{\kappa}, \; M_{\alpha,\beta}(\kappa) \cdot \kappa\right) \\
&= \min_{M_{\alpha,\beta}(\kappa) \cdot \kappa \leq \tau \leq \alpha \cdot \left(M_{\alpha,\beta}(\kappa) \cdot \kappa - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}\left(\tau, \; M_{\alpha,\beta}(\kappa) \cdot \kappa\right).
\end{aligned}
\tag{96}
$$

It can be easily verified that $M_{\alpha,\beta}(\kappa) \leq 1$. Thus, we can use (96) to change the range of y in (92) as follows:

$$
\begin{aligned}
\min_{(\tau,y) \in D_{\alpha,\beta}(k)} \tilde{g}_c^{[\kappa]}(\tau, y) &= \min_{0 \leq y \leq \kappa} \quad \min_{y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau, y) \\
&= \min_{M_{\alpha,\beta}(\kappa) \cdot \kappa \leq y \leq \kappa} \quad \min_{y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau, y).
\end{aligned}
\tag{97}
$$

By Claim 9.5, for every $0 \leq y \leq \kappa$ such that $\frac{y}{\kappa} \geq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)$, it holds that

$$
\min_{y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau, y) = \tilde{g}_c^{[\kappa]}\left(\alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right), \; y\right).
\tag{98}
$$

Similarly to (95) it holds that

$$
\frac{y}{\kappa} \geq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right) \quad \iff \quad y \geq M_{\alpha,\beta}(\kappa) \cdot \kappa.
\tag{99}
$$

Thus, using (98) and (97), we get

$$
\begin{aligned}
\min_{(\tau,y) \in D_{\alpha,\beta}(k)} \tilde{g}_c^{[\kappa]}(\tau, y) &= \min_{M_{\alpha,\beta}(\kappa) \cdot \kappa \leq y \leq \kappa} \quad \min_{y \leq \tau \leq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)} \tilde{g}_c^{[\kappa]}(\tau, y) \\
&= \min_{M_{\alpha,\beta}(\kappa) \cdot \kappa \leq y \leq \kappa} \tilde{g}_c^{[\kappa]}\left(\alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right), \; y\right) \\
&= \min_{M_{\alpha,\beta}(\kappa) \leq \tau \leq \beta \cdot \kappa} \tilde{g}_c^{[\kappa]}(\tau, X_{\alpha,\beta}(\kappa, \tau)) \\
&= \min_{M_{\alpha,\beta}(\kappa) \leq \tau \leq \beta \cdot \kappa} g_{\alpha,\beta,c}(\kappa, \tau) \\
&= g_{\alpha,\beta,c}^*(\kappa).
\end{aligned}
\tag{100}
$$

The third equality follows from substitution y with $\tau = \alpha \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)$, which is equivalent to $y = X_{\alpha,\beta}(\kappa, \tau)$. The forth equality follows from $\tilde{g}^{[\kappa]}(\tau, X_{\alpha,\beta}(\kappa, \tau)) = g_{\alpha,\beta,c}(\kappa, \tau)$. That last equality follows from (12). Observe that (100) implies (86).

Let $(\tau, y) \in D_{\alpha,\beta}(\kappa) = E_{\alpha,\beta}(\kappa)$ such that $g^*_{\alpha,\beta,c}(\kappa) = \tilde{g}^{[\kappa]}_c(\tau, y)$. To complete the proof of the lemma we are left to show that $y = X_{\alpha,\beta}(\kappa, \tau)$. By (100) it follows that

$$\tilde{g}^{[\kappa]}_c(\tau, y) \;=\; \min_{(\tau',y')\in D_{\alpha,\beta}(\kappa)} \tilde{g}^{[\kappa]}_c(\tau', y') \;=\; \min_{(\tau',y')\in E_{\alpha,\beta}(\kappa)} \tilde{g}^{[\kappa]}_c(\tau', y'). \tag{101}$$

Assume towards contradiction that $y < M_{\alpha,\beta}(\kappa) \cdot \kappa$. Then, since $(y, \tau) \in E_{\alpha,\beta}(\kappa)$, we have

$$\begin{aligned}
\tilde{g}^{[\kappa]}_c(\tau, y) &\geq \min_{y \leq \tau' \leq \alpha\cdot\left(y - \left(1-\frac{\beta}{\alpha}\right)\cdot\kappa\right)} \tilde{g}^{[\kappa]}_c(\tau', y) \\
&= (\kappa - y) \cdot \ln(c) \\
&> (\kappa - M_{\alpha,\beta}(\kappa) \cdot \kappa) \cdot \ln(c) \\
&= \min_{M_{\alpha,\beta}(\kappa)\cdot\kappa \,\leq\, \tau' \,\leq\, \alpha\cdot\left(M_{\alpha,\beta}(\kappa)\cdot\kappa - \left(1-\frac{\beta}{\alpha}\right)\cdot\kappa\right)} \tilde{g}^{[\kappa]}_c\left(\tau',\; M_{\alpha,\beta}(\kappa) \cdot \kappa\right) \\
&\geq \min_{(\tau',t')\in E_{\alpha,\beta}(\kappa)} \tilde{g}^{[\kappa]}_c\left(\tau',\; y'\right).
\end{aligned} \tag{102}$$

The first inequality follows from (101) and the definition of $E_{\alpha,\beta}$ (87). The first equality follows from (94) and (95). The second equality uses (96). The last inequality is also a consequence of the definition of $E_{\alpha,\beta}$ (87). Observe that (102) contradict (101), so $y \geq M_{\alpha,\beta}(\kappa) \cdot \kappa$.

By (101) it holds that

$$\tilde{g}^{[\kappa]}_c(\tau, y) \;=\; \min_{y \leq \tau' \leq \alpha\cdot\left(y - \left(1-\frac{\beta}{\alpha}\right)\cdot\kappa\right)} \tilde{g}^{[\kappa]}_c(\tau', y) \tag{103}$$

By Claim 9.5 we have that $\tilde{g}^{[\kappa]}(\tau', y)$ is strictly convex as a function of τ' with a minimum at $\tau' = \frac{y}{\kappa}$ and by (99) we have $\frac{y}{\kappa} \geq \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)$. Thus, the minimum of the RHS of (103) is at $\tau' = \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)$. Therefore, $\tau = \alpha \cdot \left(y - \left(1 - \frac{\beta}{\alpha}\right) \cdot \kappa\right)$, and by rearranging the term we get $y = X_{\alpha,\beta}(\kappa, \tau)$. $\qquad\square$

Using Lemma 9.2 we can easily derive the following lemma.

Lemma 9.6. *Let* $\beta \geq \alpha' > \alpha \geq 1$, $c > 1$ *and* $\kappa \in \left(0, \frac{1}{\beta}\right)$. *Then* $g^*_{\alpha,\beta,c}(\kappa) < g^*_{\alpha',\beta,c}(\kappa)$.

Proof. By Lemma 9.2 we have $g^*_{\alpha',\beta,c}(\kappa) = \min_{(\tau,y)\in D_{\alpha',\beta}(\kappa)} \tilde{g}^{[\kappa]}_c(\tau, y)$. Thus, there is $(\tau', y') \in D_{\alpha',\beta}$ such that $\tilde{g}^{[\kappa]}_c(\tau', y') = g^*_{\alpha',\beta,c}(\kappa)$ and $y' = X_{\alpha',\beta}(\kappa, \tau')$.

Assume towards contradiction that $\tau' = \beta\kappa$. Then

$$g^*_{\alpha',\beta,c}(\kappa) \;=\; \tilde{g}^{[\kappa]}_c(\tau', y') \;=\; \tilde{g}^{[\kappa]}_c(\beta \cdot \kappa, X_{\alpha',\beta}(\kappa, \beta \cdot \kappa)) \;=\; g_{\alpha',\beta,c}(\kappa, \beta \cdot \kappa).$$

where the last equality follows from (85). However, by Lemma 2.13 it holds that $g^*_{\alpha',\beta,c}(\kappa) < g_{\alpha',\beta,c}(\kappa, \beta \cdot \kappa)$, contradicting the above. Thus $\tau' \neq \beta\kappa$.

As $(\tau', y') \in D_{\alpha',\beta}$ it also holds that $0 \leq \tau' < \beta \cdot \kappa$ and $0 \leq y' \leq \min\{\kappa, \tau'\}$. Furthermore,

$$\begin{aligned}
y' &= X_{\alpha',\beta}(\kappa, \tau') \\
&= \left(1 - \frac{\beta}{\alpha'}\right) \cdot \kappa + \frac{\tau'}{\alpha'} \\
&= 1 - \frac{1}{\alpha'}\left(\beta \cdot \kappa - \tau'\right) \\
&> 1 - \frac{1}{\alpha}\left(\beta \cdot \kappa - \tau'\right) \\
&= X_{\alpha,\beta}(\kappa, \tau'),
\end{aligned}$$

where the inequality holds since $\tau' < \beta \cdot \kappa$, and $\alpha' > \alpha$. Thus $(\tau', y') \in D_{\alpha,\beta}(\kappa)$ and $y' \neq X_{\alpha,\beta}(\kappa, \tau')$. Using Lemma 9.2 once more we get

$$g^*_{\alpha',\beta,c}(\kappa) \;=\; \tilde{g}^{[\kappa]}_c(\tau', y') \;>\; \min_{(\tau,y)\in D_{\alpha,\beta}} \tilde{g}^{[\kappa]}_c(\tau, t) \;=\; g^*_{\alpha,\beta,c}(\kappa). \qquad\square$$

Lemma 9.1 essentially follows from Lemma 9.6, though the proof itself involves some technical steps which exclude the possibility of a corner case.

Proof of Lemma 9.1. Define $\tilde{\kappa} = \frac{1}{2} \cdot \frac{1}{\beta}$. Since $\mathcal{H}(x)$ is concave, for every $M_{\alpha,\beta}(\tilde{\kappa}) \leq \tau < \beta\tilde{\kappa}$, it holds that

$$
\begin{aligned}
g_{\alpha,\beta,c}(\tilde{\kappa}, \tau) & = \frac{\beta\tilde{\kappa} - \tau}{\alpha} \ln c - \tau \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\tilde{\kappa}, \tau)\right) - (1 - \tau) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\tilde{\kappa}, \tau)\right) + \mathcal{H}\left(\tilde{\kappa}\right) \\
& \geq \frac{\beta\tilde{\kappa} - \tau}{\alpha} \ln c - \mathcal{H}\left(\tau \cdot \gamma_{\alpha,\beta}(\tilde{\kappa}, \tau) + (1 - \tau) \cdot \delta_{\alpha,\beta}(\tilde{\kappa}, \tau)\right) + \mathcal{H}\left(\tilde{\kappa}\right) \\
& = \frac{\beta\tilde{\kappa} - \tau}{\alpha} \ln c - \mathcal{H}\left(\tilde{\kappa}\right) + \mathcal{H}\left(\tilde{\kappa}\right) \\
& = \frac{\beta\tilde{\kappa} - \tau}{\alpha} \ln c \; > \; 0,
\end{aligned}
\tag{104}
$$

where the second equality holds as $\tau \cdot \gamma_{\alpha,\beta}(\tilde{\kappa}, \tau) + (1 - \tau) \cdot \delta_{\alpha,\beta}(\tilde{\kappa}, \tau) = \tilde{\kappa}$ and the last inequality uses $c > 1$. Furthermore,

$$
\begin{aligned}
& g_{\alpha,\beta,c}(\tilde{\kappa}, \beta \cdot \tilde{\kappa}) \\
& = \frac{\beta\tilde{\kappa} - \beta \cdot \tilde{\kappa}}{\alpha} \ln c - \beta \cdot \tilde{\kappa} \cdot \mathcal{H}\left(\gamma_{\alpha,\beta}(\tilde{\kappa}, \beta \cdot \tilde{\kappa})\right) - (1 - \beta \cdot \tilde{\kappa}) \cdot \mathcal{H}\left(\delta_{\alpha,\beta}(\tilde{\kappa}, \beta \cdot \tilde{\kappa})\right) + \mathcal{H}\left(\tilde{\kappa}\right) \\
& = -\beta \cdot \tilde{\kappa} \cdot \mathcal{H}\left(\frac{1}{\beta}\right) - (1 - \beta \cdot \tilde{\kappa}) \cdot \mathcal{H}\left(0\right) + \mathcal{H}\left(\tilde{\kappa}\right) \\
& > -\mathcal{H}\left(\beta\tilde{\kappa} \cdot \frac{1}{\beta} + (1 - \beta\tilde{\kappa}) \cdot 0\right) + \mathcal{H}\left(\tilde{\kappa}\right) \; = \; 0,
\end{aligned}
\tag{105}
$$

where the inequality holds as $\mathcal{H}$ is strictly concave. By (104) and (105) it follows that

$$
g^*_{\alpha,\beta,c}(\tilde{\kappa}) = \min_{M_{\alpha,\beta}(\tilde{\kappa}) \leq \tau \leq \beta\tilde{\kappa}} g_{\alpha,\beta,c}(\tilde{\kappa}, \tau) > 0.
$$

Now, there exists $\kappa \in \left[0, \frac{1}{\beta}\right]$ such that

$$
g^*_{\alpha,\beta,c}(\kappa) \; = \; \max_{\kappa' \in \left[0, \frac{1}{\beta}\right]} g^*_{\alpha,\beta,c}(\kappa') \; > \; 0.
$$

It can be easily verified that $g^*_{\alpha,\beta,c}(0) = g^*_{\alpha,\beta,c}\left(\frac{1}{\beta}\right) = 0$ (select $\tau = 0$ in the first and $\tau = \beta \cdot \kappa$ in the latter case). So $0 < \kappa < \frac{1}{\beta}$.

Thus, by Lemma 9.6, we have

$$
\begin{aligned}
\ln\left(\mathtt{amls}(\alpha, \beta, c)\right) & = \max_{k' \in \left[0, \frac{1}{\beta}\right]} g^*_{\alpha,\beta,c}(\kappa') \\
& = g^*_{\alpha,\beta,c}(\kappa) \\
& < g^*_{\alpha',\beta,c}(\kappa) \\
& \leq \max_{k' \in \left[0, \frac{1}{\beta}\right]} g^*_{\alpha',\beta,c}(\kappa') \\
& = \ln\left(\mathtt{amls}(\alpha', \beta, c)\right)
\end{aligned}
$$

The first and last equality follows from (13). Thus, $\mathtt{amls}(\alpha, \beta, c) < \mathtt{amls}(\alpha', \beta, c)$. $\qquad\square$

10 Conclusion

In this paper we studied how exponential-time approximation algorithms can be obtained from existing polynomial-time approximation algorithms, existing parameterized exact algorithms and existing parameterized approximation algorithms. We provided a theoretical oracle model by which the above question can be rigorously studied and showed that the approximate monotone local search

approach [26, 31] attains optimal running times (up to polynomial factors). Furthermore, we provided the mathematical machinery to compute the running time of the resulting algorithms in practice.

While previous works on monotone local search [26, 31, 48] only provided algorithmic results, in this work we use a restricted oracle model in which we are also able to show the optimality of our algoritms (which in particular implies the algorithms from [26, 31] are optimal). This way, we provide a complete answer of how to repurpose (any finite number of) parameterized approximation algorithms (which includes polynomial-time approximations and exact parameterized algorithms as special cases) for the design of exponential-time (approximation) algorithms.

Still, our work raises a number of follow-up questions. First, we focused on allowing a finite number of extension oracles. However, for problems such as VERTEX COVER, there is a parameterized α-approximation algorithm for every $\alpha \geq 1$. In the language of this work, this gives rise to an infinite number of extension oracles; it would be interesting to properly formalize such a setting and extend our result to it. Even if we only want to provide a finite number of extension oracles, it is already unclear how to choose these oracles in an optimal way. In fact, we already encountered this problem in Section 3.3 where we adopted a simple discretization appraoch (which is likely not optimal) to choose a finite number of extension oracles.

The second question asks what happens with other types of oracles. Indeed, in this work, we focused on repurposing parameterized approximation algorithms (which includes polynomial-time approximations and exact parameterized algorithms as special cases) for the design of exponential-time approximation algorithms. Can we find other types of algorithms that can be repurposed in a similar way? For example, is it possible to repurpose exact exponential-time algorithms (e.g., VERTEX COVER can be solved in time $\mathcal{O}^*(1.1996^n)$ [66]) in a meaningful way? More generally, in [34] the authors show that monotone local search can also be used to convert a $c^k \cdot b^n \cdot n^{O(1)}$ time algorithm into a $\left(1 + b - \frac{1}{c}\right)^n \cdot n^{O(1)}$ time algorithm for the same problem. What is the optimal way of repurposing such an algorithm in order to obtain an exponential β-approximation algorithm for any $\beta > 1$? We remark that our lower-bound technique can be used to show the result of [34] already repurposes such algorithms in an optimal way in the exact setting.

Finally, we ask about weighted problems. Similarly to our setting, one can define an extension oracle model for *weighted* problems in which the objective is to find set $S \in \mathcal{F}$ of (approximately) minimum weight. Using this model its possible to define a weighted variant of `best`. Is this variant equal to the function `best` defined in this paper? Which algorithm attains this cost for weighted problems?

References

[1] Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshtanov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In Evangelos Kranakis, Gonzalo Navarro, and Edgar Chávez, editors, *LATIN 2016: Theoretical Informatics - 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings*, volume 9644 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2016. `doi:10.1007/978-3-662-49529-2\_1`.

[2] Manuel Aprile, Matthew Drescher, Samuel Fiorini, and Tony Huynh. A tight approximation algorithm for the cluster vertex deletion problem. *Math. Program.*, 197(2):1069–1091, 2023. `doi:10.1007/s10107-021-01744-w`.

[3] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *J. ACM*, 62(5):42:1–42:25, 2015. `doi:10.1145/2775105`.

[4] Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discret. Math.*, 12(3):289–297, 1999. `doi:10.1137/S0895480196305124`.

[5] Nikhil Bansal, Parinya Chalermsook, Bundit Laekhanukit, Danupon Nanongkai, and Jesper Nederlof. New tools and connections for exponential-time approximation. *Algorithmica*, 81(10):3993–4009, 2019. `doi:10.1007/s00453-018-0512-8`.

[6] Reuven Bar-Yehuda and Shimon Even. A linear-time approximation algorithm for the weighted vertex cover problem. *J. Algorithms*, 2(2):198–203, 1981. `doi:10.1016/0196-6774(81)90020-1`.

[7] Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. Parameterized intractability of even set and shortest vector problem. *J. ACM*, 68(3):16:1–16:40, 2021. `doi:10.1145/3444942`.

[8] Anudhyan Boral, Marek Cygan, Tomasz Kociumaka, and Marcin Pilipczuk. A fast branching algorithm for cluster vertex deletion. *Theory Comput. Syst.*, 58(2):357–376, 2016. `doi:10.1007/s00224-015-9631-7`.

[9] Nicolas Bourgeois, Bruno Escoffier, and Vangelis Th. Paschos. Approximation of max independent set, min vertex cover and related problems by moderately exponential algorithms. *Discret. Appl. Math.*, 159(17):1954–1970, 2011. `doi:10.1016/j.dam.2011.07.009`.

[10] Ljiljana Brankovic and Henning Fernau. Parameterized approximation algorithms for hitting set. In Roberto Solis-Oba and Giuseppe Persiano, editors, *Approximation and Online Algorithms - 9th International Workshop, WAOA 2011, Saarbrücken, Germany, September 8-9, 2011, Revised Selected Papers*, volume 7164 of *Lecture Notes in Computer Science*, pages 63–76. Springer, 2011. `doi:10.1007/978-3-642-29116-6\_6`.

[11] Ljiljana Brankovic and Henning Fernau. A novel parameterised approximation algorithm for minimum vertex cover. *Theor. Comput. Sci.*, 511:85–108, 2013. `doi:10.1016/j.tcs.2012.12.003`.

[12] Nader H. Bshouty and Lynn Burroughs. Massaging a linear programming solution to give a 2-approximation for a generalization of the vertex cover problem. In Michel Morvan, Christoph Meinel, and Daniel Krob, editors, *STACS 98, 15th Annual Symposium on Theoretical Aspects of Computer Science, Paris, France, February 25-27, 1998, Proceedings*, volume 1373 of *Lecture Notes in Computer Science*, pages 298–308. Springer, 1998. `doi:10.1007/BFb0028569`.

[13] Yixin Cao. Linear recognition of almost interval graphs. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1096–1115. SIAM, 2016. `doi:10.1137/1.9781611974331.ch77`.

[14] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM J. Comput.*, 49(4):772–810, 2020. `doi:10.1137/18M1166869`.

[15] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010. `doi:10.1016/j.tcs.2010.06.026`.

[16] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5):21:1–21:19, 2008. `doi:10.1145/1411509.1411511`.

[17] Yijia Chen and Bingkai Lin. The constant inapproximability of the parameterized dominating set problem. *SIAM J. Comput.*, 48(2):513–533, 2019. `doi:10.1137/17M1127211`.

[18] Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized approximation algorithms for bidirected steiner network problems. *ACM Trans. Algorithms*, 17(2):12:1–12:68, 2021. `doi:10.1145/3447584`.

[19] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, 2nd edition, 2006. `doi:10.1002/047174882X`.

[20] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. `doi:10.1007/978-3-319-21275-3`.

[21] Marek Cygan, Lukasz Kowalik, and Mateusz Wykurz. Exponential-time approximation of weighted set cover. *Inf. Process. Lett.*, 109(16):957–961, 2009. `doi:10.1016/j.ipl.2009.05.003`.

[22] Oswaldo de Oliveira. The implicit and inverse function theorems: easy proofs. *Real Anal. Exchange*, 39(1):207–218, 2013/14. `doi:10.14321/realanalexch.39.1.0207`.

[23] Matthew Drescher, Samuel Fiorini, and Tony Huynh. A simple $(2+\epsilon)$-approximation algorithm for split vertex deletion. *CoRR*, abs/2009.11056, 2020. URL: `https://arxiv.org/abs/2009.11056`, `arXiv:2009.11056`.

[24] Eduard Eiben, Clément Rambaud, and Magnus Wahlström. On the parameterized complexity of symmetric directed multicut. In Holger Dell and Jesper Nederlof, editors, *17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany*, volume 249 of *LIPIcs*, pages 11:1–11:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.IPEC.2022.11`.

[25] Bruno Escoffier, Vangelis Th. Paschos, and Emeric Tourniaire. Super-polynomial approximation branching algorithms. *RAIRO Oper. Res.*, 50(4-5):979–994, 2016. `doi:10.1051/ro/2015060`.

[26] Baris Can Esmer, Ariel Kulik, Dániel Marx, Daniel Neuen, and Roohani Sharma. Faster exponential-time approximation algorithms using approximate monotone local search. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 50:1–50:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.ESA.2022.50`.

[27] Guy Even, Joseph Naor, and Leonid Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. *SIAM J. Comput.*, 30(4):1231–1252, 2000. `doi:10.1137/S0097539798340047`.

[28] 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`.

[29] Michael R. Fellows, Ariel Kulik, Frances A. Rosamond, and Hadas Shachnai. Parameterized approximation via fidelity preserving transformations. *J. Comput. Syst. Sci.*, 93:30–40, 2018. `doi:10.1016/j.jcss.2017.11.001`.

[30] Fedor V. Fomin, Serge Gaspers, Dieter Kratsch, Mathieu Liedloff, and Saket Saurabh. Iterative compression and exact algorithms. *Theor. Comput. Sci.*, 411(7-9):1045–1053, 2010. `doi:10.1016/j.tcs.2009.11.012`.

[31] Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, and Saket Saurabh. Exact algorithms via monotone local search. *J. ACM*, 66(2):8:1–8:23, 2019. `doi:10.1145/3284176`.

[32] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010. `doi:10.1007/978-3-642-16533-7`.

[33] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover. In Andrzej Lingas, Rolf G. Karlsson, and Svante Carlsson, editors, *Automata, Languages and Programming, 20nd International Colloquium, ICALP93, Lund, Sweden, July 5-9, 1993, Proceedings*, volume 700 of *Lecture Notes in Computer Science*, pages 64–75. Springer, 1993. `doi:10.1007/3-540-56939-1\_62`.

[34] Serge Gaspers and Edward J. Lee. Exact algorithms via multivariate subroutines. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 69:1–69:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPIcs.ICALP.2017.69`.

[35] Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, and M. S. Ramanujan. Faster parameterized algorithms for deletion to split graphs. *Algorithmica*, 71(4):989–1006, 2015. `doi:10.1007/s00453-013-9837-5`.

[36] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988. `doi:10.1007/978-3-642-97881-4`.

[37] Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Parameterized complexity of vertex cover variants. *Theory Comput. Syst.*, 41(3):501–520, 2007. `doi:10.1007/s00224-007-1309-3`.

[38] Venkatesan Guruswami and Euiwoong Lee. Simple proof of hardness of feedback vertex set. *Theory Comput.*, 12(1):1–11, 2016. `doi:10.4086/toc.2016.v012a006`.

[39] Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, lp-branching, and FPT algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016. `doi:10.1137/140962838`.

[40] Iyad A. Kanj, Guohui Lin, Tian Liu, Weitian Tong, Ge Xia, Jinhui Xu, Boting Yang, Fenghui Zhang, Peng Zhang, and Binhai Zhu. Improved parameterized and exact algorithms for cut problems on trees. *Theor. Comput. Sci.*, 607:455–470, 2015. `doi:10.1016/j.tcs.2015.06.010`.

[41] 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`.

[42] Ken-ichi Kawarabayashi and Bingkai Lin. A nearly 5/3-approximation FPT algorithm for min-k-cut. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 990–999. SIAM, 2020. `doi:10.1137/1.9781611975994.59`.

[43] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002*, page 25. IEEE Computer Society, 2002. `doi:10.1109/CCC.2002.1004334`.

[44] Steven G. Krantz and Harold R. Parks. *The implicit function theorem: history, theory, and applications*. Birkhäuser Boston, Inc., Boston, MA, 2002. `doi:10.1007/978-1-4612-0059-8`.

[45] Ariel Kulik and Hadas Shachnai. Analysis of two-variable recurrence relations with application to parameterized approximations. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 762–773. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00076`.

[46] Mithilesh Kumar and Daniel Lokshtanov. Faster exact and parameterized algorithm for feedback vertex set in tournaments. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 49:1–49:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPIcs.STACS.2016.49`.

[47] Beatriz Lafferriere, Gerardo Lafferriere, and Mau Nam Nguyen. *Introduction to Mathematical Analysis I*. Portland State University Library, 3rd edition, 2022. `doi:10.15760/pdxopen-34`.

[48] Edward Lee. *Exponential time algorithms via separators and random subsets*. PhD thesis, University of New South Wales, 2021. `doi:10.26190/unsworks/22740`.

[49] Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O(2.7^k)$ time. *ACM Trans. Algorithms*, 18(4):34:1–34:26, 2022. `doi:10.1145/3504027`.

[50] Bingkai Lin. Constant approximating k-clique is W[1]-hard. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1749–1756. ACM, 2021. `doi:10.1145/3406325.3451016`.

[51] Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. On lower bounds of approximating parameterized k-clique. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 90:1–90:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.ICALP.2022.90`.

[52] 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`.

[53] Daniel Lokshtanov, Pranabendu Misra, Joydeep Mukherjee, Geevarghese Philip, Fahad Panolan, and Saket Saurabh. A 2-approximation algorithm for feedback vertex set in tournaments. *CoRR*, abs/1809.08437, 2018. URL: `http://arxiv.org/abs/1809.08437`, `arXiv:1809.08437`.

[54] Daniel Lokshtanov, Pranabendu Misra, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. FPT-approximation for FPT problems. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 199–218. SIAM, 2021. `doi:10.1137/1.9781611976465.14`.

[55] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014. `doi:10.1145/2566616`.

[56] Pasin Manurangsi and Luca Trevisan. Mildly exponential time approximation algorithms for vertex cover, balanced separator and uniform sparsest cut. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 20:1–20:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPIcs.APPROX-RANDOM.2018.20`.

[57] Dániel Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008. `doi:10.1093/comjnl/bxm048`.

[58] The mpmath development team. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 1.3.0)*, 2023. URL: `https://mpmath.org/`.

[59] George Osipov and Magnus Wahlström. Parameterized complexity of equality mincsp. *CoRR*, abs/2305.11131, 2023. `arXiv:2305.11131`, `doi:10.48550/arXiv.2305.11131`.

[60] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes: the art of scientific computing, 3rd Edition*. Cambridge University Press, 2007. URL: `https://www.worldcat.org/oclc/123285342`.

[61] Georg Still. Lectures on parametric optimization: An introduction. *Optimization Online*, 2018.

[62] Pim van 't Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013. `doi:10.1007/s00453-012-9661-3`.

[63] Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001. `doi:10.1007/978-3-662-04565-7`.

[64] Magnus Wahlström. *Algorithms, measures and upper bounds for satisfiability and related problems.* PhD thesis, Linköping University, Sweden, 2007. URL: https://nbn-resolving.org/urn:nbn:se:liu:diva-8714.

[65] Magnus Wahlström. A single-exponential parameterized approximation algorithm for symmetric directed multicut. Personal communication, 2022.

[66] Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. *Inf. Comput.*, 255:126–146, 2017. doi:10.1016/j.ic.2017.06.001.

A   Problem Definitions

In this section, we give the problem definitions of all the problems discussed in the paper.

VERTEX COVER (VC)
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ has no edges.

PARTIAL VERTEX COVER (VC)
Input: An undirected graph G and an integer $t \geq 0$.
Question: Find a minimum set S of vertices of G such that $G - S$ has at most $|E(G)| - t$ many edges.

---
d-HITTING SET (d-HS)
Input: A universe U and set family $\mathcal{F} \subseteq \binom{U}{\leq d}$.
Question: Find a minimum set $S \subseteq U$ such that for each $F \in \mathcal{F}$, $S \cap F \neq \emptyset$.

FEEDBACK VERTEX SET (FVS)
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is an acyclic graph.

SUBSET FEEDBACK VERTEX SET (SUBSET FVS)
Input: An undirected graph G and a set $T \subseteq V(G)$.
Question: Find a minimum set S of vertices of G such that $G - S$ has no cycle that contains at least one vertex of T.

TOURNAMENT FEEDBACK VERTEX SET (TFVS)
Input: A tournament graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is an acyclic tournament.

DIRECTED FEEDBACK VERTEX SET (DFVS)
Input: A directed graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is a directed acyclic graph.

DIRECTED SUBSET FEEDBACK VERTEX SET (SUBSET DFVS)
Input: A directed graph G and a set $T \subseteq V(G)$.
Question: Find a minimum set S of vertices of G such that $G - S$ has no directed cycle that contains at least one vertex of T.

ODD CYCLE TRANSVERSAL (OCT)
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ has no cycle of odd length.

Directed Odd Cycle Transversal (DOCT)
Input: A directed graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ has no directed cycle of odd length.

Multicut
Input: An undirected graph G and a set $\mathcal{P} \subseteq V(G) \times V(G)$.
Question: Find a minimum set S of vertices of G such that $G - S$ has no path from u to v for any $(u, v) \in \mathcal{P}$

Edge Multicut on Trees
Input: A tree T and a set $\mathcal{P} \subseteq V(G) \times V(G)$.
Question: Find a minimum set S of edges of T such that $T - S$ has no path from u to v for any $(u, v) \in \mathcal{P}$

---

d-Steiner Multicut
Input: An undirected graph G, a family of at most d-sized sets $\mathcal{P} \subseteq \binom{V(G)}{d}$.
Question: Find a minimum set S of vertices of G such that for each $T \in \mathcal{P}$, there exists $u, v \in T$ such that $G - S$ either has no u to v path.

Directed Symmetric Multicut
Input: A directed graph G and a set $\mathcal{P} \subseteq V(G) \times V(G)$.
Question: Find a minimum set S of vertices of G such that for each $(u, v) \in \mathcal{P}$, $G - S$ either has no u to v path, or no v to u path.

Interval Vertex Deletion
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is an interval graph.

Proper Interval Vertex Deletion
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is a proper interval graph.

For the next problems, we require some additional definitions. A graph G is *cluster graph* if every connected component of G is a complete graph. We say G is a *block graph* if every 2-connected component of G is a complete graph. A *cograph* is a graph G which does not contain P_4 (a path on 4 vertices) is an induced subgraph. Finally, a graph G is a *split graph* if the vertex set can be partitioned into two sets $V(G) = I \uplus C$ such that I is an independent set and C is a clique in G.

Block Graph Vertex Deletion
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is a block graph.

Cluster Graph Vertex Deletion
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is a cluster graph.

Cograph Vertex Deletion
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is a cograph.

Split Vertex Deletion
Input: An undirected graph G.
Question: Find a minimum set S of vertices of G such that $G - S$ is a split graph.

B   Running Times of Exponential Approximation Algorithms

We provide extensive data sets on the running times for the obtained exponential approximation algorothms for the problems listed in Section 3.1. More precisely, we provide data sets fr the problems FVS, Tournament FVS, Subset FVS, 4-Hitting Set, Odd Cycle Transversal, Interval Vertex Deletion, Proper Interval Vertex Deletion, Block Graph Vertex Deletion, Cluster Graph Vertex Deletion, Cograph Vertex Deletion, Split Vertex Deletion, Edge Multicut on Trees and Partial Vertex Cover. Tables 6 and 7 contain the running times for selected approximation ratios, and graphical visualizations can be found in Figures 3 to 5

Feedback Vertex Set

(α, c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta, 2.69998)$	1.465	1.3861	1.3331	1.294	1.2637	1.2393	1.2193	1.2024	1.188
$(1.0, 2.69998)$	1.4156	1.3289	1.2753	1.2378	1.2099	1.1881	1.1706	1.1561	1.144
$(2.0, 1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298
combined	1.4156	1.3289	1.2753	1.2378	1.2068	1.1507	1.1037	1.064	1.0298

Tournament Feedback Vertex Set

(α, c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta, 1.618)$	1.2912	1.2463	1.2152	1.1918	1.1734	1.1583	1.1458	1.1352	1.126
$(1.0, 1.618)$	1.2348	1.1837	1.1531	1.132	1.1164	1.1042	1.0945	1.0865	1.0798
$(2.0, 1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298
combined	1.2348	1.1837	1.1531	1.132	1.1164	1.1042	1.0945	1.064	1.0298

Subset Feedback Vertex Set

(α, c)	1.1	1.8	2.5	3.2	3.9	4.6	5.3	6.0	6.7	7.4
$(\beta, 4.0)$	1.5474	1.2323	1.1507	1.1118	1.0889	1.0738	1.0631	1.0552	1.049	1.044
$(1.0, 4.0)$	1.5098	1.1952	1.1235	1.0906	1.0716	1.0592	1.0505	1.044	1.039	1.035
$(8.0, 1.0)$	1.7153	1.2891	1.1787	1.1225	1.0871	1.0623	1.0436	1.029	1.0171	1.0073
combined	1.5098	1.1952	1.1235	1.0906	1.0716	1.0592	1.0436	1.029	1.0171	1.0073

4-Hitting Set

(α, c)	1.1	1.4	1.7	2.0	2.3	2.6	2.9	3.2	3.5	3.8
$(\beta, 3.0755)$	1.4961	1.3117	1.2318	1.1853	1.1546	1.1327	1.1163	1.1035	1.0933	1.0849
$(1.0, 3.0755)$	1.4506	1.2592	1.1861	1.1459	1.1202	1.1023	1.0891	1.0789	1.0708	1.0643
$(4.0, 1.0)$	1.7147	1.4208	1.2882	1.2072	1.151	1.1093	1.0768	1.0506	1.029	1.0107
combined	1.4506	1.2592	1.1861	1.1459	1.1202	1.1023	1.0768	1.0506	1.029	1.0107

Odd Cycle Transversal

(α, c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta, 2.3146)$	1.4223	1.3521	1.3046	1.2695	1.2421	1.22	1.2018	1.1864	1.1733
$(1.0, 2.3146)$	1.3689	1.2908	1.243	1.2098	1.185	1.1658	1.1503	1.1376	1.1269

Interval Vertex Deletion

(α, c)	1.1	1.8	2.5	3.2	3.9	4.6	5.3	6.0	6.7	7.4
$(\beta, 8.0)$	1.6319	1.262	1.1688	1.1248	1.0991	1.0822	1.0702	1.0613	1.0544	1.0489
$(1.0, 8.0)$	1.6111	1.2401	1.1526	1.1121	1.0886	1.0733	1.0625	1.0545	1.0483	1.0434
$(8.0, 1.0)$	1.7153	1.2891	1.1787	1.1225	1.0871	1.0623	1.0436	1.029	1.0171	1.0073
combined	1.6111	1.2401	1.1526	1.1121	1.0871	1.0623	1.0436	1.029	1.0171	1.0073

Table 6: An entry at in row (α, c) and column β is best(α, c, β).

Proper Interval Vertex Deletion

(α,c)	1.1	1.6	2.1	2.6	3.1	3.6	4.1	4.6	5.1	5.6
$(\beta,6.0)$	1.6038	1.3001	1.204	1.1551	1.1252	1.105	1.0905	1.0795	1.0708	1.0639
$(1.0,6.0)$	1.577	1.2698	1.1799	1.1354	1.1087	1.0908	1.078	1.0684	1.0609	1.0549
$(6.0,1.0)$	1.7153	1.3436	1.2215	1.1546	1.111	1.0797	1.0559	1.0371	1.0217	1.0089
combined	1.577	1.2698	1.1799	1.1354	1.1087	1.0797	1.0559	1.0371	1.0217	1.0089

Block Graph Vertex Deletion

(α,c)	1.1	1.4	1.7	2.0	2.3	2.6	2.9	3.2	3.5	3.8
$(\beta,4.0)$	1.5474	1.3406	1.2521	1.201	1.1674	1.1435	1.1257	1.1118	1.1007	1.0916
$(1.0,4.0)$	1.5098	1.2961	1.2131	1.1672	1.1379	1.1174	1.1022	1.0906	1.0813	1.0738
$(4.0,1.0)$	1.7147	1.4208	1.2882	1.2072	1.151	1.1093	1.0768	1.0506	1.029	1.0107
combined	1.5098	1.2961	1.2131	1.1672	1.1379	1.1093	1.0768	1.0506	1.029	1.0107

Cluster Graph Vertex Deletion

(α,c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta,1.9102)$	1.3584	1.3007	1.2614	1.232	1.209	1.1904	1.1749	1.1619	1.1507
$(1.0,1.9102)$	1.3015	1.2367	1.1974	1.1703	1.1501	1.1345	1.1219	1.1116	1.1029
$(2.0,1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298
combined	1.3015	1.2367	1.1974	1.1703	1.1501	1.1345	1.1037	1.064	1.0298

Split Vertex Deletion

(α,c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta,2.0)$	1.3749	1.314	1.2726	1.2418	1.2176	1.1981	1.1819	1.1683	1.1566
$(1.0,2.0)$	1.3186	1.2503	1.2089	1.1802	1.1589	1.1423	1.129	1.1181	1.1089
$(2.0001,1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1038	1.064	1.0298
combined	1.3186	1.2503	1.2089	1.1802	1.1589	1.1423	1.1038	1.064	1.0298

Multicut on Trees

(α,c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(\beta,1.5538)$	1.2729	1.2313	1.2025	1.1807	1.1634	1.1494	1.1376	1.1277	1.1191
$(1.0,1.5538)$	1.2173	1.1699	1.1416	1.1221	1.1076	1.0964	1.0874	1.08	1.0738
$(2.0,1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298
combined	1.2173	1.1699	1.1416	1.1221	1.1076	1.0964	1.0874	1.064	1.0298

Partial Vertex Cover

(α,c)	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
$(2.0,1.0)$	1.6588	1.4847	1.3657	1.2768	1.2072	1.1507	1.1037	1.064	1.0298

Table 7: An entry at in row (α,c) and column β is `best`(α,c,β).
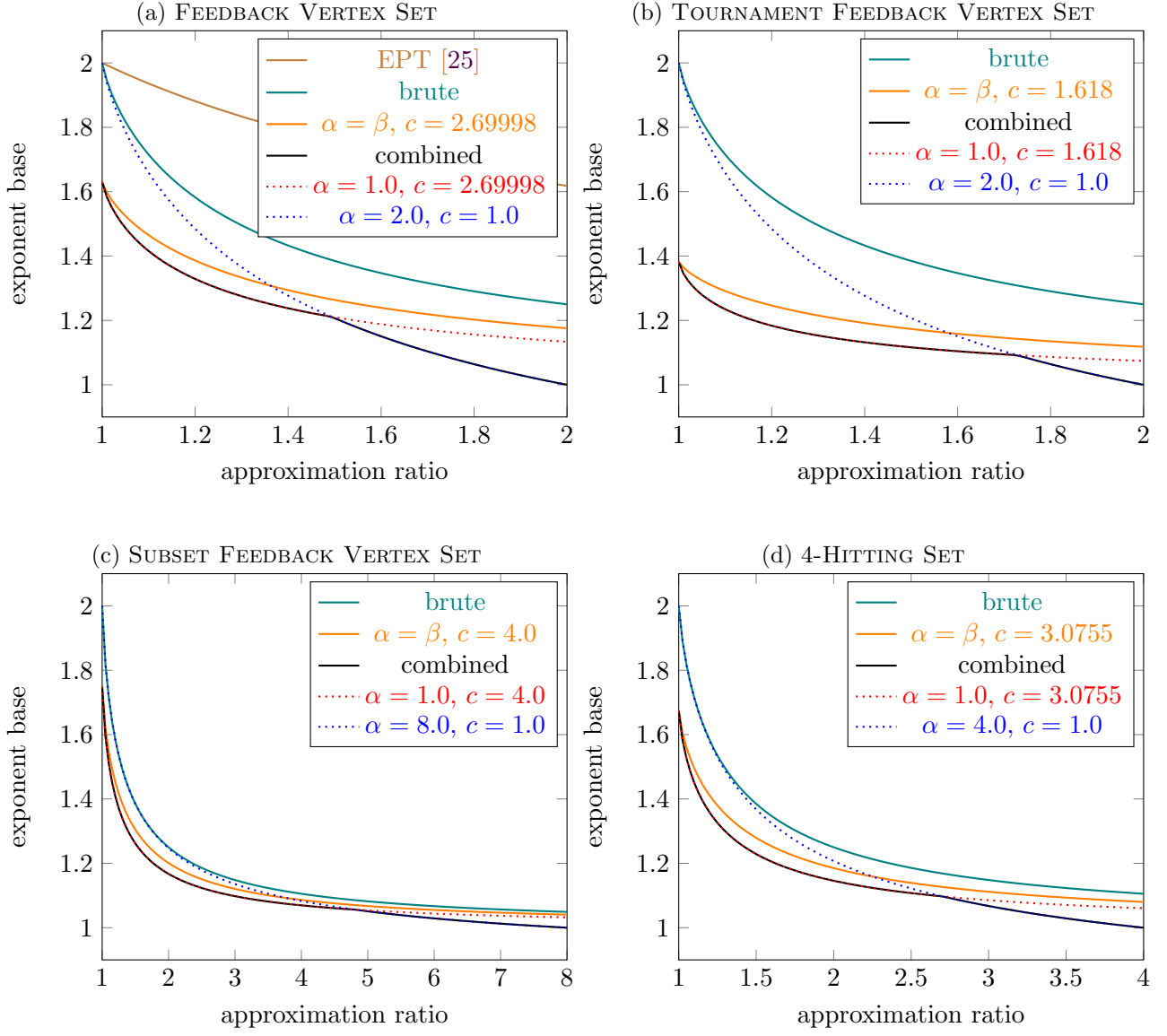
Figure 3: Results for FEEDBACK VERTEX SET, TOURNAMENT FEEDBACK VERTEX SET, SUBSET FEEDBACK VERTEX SET and 4-HITTING SET. A dot at (β, d) means that the respective algorithm outputs an β-approximation in time $\mathcal{O}^*(d^n)$.
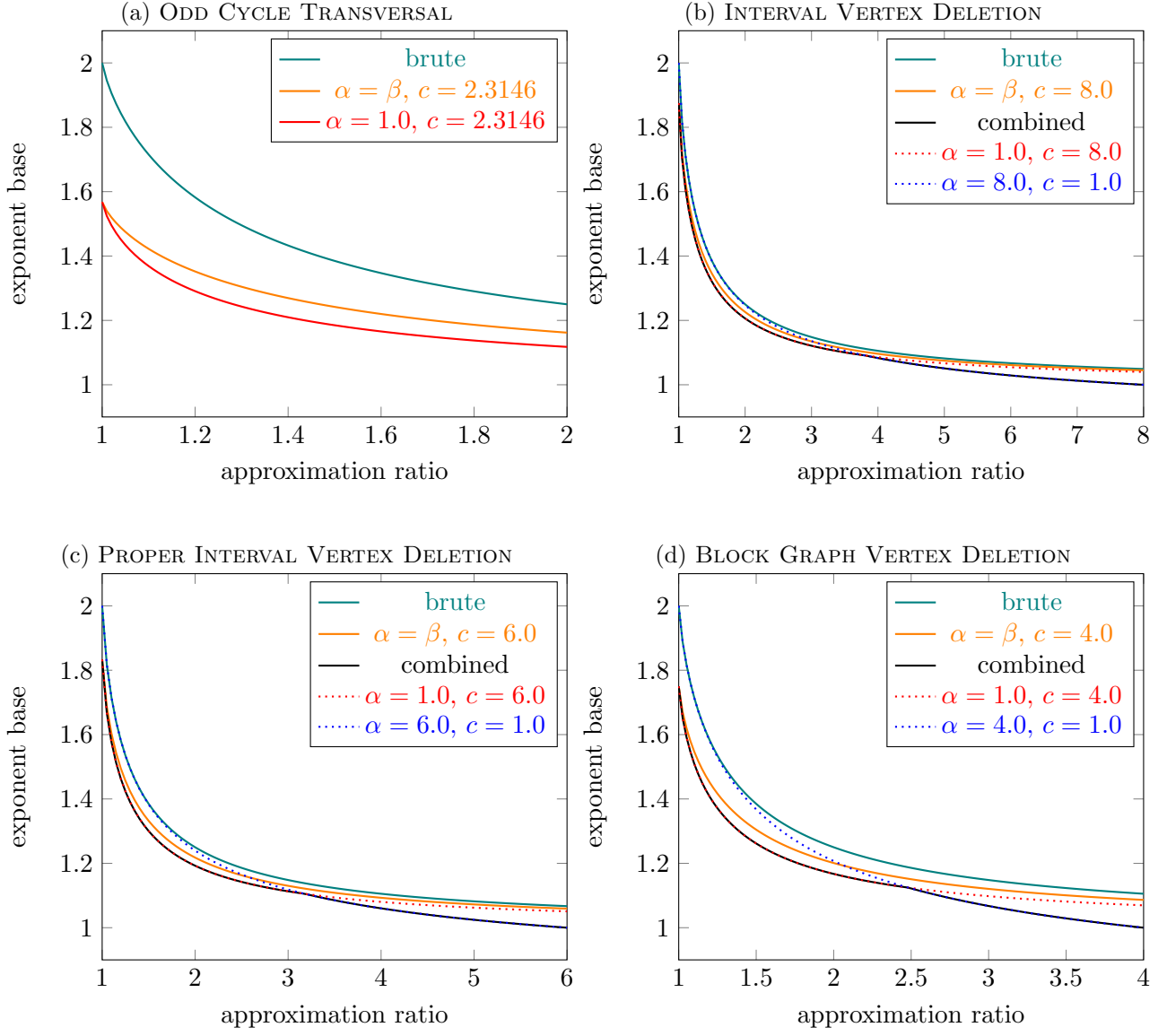
Figure 4: Results for ODD CYCLE TRANSVERSAL, INTERVAL VERTEX DELETION, PROPER INTERVAL VERTEX DELETION and BLOCK GRAPH VERTEX DELETION. A dot at (β, d) means that the respective algorithm outputs an β-approximation in time $\mathcal{O}^*(d^n)$.
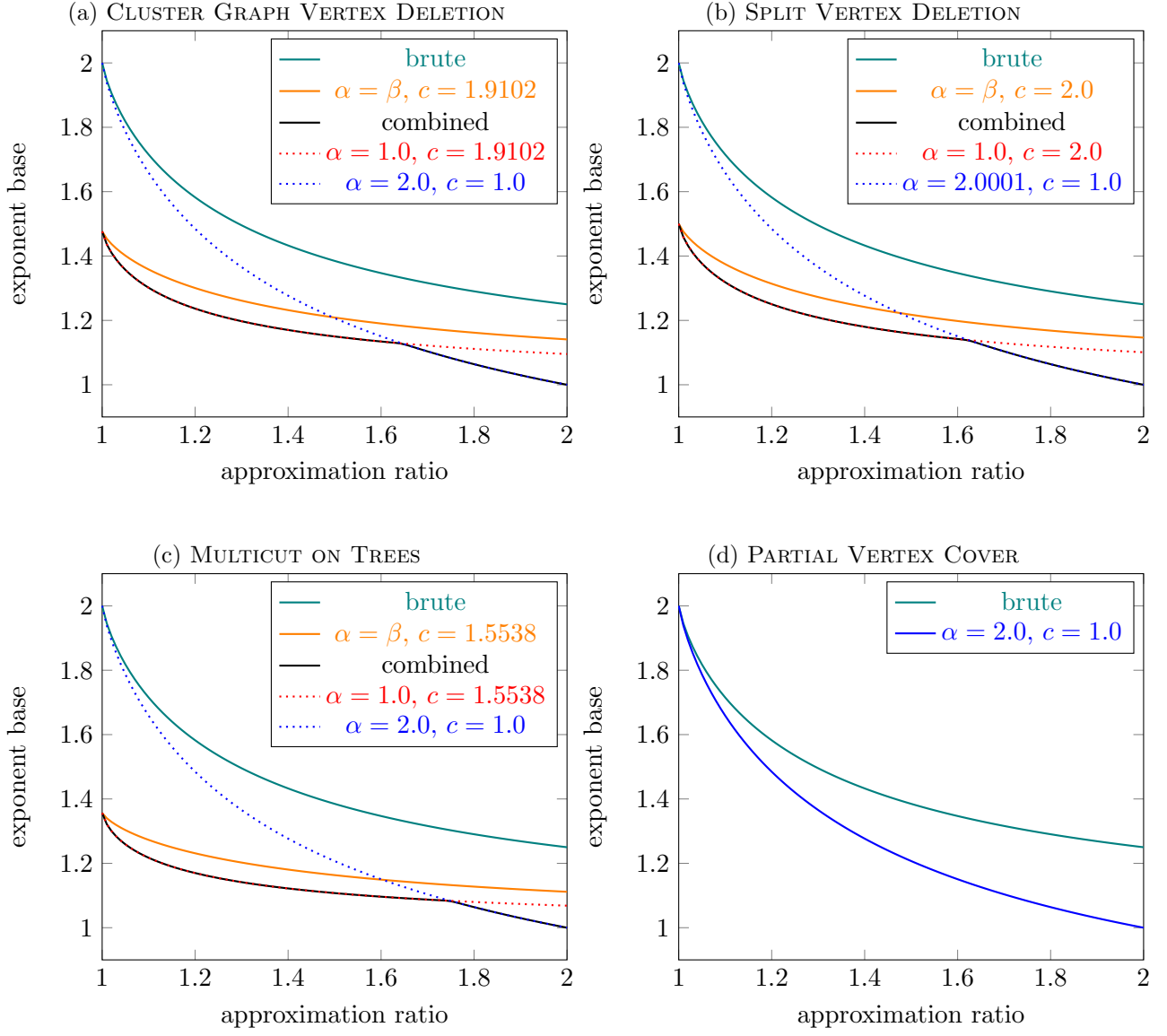
Figure 5: Results for CLUSTER GRAPH VERTEX DELETION, SPLIT VERTEX DELETION, MULTICUT ON TREES and PARTIAL VERTEX COVER. A dot at (β, d) means that the respective algorithm outputs an β-approximation in time $\mathcal{O}^*(d^n)$.