Optimal Online Frequency Capping Allocation using the Weight Approach

Martin Zinkevich

October 29, 2010

Abstract

In this document, we present a simple approach for serving guaranteed delivery contracts with frequency capping which we call the weight method, which is optimal when targeting is user-dependent, as well as when there is no frequency caps. This methodology can also serve as a heuristic for more general problems. The method is consistent, in that the booking and planning methods work by simulating the ad server, and therefore, modulo issues of forecasting, the system will be consistent.

1 Introduction

Suppose that you have a website where a million users visit daily. Some visit once, some visit twenty times. In general, it is difficult to determine who will visit how many times. However, predicting what fraction of users will visit a certain number of times in a time period is easier. In this paper, we come up with algorithms where, given an accurate prediction of aggregate behavior, can come up with an optimal plan for advertising, even in the presence of frequency capping.

In online advertising, publishers sell opportunities to place advertisements on their websites to advertisers. Usually this is done in the context of a contract. For a non-guaranteed contract, the advertiser might pay on a per-impression basis, or it may pay each time a user clicks on an advertisement, or for some other related event. In this paper, we focus on guaranteed delivery, where the advisor pays in advance for a certain number of impressions.

There are several issues that publishers must deal with to sell guaranteed delivery contracts. The first is that they must predict the number of opportunities they will have to sell in the future (**forecasting**). The second is that they must determine the value of these opportunities and the prices for contracts (**booking**). Finally, they must match the opportunities to advertisers when the opportunities arrive (**serving**).

Guaranteed delivery is made more complicated by two factors. The first is that there is an inherent stochasticity to the contracts, in that both parties realize that there is a chance that the contract may not be satisfied, mostly due to the difficulties in forecasting. We do not deal with this issue in this paper, but mention some open problems in the conclusion. The second is that not all advertisers are interested in all opportunities.

1. **User Segment targeting:** They may be interested only in certain users (selling razors to men, spring break vacations to college students, et cetera).
2. **Property targeting:** They may only want to advertise next to a subset of the user's publications (advertising TMZ next to the gossip column, football tickets near a sports column).
3. **Frequency capping:** They may wish to limit the exposure of a user to an advertisement (show a user an advertisement no more than three times a day).

There has been a lot of research into serving impressions with property targeting [FIMN08, VVS10, DCMCS10], and some research into frequency capping [Far09], but there is no universal method that addresses both. In this paper, we explain a method that can address either optimally, and heuristically, it may address both.

The basic idea revolves around assigning each campaign a **weight**. Whenever an opportunity to serve an ad arises, we first determine which campaigns are interested (the opportunity meets any user segment, property, and frequency capping constraints the campaign has) in the advertisement. Then, we randomly select an advertisement to show, with the probability of selecting each advertisement being proportional to the weight of the advertisement. Note that unlike in [Far09], by definition we never violate frequency capping conditions.[1] However, this only describes the serving algorithm. There are several other aspects.

1. WeightedBooking: An algorithm for deciding how many impressions a new contract can be given, and at what price.
2. WeightedPlanning: An algorithm for deciding how to determine the weights for contracts, modifying them if necessary.
3. WeightedServing: An algorithm for choosing which contract to match an opportunity to, live.

WeightedServing has been described in loose terms above. However, by design, it is not hard to understand or implement. What is more difficult is deciding on the weights, given a specific booking. We present an algorithm below that can not only guarantee ϵ-optimal allocations in a variety of scenarios, but the allocations can be obtained efficiently. WeightedPlanning is also easy to implement, though it is nontrivial to prove it is correct. For each contract $c \in C$, define w_c to be its weight, and b_c to be the number of booked impressions.

Algorithm 1 A method to find contract weights

Require: A simulator, and padding $\epsilon > 0$
Ensure: The weights, w_c, for all contracts c are set appropriately
  Initialize $w_c \leftarrow 1$ for all contracts c
  Simulate serving to get e_c for all $c \in C$, where e_c is an estimate of how many impressions are obtained by contract c given the weights w
  while $e_c < b_c$ for some c **do**
    // Update the weights
    for contract $c \in C$ where $e_c \leq b_c$ **do**
      $w_c \leftarrow w_c \times (1+\epsilon) w_c b_c / e_c$
    end for
    Simulate serving to get e_c for all $c \in C$, where e_c is an estimate of how many impressions are obtained by contract c given the weights w
  end while

There are a variety of ways to design a WeightedBooking algorithm. The best approach is to run a traditional linear programming technique [YT08], increasing the requirements on each contract by a small multiplicative factor (see Section 5). There is some literature on solving the problem without frequency capping using linear programming. In Section 5.3, we discuss the linear program when there are frequency cap constraints.

In cases where a linear programming cannot represent all the constraints, we can try to get a rough idea of the maximum amount that can be booked, and then confirm the feasibility using the WeightedPlanning algorithm directly.

The greatest thing about this method is that, when there exists a linear programming technique for booking, it is perfectly synchronized. The planning algorithm simulates the serving, and the booking provides guarantees for the planning. Therefore, barring errors in forecasting, the contracts will get the right number of impressions. The second thing that is wonderful is that this method is guaranteed to work on a wider variety of contracts than systems developed in the past: in particular, it can handle daily frequency capping

[1] Of course, this does assume we can maintain counts for each user and contract pair. If we have multiple HTTP servers that are geographically disparate, this is nontrivial and may lead to a few violations if users do not always connect to the same machine.

and user segment targeting contracts so long as there is no property targeting, and it can handle user segment targeting and property targeting contracts so long as there is no frequency capping. Next, we go through some examples that show the power of the technique. Then, we go into some theoretical work that demonstrates that the algorithm works. Finally, we conclude.

2 Examples of WeightedServing and WeightedPlanning

Let us begin with a few examples.

Example 1 *There are one million opportunities, and two contracts. Contract A requires 900,000 impressions, and Contract B requires 50,000 impressions.*

To correctly serve these impressions, $9 \leq w_A/w_B \leq 19$.

As we run WeightedPlanning, we set $\epsilon = 0.01$, then initially set $w_A = w_B = 1$, so we estimate contract A gets 500,000 impressions, and contract B gets 500,000 impressions. After one iteration, $w_A = 1.818$, $w_B = 1$. Now, A gets 645,000 impressions, and B gets 465,000. Asymptotically, until $e_A \geq b_A$, w_A will increase by a factor of at least 1.01, and w_B will stay at 1. After a few iterations, there will be sufficient impressions for A. Note that even if we increased w_A only by a factor of 1.01, it would still eventually reach 9. However, it would take more iterations.

Example 2 *The are one million opportunities, and three contracts. Contract A requires 900,000 impressions, Contract B requires 90,000 impressions, and Contract C requires 9,000 impressions.*

As we run WeightedPlanning, we set $\epsilon = 0.005$, then first w_A will increase until it has a sufficient number of impressions. At some point, it will take too many impressions away from contract B, and w_B will increase as well. At iteration 132, the system stabilizes.

Example 3 *There are one million users visiting twice in a day, and three contracts with frequency cap of 1. Contract A requires 900,000 impressions, Contract B requires 600,000 impressions, and Contract C requires 300,000 impressions.*

Here, the same behavior occurs. Contract A increases its weight until $w_A/(w_B+w_C)$ is greater than (roughly) 9, and w_B increases its weight until w_B/w_C is greater than 2. Note that it is not enough that w_A is twice as large as $w_B + w_C$.

WeightedPlanning is based on a few intuitions.

1. There *is* a solution to the problem via weights.
2. On every iteration, some weight increases by a factor of at least $(1 + \epsilon)$.
3. As the weight of contract B increases, the number of impressions for contract A does not increase.
4. As the weight of contract A increases by a factor of k, the number of impressions increases by *less than* a factor of k.

These intuitions allow us to approach the correct weights "from below," and they hold if there is no frequency capping. They also hold if there is daily frequency capping, but no property targeting. However, there are counterexamples where it does not hold if there is frequency capping and property targeting. The fourth constraint is so that we can accelerate the process when we are severely underdelivering. We revisit this fourth constraint later to produce a faster (if slightly more complicated) algorithm.

There are examples where these constraints do not hold.

Example 4 *There are one million users visiting twice in a day, once on the home page and once on a sports article. Contract A requires 900,000 impressions from anywhere, Contract B requires 400,000 impressions from the home page, and Contract C requires 400,000 impressions from sports. Each contract has a frequency cap of 1.*

Here, increasing the weight of contract A can increase the impressions of contract C. Effectively, when you increase the weight of contract A such that $w_A > w_B$, then contract A gets over half of the home page impressions, such that when those users come to the sports page, they cannot see another impression from contract A. For example, if you have $w_A = w_B = w_C = 1$, then C gets 750,000 impressions. However, if you set $w_A = 10$, then C gets over 900,000 impressions.

Even in this example, the algorithm does find the solution. In particular, C always receives enough impressions, so its weight is fixed at one. On the other hand, the weights of w_A and w_B both increase until w_A gets almost half of the impressions in sports and on the home page.

Example 5 *There are one million users who visit twice a day. Half view a sports article, then a finance article. Half visit the home page, then an auto article. Contract A is targeting the home page, sports, and finance. Contract B is targeting the home page, sports, and auto. Each requires 900,000 impressions, and has a frequency cap of 1.*

In this example, ideally one would sell all home page and sports opportunities to contract A, and sell all finance and auto to contract B. Note that this cannot be accomplished with weights.

The problem is that past experience can be used to predict future behavior. However, even if visits on pages were independent, you would still run into problems.

Example 6 *People randomly visit the home page, a sports article, or a finance article, with a 1/3rd probability of leaving after every visit. Contract A is targeting the sports article and the home page, and Contract B is targeting the finance article and the home page. Each is interested in an equal number of impressions and has a frequency cap of 2.*

In this case, if you a user has seen one ad from Contract A and is visiting the home page, you should show him an ad from Contract B. Similarly, if a user has seen one ad from Contract B and is visiting the home page, you should show him an ad from Contract A. This relationship is impossible to capture with weights.

Thus, there are technical limitations to the number of impressions that can be obtained using weight allocations. These are not limited to weights, but any time that the distribution over the ad shown is a monotonic function of the values of the applicable ads. Extending the technique to resolve these problems is an interesting area of future work.

As a final point, there is no known example where there exists a set of weights that solve a problem (and there is some slack), and yet the **WeightPlanning** algorithm does not find them.

3 Formalisms

The most complex element of this algorithm is the model we use for predicting the expected number of impressions given a set of weights. For simplicity, we assume $C = \{1 \ldots n\}$. Formally, a model is a function $A : (\mathbf{R}^+)^n \to (\mathbf{R}^+)^n$.

Definition 7 *A model A is **well-behaved** if for any $v \in (\mathbf{R}^+)^n$, and any $\alpha \geq 1$, any $i \in \{1, \ldots, n\}$, and $w \in (\mathbf{R}^+)^n$ where $w_j = v_j$ if $j \neq i$ and $w_i = \alpha v_i$:*

1. *for all $j \neq i$, $A_j(w) \leq A_j(v)$,*
2. *$A_i(v) \leq A_i(w)$,*
3. *$A_i(w) \leq \alpha A_j(v)$,*
4. *$A_i(\alpha v) = A_i(v)$,*

For any $x, y \in \mathbf{R}^n$, $x \geq y$ if and only if for all i, $x_i \geq y_i$.

Definition 8 *Given weights $w \in (\mathbf{R}^+)^n$, a booking b_i for each contract, such that $b \in (\mathbf{R}^+)^n$, and a model A, if $A(w) \geq b$, then* ***w satisfies b****. b is* ***satisfiable with weights*** *if there exists an $w' \in (\mathbf{R}^+)^n$ such that w' satisfies b. If $(1+\epsilon)b$ is satisfiable, then b is* ***ϵ-oversatisfiable with weights****.*

Note, that the above definition explicitly states a solution in terms of weights. In later parts, we connect satisfiability in terms of weights to satisfiability in terms of Hall's Theorem.

The main theorem we state below:

Theorem 9 *Given a well-behaved model A and a booking b which is ϵ-oversatisfiable with weights, if the algorithm runs with $\epsilon' < \epsilon$, the algorithm terminates and returns a w satisfying b.*

Proof: First, since b is over-satisfiable, there exists a $w' \in (\mathbf{R}^+)^n$ such that $A(w') \geq b(1+\epsilon)$. Next, we scale the vector such that the lowest element is one, i.e.we define $\overline{w} = (\min_i w'_i)^{-1} w'$. Because A is well-behaved (4), $A(\overline{w}) = A(w') \geq b(1+\epsilon)$. The rest of the proof is in two parts: in the first part we establish recursively that $w \leq \overline{w}$ throughout the algorithm. Define $\Phi(w) = \frac{1}{\log(1+\epsilon)} \sum_{i=1}^n \log \overline{w}_i - \log w_i$. Note that by the first part, $\Phi(w) \geq 0$. In the second part we will show that Φ decreases by at least one in every iteration.

Part 1: $w \leq \overline{w}$: Observe that $w \leq \overline{w}$ initially, because by definition $\min_i \overline{w}_i = 1$. Consider a single iteration of the algorithm. Define w^{initial} to be the initial value of w during the iteration, and w^{final} to be the final value of w. If for some i, $w_i^{\text{final}} \neq w_i^{\text{initial}}$, then $w_i^{\text{final}} = w_i^{\text{initial}}(1+\epsilon')b_i/e_i$. Define w^{hybrid} such that $w_i^{\text{hybrid}} = w_i^{\text{final}}$ and for all $j \neq i$, $w_j^{\text{hybrid}} = w_j^{\text{final}}$. Because A is well behaved (3), $A_i(w^{\text{hybrid}}) \leq (1+\epsilon')(b_i/e_i)A_i(w^{\text{initial}})$. Since $e_i = A_i(w^{\text{initial}})$, $A_i(w^{\text{hybrid}}) \leq (1+\epsilon')b_i$. Define $w^{\text{hybrid2}} \in (\mathbf{R}^+)^n$ such that $w_i^{\text{hybrid2}} = w_i^{\text{hybrid}}$ and for all $j \neq i$, $w_j^{\text{hybrid2}} = \overline{w}_j$. Because A is well-behaved (1),

$$A_i(w^{\text{hybrid2}}) \leq A_i(w^{\text{hybrid}}) \leq (1+\epsilon')b_i < (1+\epsilon)b_i \leq A_i(\overline{w}). \tag{1}$$

Note that if $w_i^{\text{hybrid2}} \geq \overline{w}_i$, then $w^{\text{hybrid}} \geq \overline{w}$, and because A is well-behaved (2), $A_i(w^{\text{hybrid2}}) \geq A_i(\overline{w})$ (a contradiction to Equation 1). So $\overline{w}_i \leq w_i^{\text{hybrid2}} = w_i^{\text{hybrid}} = w_i^{\text{final}}$, so by induction, $w^{\text{final}} \leq \overline{w}$.

Part 2: $\Phi(w)$ decreases by at least 1 every iteraton: If the algorithm does not terminate, then there is at least one $i \in \{1, \ldots, n\}$ where w_i is changed. If w_i is increased, then $e_i < b_i$, implying $w_i^{\text{final}} = w_i^{\text{initial}}(1+\epsilon)b_i/e_i \geq (1+\epsilon)w^{\text{initial}}$. Also, note that $w^{\text{initial}} \leq w^{\text{final}}$. Thus,

$$\Phi(w^{\text{initial}}) - \Phi(w^{\text{final}}) = \frac{1}{\log(1+\epsilon)} \sum_{j=1}^n \log w_j^{\text{final}} - \log w_j^{\text{initial}} \tag{2}$$

$$\Phi(w^{\text{initial}}) - \Phi(w^{\text{final}}) \geq \frac{1}{\log(1+\epsilon)} \log w_i^{\text{final}} - \log w_i^{\text{initial}} \tag{3}$$

$$\Phi(w^{\text{initial}}) - \Phi(w^{\text{final}}) \geq \frac{1}{\log(1+\epsilon)} \log(w_i^{\text{final}}/w_i^{\text{initial}}) \tag{4}$$

$$\Phi(w^{\text{initial}}) - \Phi(w^{\text{final}}) \geq \frac{1}{\log(1+\epsilon)} \log(1+\epsilon) \tag{5}$$

$$\Phi(w^{\text{initial}}) - \Phi(w^{\text{final}}) \geq 1 \tag{6}$$

Thus, on every step, Φ decreases by 1. Since from Part 1, $w < \overline{w}$, $\Phi(w) \geq 0$, so after $\Phi(1)$ steps, the algorithm must converge. By substitution, this is $\lfloor \sum_i \log_{1+\epsilon} \overline{w}_i \rfloor$ steps. ■

Therefore, for certain constraints on the model, the algorithm terminates. Moreover, the number of steps required is logarithmic in the magnitude of the weights. What we need to establish is that many models of interest (e.g., property targeting without frequency capping, frequency capping without property targeting), are well-behaved.

However, before diving into all the formalisms of guaranteed delivery, we present a very simple, very powerful lemma that allows us to decompose seemingly intractable problems into simple ones. First, we present the traditional definition of addition in function space.

Definition 10 *Given $A, A_1, A_2 : (\mathbf{R}^+)^n \to (\mathbf{R}^+)^n$ (such that they are functions from $(\mathbf{R}^+)^n$ to $(\mathbf{R}^+)^n$), if for all $w \in (\mathbf{R}^+)^n$, $A(w) = A_1(w) + A_2(w)$, we say that $A = A_1 + A_2$.*

Theorem 11 *If $A = A_1 + A_2$, and A_1 and A_2 are well-behaved, then A is well-behaved.*

Corollary 12 *If $A = \sum_{i=1}^n A_i$, and for all i, A_i is well-behaved, then A is well-behaved.*

Proof: The theorem follows directly from the definition of well-behaved, and the corollary follows from recursion. ∎

4 Models of Serving

A model of opportunities can be characterized by several parameters. In this work, we consider only daily frequency caps. First, we have a set of opportunities O. Each opportunity occurs on a given day and is from a given user. For every user u and day d, define $F^{u,d}$ to be the sequence of opportunities arriving for user u at day d. Then, a contract c can be represented by a set of matching opportunities ($M_c \subseteq O$), a target number of impressions (b_c), and a frequency cap (f_c) which may be infinite.

4.1 Modeling without Frequency Caps

Suppose that there are no frequency caps, or more formally, all the frequency caps are infinite. Then critically, what contract is served in opportunity o_1 is independent of what contract is served in opportunity o_2. Thus, for all $o \in O$ we can define A^o, which is the expected number (i.e.probability) of impressions served from opportunity o to each contract. For simplicity, define $C_o \subseteq C$ to be the set of contracts c where $o \in M_c$. Therefore:

$$A^o_c(w) = \begin{cases} \frac{w_c}{\sum_{c' \in C_o} w_{c'}} & \text{if } c \in C_o \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

In particular, the probability of selecting a matching contract is proportional to its weight. If there is no frequency capping, then $A^{O,C} = \sum_{o \in O} A^o$.

Lemma 13 *A^o (from Equation 7) is well-behaved.*

Proof: Observe that for a given $c, c' \in C$ where $c \neq c'$, if $w_{c'}$ increases, then A^o_c decreases or stays the same (well-behaved property 1), because the denumerator increases or stays the same. If w_c increases, then A^o_c increases or stays the same (well-behaved property 3), but because the denominator increases as well, the allocation increases by a factor smaller than the weight (well-behaved property 2). Since the weights occur in the numerator and the denominator, scaling them changes nothing (well-behaved property 4). Thus, A^o is well-behaved. ∎

Theorem 14 *If there is no frequency capping, $A^{O,C}$ is well-behaved.*

Proof: This follows from Lemma 13 and Theorem 11. ∎

4.2 Modeling With Frequency Caps

For each user-day pair (u, d), there is a expected number of opportunities for each contract $A^{u,d} : (\mathbf{R}^+)^n \to (\mathbf{R}^+)^n$ from that user-day pair. We show in the appendix how we could exactly calculate this value. For the general model, $A(w) = \sum_{u,d} A^{u,d}(w)$. In other words, across users the function is additive.

Definition 15 *Given the $F^{u,d}$ and M_c that underly model A, if for every user-day pair (u, d), for every contract $c \in C$, either $F^{u,d} \subseteq M_c$ or $F^{u,d} \cap M_c = \emptyset$, then the model A has **no property targeting**. In other words, a contract either matches every opportunity from a user or none of them.*

If there is no property targeting, then we can represent $A^{u,d}(w)$ as a function of:

1. the length of $F^{u,d}$,
2. the number of matching contracts to an element of $F^{u,d}$, and
3. the frequency cap of each matching contract to $F^{u,d}$.
4. the weights of the matching contracts.

Define a^k to be this function, where k is the length of $F^{u,d}$. If we simply put a zero for the frequency cap of a contract that does not match, and define $\mathcal{F} = \{0, 1, 2 \ldots\}$ to be the non-negative integers, we can write $a^k : (\mathbf{R}^+)^n \times \mathcal{F}^n \to (\mathbf{R}^+)^n$. Formally, there exists k, f, such that for all $w \in (\mathbf{R}^+)^n$, $A^{u,d}(w) = a_k(w, f)$. This derivation is covered in detail in the appendix.

Lemma 16 *If for every $c \in C$, either $F^{u,d} \subseteq M_c$ or $F^{u,d} \cap M_c = \emptyset$, then $A^{u,d}$ is well-behaved.*

Proof Sketch: We have to prove that a_c^k is well-behaved. First, since weights are always multiplicatively normalized to determine allocation, scaling does not impact allocation.

With clever accounting one can prove that increasing the weight of c does not decrease the number of impressions to c. The key insight is that decreasing the frequency cap of c by 1 and increasing the frequency cap of c' by one does not affect the number of impressions to c by more than 1 per user. Thus, recursively one can argue that serving a c instead of a c' at the first opportunity is not overshadowed by the consequences when the user continues to visit with the current number of impressions remaining modified.

A similar recursive argument can be used to argue that the number of impressions to c is not *too* high after the weight of c has been adjusted. This is another recursive argument, leveraging the fact that decreasing the frequency cap of c and increasing the frequency cap of c' does not increase the number of impressions to c.

Finally, we must demonstrate that increasing the weight of c does not increase the number of impressions to c'. This is the trickiest proof, because that while increasing the weight of c implies that c' will get fewer impressions at the first opportunity, it also implies that c'' will receive fewer impressions. Consider the case where c has a weight of 9 and c' and c'' have weights of 1, each contract has a frequency cap of 1, and the user visits twice. Then, if we just tweaked the first impression, giving it to c instead of c'', then this would actually increase the probability that c will get the second impression from 10% to 90%. However, we can prove that this increase is offset by decreases elsewhere, by providing a direct mapping between opportunities after c gets the first impression and opportunities after c does not get the first impression. This is further complicated when the frequency cap is higher, as one must map the opportunities when c gets the first n impressions back to opportunities when c does not get the first impression.

This is proven in detail in the appendix. ■

5 Hall's Theorem

Hall's Theorem is often applied to the marriage problem (finding a set of marriages between men and women matching certain constraints), which can be extended to an arbitrary allocation problem. What we show here is that it also can be applied to frequency capped contracts.

5.1 Hall's Theorem without Frequency Capping

For each contract $c \in C$, there is an amount booked b_c. For a subset $C' \subseteq C$, define $b_{C'} = \sum_{c \in C'} b_c$. This is the cumulative booked for C'. Without frequency capping, we can also define $s_{C'} = |\bigcup_{c \in C'} M_c|$, to be the supply available to C'. Notice that if M_c and $M_{c'}$ intersect, then $s_{\{c,c'\}} \leq s_c + s_{c'}$.

We will now define **Hall satisfiability** to be the property that there exists $M'_c \subseteq M_c$ such that for all $c, c' \in C$, $M'_c \cap M_{c'} = \emptyset$ and $|M_c| \geq b_c$. In other words, there exists some way to serve opportunities to contracts, which may not use weights.

We define a problem to be **Hall ϵ-oversatisfiable** if in addition to the above constraints, there exists a solution such that $|O_c| \geq (1+\epsilon)b_c$.

Theorem 17 ***Hall's Theorem****A problem without frequency capping is Hall satisfiable (if and) only if for all $C' \subseteq C$, $b_{C'} \leq s_{C'}$.*

We parenthesize the if part, because it is the more difficult part to prove and is not required for our purposes here. It is the more obvious part of Hall's Theorem that is necessary. Define $R_{C'}(w) = \frac{\min_{c \in C'} w_c}{\max_{c \in C \setminus C'} w_c}$ to describe how much larger the weights in C' are than $C \setminus C'$.

Fact 18 *As $R_{C'}(w) \to \infty$, $\sum_{c \in C'} A_c^{O,C}(w,f) \to s_{C'}$. In other words, as the weights of the contracts in C' grows with respect to the contracts in $C \setminus C'$, the contracts in C' will grab almost all the inventory available to them.*

By virtue of its stopping condition, Algorithm 1 is guaranteed to produce an allocation plan which ensures every contract meets its demand (in expectation), so long as the algorithm actually stops. In fact, we can show that it is guaranteed to converge, so long as the demands of all contracts (padded by $(1+2\epsilon')$) are feasible. In practice, we can trim the demand of contracts somewhat to ensure feasibility. We sketch the proof below.

Theorem 19 *Without frequency capping, if for some $\epsilon > 0$, a problem is Hall-ϵ oversatisfiable, it is satisfiable.*

Proof: We define $\epsilon' = \epsilon/2$. Denote the value of the weights in the t-iteration by $w^{(t)}$. First, suppose that at any point during the algorithm that $A_j^{O,C}(w^{(t)}) \leq b_j(1+\epsilon')$. Then we claim that $A_j^{O,C}(w^{(t')}) \leq b_j(1+\epsilon')$ for all $t' \geq t$, by an argument similar to that of Part 1 of the proof of Lemma 6.

Now, suppose that the algorithm never halts. For each j, the sequence $w_j^{(t)}$ for $t = 1, 2, ...$ is increasing (by the update rule of the algorithm). Hence, if it is bounded above, it must converge. Let C' be the set of j such that the sequence is unbounded. Since the problem is feasible with demands of $(1+2\epsilon')b_j$, and by Hall's Theorem,

$$s_{C'} \geq (1+2\epsilon')b_{C'}$$

Now, since the masses for $j \in C'$ increase without bound, by Fact 18 there is a point, say $t = T$, at which the total delivery for the contracts with unbounded masses will be all but an arbitrarily small fraction of the total inventory available. Hence, there is some $j \in C'$ for which

$$A_j^{O,C}(w^{(T)}) > (1+\epsilon')d_j$$

But this contradicts our previous claim— since all of the masses for $j \in C'$ are updated repeatedly, it must be the case that $A_j^{O,C} < d_j(1+\epsilon')$. Hence C' must be empty, and the algorithm converges. ∎

5.2 Hall's Theorem with Frequency Capping

With frequency capping, very odd properties can hold. In particular, even if $O_c = O_{c'}$, the supply available to $\{c, c'\}$ can actually be equal to the sum of c and c'. This happens if every user visits twice and $f_c = f_{c'} = 1$.

A problem is **Hall satisfiable with frequency capping** if there exists $M'_c \subseteq M_c$ where for any $c, c' \in C$ where $c \neq c'$, $M'_c \cap M'_{c'} = \emptyset$, for all $c \in C$, $|M'_c| \geq b_i$, and for all (u,d), for all c, $M'_c \cap F^{u,d} \leq f_c$. We define a problem to be **Hall ϵ-oversatisfiable with frequency capping** if in addition to the above constraints, there exists a solution such that $|O_c| \geq (1+\epsilon)b_c$.

Consider a set of contracts $C' \subseteq C$. For any user-day pair (u,d), define $C_{u,d} = \{c \in C : F^{u,d} \subseteq M_c\}$. Define:

$$s_{C'} = \sum_{u,d} \min\left(|F^{u,d}|, \sum_{c \in C' \cap C_{u,d}} f_c\right) \tag{8}$$

Therefore, for each user-day pair, if the frequency caps of matching contracts are high enough we can either serve every impression to a contract in C'. Alternatively, we are bounded by the sum of the frequency caps. We can create a Hall's Theorem for frequency capping.

Theorem 20 *A problem with frequency capping without property targeting is Hall satisfiable only if for all* $C' \subseteq C$, $b_{C'} \leq s_{C'}$.

Proof: The only if part is clear: if trying to serve as many opportunities as possible to C' failed to obtain $b_{C'}$ opportunities, then there is no way to serve b_c opportunities to each contract $c \in C'$ simultaneously. ■

While the if part of this Hall's Theorem is interesting, it goes beyond the scope of this paper.

Fact 21 *If there is no property targeting, as* $R_{C'}(w) \to \infty$, $\sum_{c\in C'} A(w,f) \to s_{C'}$. *In other words, as the weights of the contracts in* C' *grows with respect to the contracts in* $C \backslash C'$, *the contracts in* C' *will grab almost all the inventory available to them.*

This can be easily seen if one considers an individual contract.

Theorem 22 *With frequency capping but without property targeting, if for some* $\epsilon > 0$, *a problem is Hall-*ϵ *oversatisfiable, it is satisfiable.*

The proof follows the exact same lines as Theorem 19. Because the "only-if" part of Hall's Theorem is satisfied by Theorem 20, the planning algorithm is well-behaved by Theorem 16, and Fact 21, the result holds.

5.3 A Linear Program for Frequency Capping

In [Far09], they discuss how to formulate the problem as a dynamic program. While this approach can be effective, there is the issue that frequency capping constraints can be violated with some probability using the solution. Therefore, here we present a solution which in a restricted case, guarantees correct behavior. There are two possibilities. In one case, we have a distribution over the number users who make a certain number of visits per day. For each number of visits v, for each user segment s, define $q_{s,v}$ to be the number of users with that number of visits. Then, for each contract $c \in C$, for each number of visits v, for each segment s, we define $x_{c,v,s}$ to be the number of impressions from users with that number of visits. We formulate a linear program for this problem and solve for x:

$$\sum_{c \in C} x_{c,v,s} \leq q_{s,v} \tag{9}$$

$$x_{c,v,s} \leq f_c q_{s,v} \tag{10}$$

$$\sum_{v,s} x_{c,v,s} \geq b_c. \tag{11}$$

These constraints make certain each contract is satisfied. If c is the new contract and we maximize b_c, then we can see how many impressions we can book to contract c. In some cases, the number of user segments may become prohibitive, so we can sample them, similar to [VVS10].

6 System Guarantees

Thus, by showing the algorithm is satisfiable in the sense of weights with a small amount of slack, we can prove that the weights architecture works. First, we can book contracts by solving a constraint optimization problem. We insist that all the constraints are slack. Define $b' = (1+\epsilon)b$. If the problem with b' is Hall ϵ'-oversatisfiable, then b' is satisfiable with weights, so b is ϵ-oversatisfiable with weights.

Theorem 23 *With frequency capping but without property targeting, or with property targeting but without frequency capping, given a booking which is ϵ-oversatisfiable and accurate forecasting:*

1. *the* WeightedPlanning *algorithm can generate a plan.*
2. *the* WeightedServing *algorithm can for each contract deliver the right number of expected impressions.*
3. *the* WeightedBooking *algorithm determining how many impressions can be guaranteed with a certain ϵ-oversatisfiability.*

Proof: The WeightedPlanningalgorithm's properties are guaranteed mostly by Theorem 16, Theorem 22, and Theorem for frequency capping, and Theorem 14, Theorem 19, and Theorem 6 for property targeting. These imply that the plan (the weights) generated by WeightedPlanningwill if applied by WeightedServingin expectation deliver the right number of impressions to each contract. Finally, because WeightedPlanningcan handle any Hall ϵ-oversatisfiable booking, by ensuring that the Hall constraints are satisfied with inflated booked amounts (which can be done as in [VVS10, DCMCS10], or for frequency capped contracts as described in Section 5.3, we can guarantee that the overall system will work. ■

We discuss forecasting and the actual number of impressions served in future work.

7 Heuristic Improvements

Although it is difficult to prove theoretically, intuitively, more aggressive increases of the weights will result in faster algorithms. To this end, we present a slightly more aggressive update rule. This rule takes into consideration the fact that a contract competes with its own weight as the weight increases.

Consider Example 1. Here, contract A has a weight w_A, and contract B has a weight w_B. If B is already satisfied, we can exactly calculate the right weight for w_A based upon the current w_A, b_A, e_A, and a new variable t_A, which is the total number of opportunities available to A when there is no contention.

Note that e_A/t_A without frequency capping equals $w_A/(w_A+w_B)$, so $w_B = \frac{t_A}{e_A}w_A - w_A$.

Thus, the new w'_A should be:

$$\frac{w'_A}{w'_A + w_B} = \frac{b_A}{t_A} \tag{12}$$

$$w'_A\left(1-\frac{b_A}{t_A}\right) = \frac{b_A}{t_A}w_B \tag{13}$$

$$w'_A\left(1-\frac{b_A}{t_A}\right) = \frac{b_A}{t_A}\left(w_A\frac{t_A}{e_A} - w_A\right) \tag{14}$$

$$w'_A = w_A\frac{\frac{t_A}{e_A}-1}{\frac{t_A}{b_A}-1} \tag{15}$$

Note that while it is no longer an exact solution if we move to scenarios more complex than Example 1, we can still apply it. This weight modification will be nearly as fast as $w'_A = \frac{b_A}{e_A}w_A$. Moreover, you can replace b_A with $b_A(1+\epsilon)$, which will guarantee a $(1+\epsilon)$ multiplicative increase every iteration.

The downside is that for frequency capped contracts, this estimate is *too aggressive*. In particular, for Example 3, the above rule starts to overserve the first contract. A way that this could be mitigated would be to ignore frequency caps when calculating t_c: this will result in underestimates of what w_A should be, but note that as $t_A \to \infty$, the weight update rule approaches the original weight update rule.

8 Conclusion

In this paper, we have introduced a new method (the weights method) for guaranteed delivery. We have introduced a new method of serving, planning, and booking that are internally consistent with one another, and which can be integrated with any method of forecasting which supports simulations. Moreover, we have proven how it can guarantee results in more domains than existing methodologies.

Finally, this method could in practice be used heuristically in cases where there were not guarantees. There is no known example where there exists a set of weights that solve a problem (and there is some slack), and yet the **WeightPlanning** algorithm does not find them. Stating this as a conjecture:

Conjecture 24 *If there exists weights w that solve a booking problem with $(1+\epsilon)b$, then the algorithm* WeightedServing *can resolve that problem.*

Two issues we have not resolved is accurate forecasting and the actual number of impressions served. The actual number of impressions for larger contracts will likely be close to the true expected number, by Chernoff Bounds. However, forecasting is not as trivial: because the past experience may (or may not) be a good indication of future performance, it is impossible to directly prove anything about accuracy here without unreasonable assumptions about the ability to forecast. For example a news site might have a sudden increase in traffic on a day that something interesting happens. Individuals can also have a dramatic effect on traffic: for example, the slashdot effect is when a large site references a small site, causing its traffic to spike to unprecedented levels. Everything from holidays to hurricanes has an effect on traffic. One interesting approach might be to have a variety of forecasting models, or a distribution over models. Is there a way to guarantee at booking time that with some (high) probability that we satisfy the contract given our distribution over models is correct?

As discussed in Section 2, there are technical limitations to the number of impressions that can be obtained using weight allocations. These are not limited to weights, but any time that the distribution over the ad shown is a monotonic function of the values of the applicable ads. These could be resolved by having different weights for different opportunities (which is basically the same as saying different probabilities for different opportunities), but deriving a new compact representation might be hard. Since these problems occur with frequency capping and property targeting, natural extensions involve different weights for highly trafficked properties, or different weights for an ad depending on how many times that ad has been seen before by that user. As these methods introduce far more complicated planning problems, the most profitable direction of research is likely in understanding and tuning the **WeightPlanning** algorithm, leveraging new empirical and theoretical insights.

References

[DCMCS10] N. Devanur, D. Charles, K. Jain M. Chickering, and M. Sanghi. Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads. In *ACM Electronic Commerce*, 2010.

[Far09] Ayman Farahat. Privacy preserving frequency capping in internet banner advertising. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 1147–1148, New York, NY, USA, 2009. ACM.

[FIMN08] Ureil Feige, Nicole Immorlica, Vahab Mirrokni, and Hamid Nazerzadeh. A combinatorial allocation mechanism with penalties for banner advertising. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 169–178, New York, NY, USA, 2008. ACM.

[VVS10] E. Vee, S. Vassilvitskii, and J. Shanmugasundaram. Optimal online assignment with forecasts. In *ACM Electronic Commerce*, 2010.

[YT08] J. Yang and J.A. Tomlin. Advertising inventory allocation based on multi-objective optimization. Presented at INFORMS, Washington, DC, October 2008.

A Calculating $A_{u,d}$

Define C^* to be the set of finite sequences of contracts. Given a sequence $h \in C^*$ and $c \in C$, define:

1. $|h|$ to be the length of the h,
2. h_i to be the ith element of h.
3. $h(i)$ is the first i elements of h.
4. $\text{count}(h,c) = \sum_{i=1}^{n} I(h_i = c)$, the number of times c occurred in h.

Similarly, $F^{u,d}$ is a sequence, with $|F^{u,d}|$ and $F_i^{u,d}$ being defined similarly to the above. However, a cleaner representation is to define $C^{u,d} \in \mathcal{P}(C)^{|F^{u,d}|}$, where $C_i^{u,d} = C_{F_i^{u,d}}$, or in other words, the contracts that match the ith opportunity in $F_i^{u,d}$. Given the model, we can also define $C_h = \{c \in C : \text{count}(h,c) < f_c\}$. Given this notation, given $C' \in \mathcal{P}(C)^*$ we can write $N(w,h,C') = \sum_{c \in C'_{|h|+1} \cap C_h} w_c$, the normalization constant, and the probability of selecting a campaign c:

$$p_c(w,h,C') = \begin{cases} w_c/N(w,h,C') & \text{if } c \in C'_{|h|+1} \cap C_h \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Note that p_c above is not a probability distribution if $C'_{|h|+1} \cap C_h = \emptyset$. Therefore, define $c_x \notin C$ to be a special token such that $p_{c_x}(w,h,C') = 1 - \sum_{c \in C} p_c(w,h,C')$. Define $C_x = C \cap \{c_x\}$. Thus, the probability of a history $h \in {C_x}^{|C'|}$ can be written:

$$p_h(w,C') = \prod_{i=1}^{|h|} p_{h_i}(w,h(i-1),C'_i) \tag{17}$$

Now, given the probability of any history, we can define the expected number of impressions served to a contract:

$$A_c^{u,d}(w) = \sum_{h \in {C_x}^{|C^{u,d}|}} p_h(w,C'^{u,d})\text{count}(h,c). \tag{18}$$

B Frequency Capping Without Property Targeting

Although we can formally define what it means to have no property targeting, when proving things about this condition it is easiest to simply redefine our model in a way that property targeting is impossible to represent. In this section, we define a_c^k. We first define it in a way similar to the previous section, but then we provide an equivalent recursive definition as well. The lemmas in this section are quite straightforward to prove, but are leveraged throughout the rest of the appendix.

If there is no property targeting, then we can represent $A^{u,d}(w)$ as a function of:

1. the length of $F^{u,d}$,
2. the number of matching contracts to an element of $F^{u,d}$, and
3. the frequency cap of each matching contract to $F^{u,d}$.
4. the weights of the matching contracts.

Define $\mathcal{F} = \{0, 1, 2 \ldots\}$ to be the non-negative integers, and we can define $a^k : (\mathbf{R}^+)^n \times \mathcal{F}^n \to (\mathbf{R}^+)^n$ such that $a_c^k(w,f') = A_c^{u,d}(w)$ if:

1. $f'_{c'} = \max(f_{c'}, |F^{u,d}|)$ if $F^{u,d} \cap M_{c'} = F^{u,d}$, and $f'_{c'} = 0$ otherwise.
2. k is the maximum number of impressions that can be shown to user u. This is usually $|F^{u,d}|$, the number of visits of u, but if $|F^{u,d}| > \sum_{c' \in C} f'_{c'}$, it is this latter number.

First, we define the probability p of showing an impression:

$$p_c(w, f) = \begin{cases} \frac{w_c}{\sum_{\{c': f_{c'} \neq 0\}} w_{c'}} & \text{if } f_c > 0 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Define $1^c \in \mathcal{F}^n$ such that $1^c_c = 1$, and $1^c_{c'} = 0$ for $c' \neq c$. For any $h \in C^*$, define 1^h such that $1^h_c = \text{count}(h, c)$. Then:

$$p_h(w, f) = \prod_{i=1}^{|h|} p_{h_i}(w, f - 1^{h(i-1)}) \tag{20}$$

If $\sum_{c' \in C} f_c \geq k$, we can define a^k_c as:

$$a^k_c(w, f) = \sum_{h \in C^k} \text{count}(h, c) p_h(w, f). \tag{21}$$

Otherwise, we define $a^k_c = a^j_c$, where $j = \sum_{c' \in C} f_{c'}$.

Lemma 25 *If there is no property targeting, then $A^{u,d}_c(w) = a^k_c(w, f')$, where $f'_{c'} = \max(f_{c'}, |F^{u,d}|)$ if $F^{u,d} \cap M_{c'} = F^{u,d}$, and $f'_{c'} = 0$ otherwise.*

Proof: Note that if $\sum_{c' \in C} f_{c'} < k$, then $A^{u,d}_c(w) = f_c = a^k_c(w, f)$.

First, we show that $p_h(w, C'^{u,d}) = p_h(w, f')$ if $k = |F^{u,d}|$. First, define $C'' = C'^{u,d}_1$, and observe that without property targeting, for all i, $C'' = C'^{u,d}_i$. Moreover, observe that $f'_{c'} = \max(f'_{c'}, |F^{u,d}|)$ if and only if $c' \in C''$, and $f'_{c'} = 0$ otherwise.

$$p_h(w, C'^{u,d}) = \prod_{i=1}^{|h|} p_{h_i}(w, h(i-1), C'^{u,d}_i) \tag{22}$$

$$p_h(w, C'^{u,d}) = \prod_{i=1}^{|h|} p_{h_i}(w, h(i-1), C'') \tag{23}$$

There are two cases.

1. If $p_h(w, C'^{u,d}) = 0$, then there exists a first i such that $p_{h_i}(w, h(i-1), C') = 0$. Then, either $w_{h_i} = 0$, in which case $p_{h_i}(w, f' - 1^{h(i-1)}) = 0$. Or, $h_i \notin C'' \cap C_{h(i-1)}$. If $h_i \notin C''$, then $f'_{h_i} = 0$ and therefore $p_{h_i}(w, f' - 1^{h(i-1)}) = 0$. If $h_i \in C''$ and $h_i \notin C_{h(i-1)}$, then $f'_{h_i} - 1^{h(i-1)}_{h_i} = f_{h_i} - 1^{h(i-1)}_{h_i} = 0$, so $p_{h_i}(w, f' - 1^{h(i-1)}) = 0$. Therefore, $p_{h_i}(w, f' - 1^{h(i-1)}) = 0$, and $p_h(w, f') = 0$.
2. If $p_h(w, C'^{u,d}) \neq 0$, then, for any i, $p_h(w, h(i-1), C'^{u,d}) > 0$, so $p_{h_i}(w, h(i-1), C'^{u,d}) = w_{h_i}/N(w, h(i-1), C'^{u,d})$, $h_i \in C''$, and $h_i \in C_{h(i-1)}$. This means that $f_{h_i} > \text{count}(h(i-1), h_i)$, or in other words, $(f' - 1^{h(i-1)})_{h_i} > 0$. Thus, $p_{h_i}(w, f' - 1^{h(i-1)}) = \frac{w_{h_i}}{\sum_{c'':(f-1^{h(i-1)})_{c''}} w_{c''}}$. Note that for all $c'' \in C$, $(f' - 1^{h(i-1)})_{c''} > 0$ if and only if $c'' \in C'' \cap C_{h(i-1)}$, so $N(w, h(i-1), C'^{u,d}) = \sum_{c'':(f-1^{h(i-1)})_{c''} > 0} w_{c''}$, so therefore $p_{h_i}(w, f' - 1^{h(i-1)}) = \frac{w_{h_i}}{N(w, h(i-1), C'^{u,d})} = p_h(w, h(i-1), C'^{u,d})$. Therefore, since each multiplicand in the two products are equal, the products are equal.

Thus, since the probabilities are equal, then

$$a^k_c(w, f) = \sum_{h \in C^k} \text{count}(h, c) p_h(w, C'^{u,d}) \tag{24}$$

$$a^k_c(w, f) = A^{u,d}_c(w). \tag{25}$$

■

Now that we have established that a_c^k accurately captures the problem without property targeting, we move on to describe the recursive nature of a_c^k. Since a lot of properties about a_c^k are easiest to prove recursively on k, having a recursive formulation simplifies their presentation dramatically. We first show the recursive nature of p:

Fact 26 $p_{h \circ h'}(w, f) = p_h(w, f) p_{h'}(w, f - 1^h)$.

Lemma 27 *If* $k \geq \sum_{c \in C} \max(f'_c, 0)$*, then:*

$$\sum_{h \in C^k} p_h(w, f') = 1 \tag{26}$$

Proof: We prove this by recursion. First, it is true by definition for $k = 0$. Secondly, assume it is true for $k - 1$. Then, we can write:

$$\sum_{h \in C^k} p_h(w, f') = \sum_{c \in C, h \in C^{k-1}} p_{c \circ h}(w, f') \tag{27}$$

$$\sum_{h \in C^k} p_h(w, f') = \sum_{c \in C} \sum_{h \in C^{k-1}} p_c(w, f') p_{c \circ h}(w, f' - 1^c) \tag{28}$$

$$\sum_{h \in C^k} p_h(w, f') = \sum_{c \in C} p_c(w, f') \sum_{h \in C^{k-1}} p_{c \circ h}(w, f' - 1^c) \tag{29}$$

Now, if $f'_c \leq 0$, then $p_c(w, f') = 0$, so:

$$\sum_{h \in C^k} p_h(w, f') = \sum_{c \in C : f'_c > 0} p_c(w, f') \sum_{h \in C^{k-1}} p_{c \circ h}(w, f' - 1^c) \tag{30}$$

If $f'_c > 0$, then $\left(\sum_{c \in C} \max(f'_i, 0)\right) - 1 = \sum_{c' \in C} \max((f' - 1^c)_{c'}, 0)$. Therefore, $\sum_{c' \in C} \max((f' - 1^c)_{c'}, 0) \leq k - 1$, so by induction:

$$\sum_{h \in C^k} p_h(w, f') = \sum_{c \in C : f'_c > 0} p_c(w, f') \tag{31}$$

Finally, since $p_c(w, f') = \frac{w_c}{\sum_{c' \in C : f'_{c'} > 0} w_{c'}}$:

$$\sum_{h \in C^k} p_h(w, f') = \sum_{c \in C : f'_c > 0} \frac{w_c}{\sum_{c' \in C : f'_{c'} > 0} w_{c'}} \tag{32}$$

$$\sum_{h \in C^k} p_h(w, f') = \frac{1}{\sum_{c' \in C : f'_{c'} > 0} w_{c'}} \sum_{c \in C : f'_c > 0} w_c \tag{33}$$

$$\sum_{h \in C^k} p_h(w, f') = 1. \tag{34}$$

■

We can also describe a^k recursively:

Lemma 28

$$a^0 = 0$$

$$a_c^k(w, f') = p_c(w, f') + \sum_{\{c' : f_{c'} \neq 0\}} p_{c'}(w, f') a_c^{k-1}(w, f' - 1^{c'}) \tag{35}$$

Proof: We need to show that Equation 35 is the same as Equation 21. This holds for zero, because $a_c^0(w,f') = \sum_{h\in C^0} p_h(w,f')\text{count}(h,c) = p_\emptyset(w,f')\text{count}(\emptyset,c) = 0$. We can prove the other case without resorting to induction. By the definition of a_c^k:

$$a_c^k(w,f) = \sum_{h\in C^k} \text{count}(h,c)p_h(w,f) \tag{36}$$

$$a_c^k(w,f) = \sum_{c'\in C, h\in C^{k-1}} \text{count}(c'\circ h,c)p_{c'\circ h}(w,f) \tag{37}$$

$$\begin{aligned} a_c^k(w,f) = &\sum_{h\in C^{k-1}} p_{c\circ h}(w,f) \\ &+ \sum_{c'\in C, h\in C^{k-1}} \text{count}(h,c)p_{c'\circ h}(w,f) \end{aligned} \tag{38}$$

By Fact 26:

$$\begin{aligned} a_c^k(w,f) = &\sum_{h\in C^{k-1}} p_c(w,f)p_h(w,f-1^c) \\ &+ \sum_{c'\in C, h\in C^{k-1}} \text{count}(h,c)p_{c'}(w,f)p_h(w,f-1^{c'}) \end{aligned} \tag{39}$$

$$\begin{aligned} a_c^k(w,f) = &p_c(w,f)\sum_{h\in C^{k-1}} p_h(w,f-1^c) \\ &+ \sum_{c'\in C} p_{c'}(w,f)\sum_{h\in C^{k-1}} \text{count}(h,c)p_h(w,f-1^{c'}) \end{aligned} \tag{40}$$

By the definition of a_c^{k-1}:

$$\begin{aligned} a_c^k(w,f) = &p_c(w,f)\sum_{h\in C^{k-1}} p_h(w,f-1^c) \\ &+ \sum_{c'\in C} p_{c'}(w,f)a_c^{k-1}(w,f-1^{c'}) \end{aligned} \tag{41}$$

By Lemma 27:

$$a_c^k(w,f) = p_c(w,f) + \sum_{c'\in C} p_{c'}(w,f)a_c^{k-1}(w,f-1^{c'}) \tag{42}$$

■

C Proof That Frequency Capping Model is Well-Behaved

Fact 29 *If for some $c'\in C$, $f_c > 0$, then $\sum_{c\in C} p_c(w,f) = 1$.*

Fact 30 $a_c^k(w,f) \leq k$.

Fact 31 $a_c^k(w,f) \leq f_c$.

First, with Lemma 32 and Lemma 33, we prove that scaling all the weights has no effect.

Lemma 32 *For any $v\in(\mathbf{R}^+)^n$, for any $f\in\mathcal{F}^n$, for any $\alpha\geq 1$, $p(v,f) = p(\alpha v,f)$.*

Proof: For any $c \in C$, if $f_c = 0$, then $p_c(v, f) = 0 = p_c(\alpha v, f)$. Otherwise:

$$p_c(\alpha v, f) = \frac{\alpha v_c}{\sum_{c':f_{c'} \neq 0} \alpha v_c} \tag{43}$$

$$p_c(\alpha v, f) = \frac{v_c}{\sum_{c':f_{c'} \neq 0} v_c} \tag{44}$$

$$p_c(\alpha v, f) = p_c(v, f) \tag{45}$$

■

Lemma 33 *For any* $v \in (\mathbf{R}^+)^n$, *for any* $f \in \mathcal{F}^n$, *for any* $\alpha \geq 1$, $a^k(v, f) = a^k(\alpha v, f)$.

Proof: Effectively, multiplying everything by a ratio changes nothing, but to be complete, we prove by induction on k. First, observe that $a^0 = 0$, so $a^0(v, f) = a^0(\alpha v, f)$. By definition:

$$a^k(\alpha v, f) = p_c(\alpha v, f) + \sum_{c':f_{c'} \neq 0} p_{c'}(\alpha v, f) a^{k-1}(\alpha v, f) \tag{46}$$

$$a^k(\alpha v, f) = p_c(v, f) + \sum_{c':f_{c'} \neq 0} p_{c'}(v, f) a^{k-1}(\alpha v, f) \text{ (by Lemma 32)} \tag{47}$$

$$a^k(\alpha v, f) = p_c(v, f) + \sum_{c':f_{c'} \neq 0} p_{c'}(v, f) a^{k-1}(v, f) \text{ (by induction)} \tag{48}$$

$$a^k(\alpha v, f) = a^k(v, f) \text{ (by definition)} \tag{49}$$

■

Now, with Lemmas 34-36, we show that increasing a weight does not decrease the number of impressions for that contract.

Lemma 34 *For any* $w \in (\mathbf{R}^+)^n$, *for any* $f \in \mathcal{F}^n$, *any* $c \in C$, *if* $f_c \geq 1$ *then* $a_c^{k-1}(w, f - 1^c) \geq a_c^k(w, f) + 1$. *In other words, increasing the frequency cap of c by 1 and increasing the number of impressions remaining will not increase the number of impressions by more than 1.*

Proof: We prove this via induction on k. First, $a^0 = 0$ and by Fact 30 $a^1(w, f) \leq 1$, so $a^0(w, f - 1^c) \geq a^1(w, f) - 1$. Assume that the result holds for $k - 1$.

1. If $f_c = 1$, then by Fact 31, $a_c^k(w, f) \leq f_c = 1$ and $a_c^{k-1}(w, f - 1^c) = 0$, so the result holds.
2. Otherwise, we know that $p(w, f) = p(w, f - 1^c)$, so:

$$a_{k-1}(w, f - 1^c) = p_c(w, f - 1^c) + \sum_{c':f_{c'} \neq 0} p_{c'}(w, f - 1^c) a^{k-2}(w, f - 1^c - 1^{c'}) \tag{50}$$

$$a_{k-1}(w, f - 1^c) = p_c(w, f) + \sum_{c':f_{c'} \neq 0} p_{c'}(w, f) a^{k-2}(w, f - 1^c - 1^{c'}) \tag{51}$$

$$a_{k-1}(w, f - 1^c) \geq p_c(w, f) + \sum_{c':f_{c'} \neq 0} p_{c'}(w, f)(a^{k-1}(w, f - 1^{c'}) - 1) \text{ (by induction)} \tag{52}$$

$$a_{k-1}(w, f - 1^c) \geq -1 + p_c(w, f) + \sum_{c':f_{c'} \neq 0} p_{c'}(w, f) a^{k-1}(w, f - 1^{c'}) \text{ (by Fact 29)} \tag{53}$$

$$a_{k-1}(w, f - 1^c) \geq -1 + a^k(w, f) \tag{54}$$

■

Lemma 35 *For any $w \in (\mathbf{R}^+)^n$, for any $f \in \mathcal{F}^n$, any $c, c' \in C$ where $c \neq c'$, if $f_c \geq 1$ and $f_{c'} \geq 1$, then $a_c^k(w, f - 1^c) \geq a_c^k(w, f - 1^{c'}) - 1$. In other words, increasing the frequency cap of c by 1 and decreasing the cap of another contract will not increase the number of impressions by more than 1.*

Proof: If $f_c = 1$, then since $a_c^k(w, f - 1^{c'}) \leq f_c = 1$, and $a_c^k(w, f - 1^c) = 0$, the result holds.

We prove this via induction on k. First, observe that $a^0 = 0$, so $a^0(w, f - 1^c) = a^0(w, f - 1^{c'}) \leq a^0(w, f - 1^{c'}) + 1$. Assume that it holds for $k - 1$.

1. If $f_c > 1$ and $f_{c'} > 1$, then by definition:

$$\begin{aligned} a_c^k(w, f - 1^{c'}) - a_c^k(w, f - 1^c) =& p_c(w, f - 1^{c'}) - p_c(w, f - 1^c) \\ &+ \sum_{c'':f_{c''} \neq 0} p_c(w, f - 1^{c'}) a_c^{k-1}(w, f - 1^{c'} - 1^{c''}) - p_c(w, f - 1^c) a_c^{k-1}(w, f - 1^c - 1^{c''}) \end{aligned} \tag{55}$$

$$a_c^k(w, f - 1^{c'}) - a_c^k(w, f - 1^c) = \sum_{c'':f_{c''} \neq 0} p_c(w, f - 1^c)(a_c^{k-1}(w, f - 1^{c'} - 1^{c'}) - a_c^{k-1}(w, f - 1^c - 1^{c''})) \tag{56}$$

By induction, $a_c^{k-1}(w, f - 1^{c'} - 1^{c''}) - a_c^{k-1}(w, f - 1^c - 1^{c''}) \leq 1$, so:

$$a_c^k(w, f - 1^{c'}) - a_c^k(w, f - 1^c) \leq \sum_{c'':f_{c''} \neq 0} p_c(w, f - 1^c)(1) \tag{57}$$

$$a_c^k(w, f - 1^{c'}) - a_c^k(w, f - 1^c) \leq 1 \tag{58}$$

2. If $f_c > 1$ and $f_{c'} = 1$, then by definition:

$$\begin{aligned} a_c^k(w, f - 1^{c'}) - a_c^k(w, f - 1^c) =& p_c(w, f - 1^{c'}) - p_c(w, f - 1^c) \\ &- p_{c'}(w, f - 1^c) a_c^{k-1}(w, f - 1^c - 1^{c'}) \\ &+ \sum_{c'':f_{c''} \neq 0, c'' \neq c'} p_c(w, f - 1^{c'}) a_c^{k-1}(w, f - 1^{c'} - 1^{c''}) - p_c(w, f - 1^c) a_c^{k-1}(w, f - 1^c - 1^{c''}) \end{aligned} \tag{59}$$

$$\begin{aligned} a_c^k(w, f - 1^c) =& p_c(w, f - 1^c) + \\ & p_{c'}(w, f - 1^c) a_c^{k-1}(w, f - 1^c - 1^{c'}) \\ &+ \sum_{c'':f_{c''} \neq 0, c'' \neq c'} p_c(w, f - 1^c) a_c^{k-1}(w, f - 1^c - 1^{c''}) \end{aligned} \tag{60}$$

By Lemma 34, $a_c^{k-1}(w, f - 1^c - 1^{c'}) \geq a^k(w, f - 1^{c'}) - 1$. Define $S = \sum_{c'' \neq c'} p_{c''}(w, f)$, so that $p_{c'}(w, f - 1^c) = p_{c'}(w, f) = \frac{w_{c''}}{w_{c''} + S}$, and that for any $c'' \neq c'$, $p_{c''}(w, f - 1^c) = p_{c''}(w, f) = \frac{S}{w_{c''} + S} p_{c''}(w, f - 1^{c''})$. Then:

$$\begin{aligned} a_c^k(w, f - 1^{c'}) - a_c^k(w, f - 1^c) =& \frac{w_{c''}}{w_{c''} + S} + \frac{S}{w_{c''} + S} a_c^k(w, f - 1^{c'}) - p_c(w, f - 1^c) \\ &- \sum_{c'':f_{c''} \neq 0, c'' \neq c'} p_c(w, f - 1^c) a_c^{k-1}(w, f - 1^c - 1^{c''}) \end{aligned} \tag{61}$$

Note that:

$$
\begin{aligned}
\frac{S}{w''_c+S}a_c^k(w,f-1^{c'}) = & \frac{S}{w_{c''}+S}p_c(w,f-1^{c'}) \\
& + \sum_{c'':f_{c''}\neq 0, c''\neq c'} \frac{S}{w_{c''}+S}p_c(w,f-1^{c'})a_c^{k-1}(w,f-1^{c'}-1^{c''}) \quad (62) \\
a_c^k(w,f-1^{c'}) - a_c^k(w,f-1^c) \leq & \frac{w_{c''}}{w_{c''}+S} - p_c(w,f-1^c) \\
& - \sum_{c'':f_{c''}\neq 0, c''\neq c'} p_c(w,f-1^c)a_c^{k-1}(w,f-1^c-1^{c''}) \\
& + \frac{S}{w_{c''}+S}p_c(w,f-1^{c'}) \\
& + \sum_{c'':f_{c''}\neq 0, c''\neq c'} \frac{S}{w_{c''}+S}p_c(w,f-1^{c'})a_c^{k-1}(w,f-1^{c'}-1^{c''}) \quad (63)
\end{aligned}
$$

Because for any $c'' \neq c'$, $p_{c''}(w,f-1^c) = \frac{S}{w_{c''}+S}p_{c''}(w,f-1^{c''})$:

$$
\begin{aligned}
a_c^k(w,f-1^{c'}) - a_c^k(w,f-1^c) \leq & \frac{w_{c''}}{w_{c''}+S} - \frac{S}{w_{c''}+S}p_c(w,f-1^{c'}) \\
& - \sum_{c'':f_{c''}\neq 0, c''\neq c'} \frac{S}{w_{c''}+S}p_c(w,f-1^{c'})a_c^{k-1}(w,f-1^c-1^{c''}) \\
& + \frac{S}{w_{c''}+S}p_c(w,f-1^{c'}) \\
& + \sum_{c'':f_{c''}\neq 0, c''\neq c'} \frac{S}{w_{c''}+S}p_c(w,f-1^{c'})a_c^{k-1}(w,f-1^{c'}-1^{c''}) \quad (64) \\
a_c^k(w,f-1^{c'}) - a_c^k(w,f-1^c) \leq & \frac{w_{c''}}{w_{c''}+S} \\
& + \sum_{c'':f_{c''}\neq 0, c''\neq c'} \frac{S}{w_{c''}+S}p_c(w,f-1^{c'})(a_c^{k-1}(w,f-1^{c'}-1^{c''}) - a_c^{k-1}(w,f-1^c-1^{c''})) \\
& \quad (65)
\end{aligned}
$$

By recursion:

$$
\begin{aligned}
a_c^k(w,f-1^{c'}) - a_c^k(w,f-1^c) \leq & \frac{w_{c''}}{w_{c''}+S} + \sum_{c'':f_{c''}\neq 0, c''\neq c'} \frac{S}{w_{c''}+S}p_c(w,f-1^{c'})1 \quad (66) \\
a_c^k(w,f-1^{c'}) - a_c^k(w,f-1^c) \leq & \frac{w_{c''}}{w_{c''}+S} + \frac{S}{w_{c''}+S} \quad (67) \\
a_c^k(w,f-1^{c'}) - a_c^k(w,f-1^c) \leq & 1 \quad (68)
\end{aligned}
$$

■

Lemma 36 *For any $v \in (\mathbf{R}^+)^n$, for any $f \in \mathcal{F}^n$, for any $\alpha \geq 1$, for any $c \in C$, given $w \in (\mathbf{R}^+)^n$ where $w_c = \alpha v_c$, and for all $c' \in C$ where $c' \neq c$, $w_{c'} = v_{c'}$, $a_c^k(w,f) \geq a_c^k(v,f)$.*

Proof: We prove this by induction on k. For $k=0$, $a^0(w,f) = a^0(v,f) = 0$. Assume that the result holds for $k-1$. If $f_c = 0$, then $a^k(w,f_c) = a^k(v,f_c) = 0$. Thus, without loss of generality, assume that $f_c \neq 0$. define $S = (\sum_{c'':f_{c''}\neq 0} v_{c''}) - v_c$. Then, for $c' \neq c$, $p_{c'}(w,f) = \frac{v_{c'}}{w_c+S} = \frac{S}{w_c+S}p_{c'}(v,f)$, and $p_c(w,f) = \frac{w_c}{w_c+S}$.

Therefore:

$$a_c^k(v,f) = p_c(v,f) + \sum_{c'':f_{c''}\neq 0} p_{c''}(v,f)a_c^{k-1}(v,f-1^{c''}) \text{ (by definition)} \tag{69}$$

$$a_c^k(v,f) \leq p_c(v,f) + \sum_{c'':f_{c''}\neq 0} p_{c''}(v,f)a_c^{k-1}(w,f-1^{c''}) \text{ (by induction)} \tag{70}$$

$$a_c^k(w,f) - a_c^k(v,f) \geq p_c(w,f) - p_c(v,f) \tag{71}$$

$$+ \sum_{c'':f_{c''}\neq 0} (p_{c''}(w,f) - p_{c''}(v,f))a_c^{k-1}(w,f-1^{c''}) \tag{72}$$

Note that since $\sum_{c'\in C} p_c(w,f) = 1 = \sum_{c'\in C} p_c(v,f)$, we can write $p_c(w,f) - p_c(v,f) = \sum_{c'\neq c}(p_c(v,f) - p_c(w,f))$. Since $p_{c''}(v,f) = p_{c''}(w,f) = 0$ if $f_{c''} = 0$, $p_c(w,f) - p_c(v,f) = \sum_{c'':f_{c''}\neq 0, c''\neq c}(p_{c''}(v,f) - p_{c''}(w,f)))$.

$$a_c^k(w,f) - a_c^k(v,f) \geq p_c(w,f) - p_c(v,f) + (p_c(w,f) - p_c(v,f))a_c^k(w,f-1^c) \tag{73}$$

$$+ \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(w,f) - p_{c''}(v,f))a_c^k(w,f-1^{c''}) \tag{74}$$

$$a_c^k(w,f) - a_c^k(v,f) \geq (p_c(w,f) - p_c(v,f))(1 + a_c^k(w,f-1^c)) \tag{75}$$

$$+ \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(w,f) - p_{c''}(v,f))a_c^k(w,f-1^{c''}) \tag{76}$$

$$a_c^k(w,f) - a_c^k(v,f) \geq (\sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f)))(1 + a_c^k(w,f-1^c)) \tag{77}$$

$$+ \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(w,f) - p_{c''}(v,f))a_c^k(w,f-1^{c''}) \tag{78}$$

$$a_c^k(w,f) - a_c^k(v,f) \geq (\sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f)))(1 + a_c^k(w,f-1^c)) \tag{79}$$

$$- \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f))a_c^k(w,f-1^{c''}) \tag{80}$$

$$a_c^k(w,f) - a_c^k(v,f) \geq \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f))(1 + a_c^k(w,f-1^c) - a_c^k(w,f-1^{c''})) \tag{81}$$

Observe that for $c'' \neq c$, $p_{c''}(v,f) = \frac{w_{c''}}{w_c+S} = \frac{v_{c''}}{w_c+S} = \frac{v_c+S}{w_c+S}\frac{v_{c''}}{v_c+S} = \frac{v_c+S}{w_c+S}p_{c''}(v,f)$. Since $v_c < w_c$, $p_{c''}(v,f) > p_{c''}(w,f)$.

Secondly observe that for all w, f, where $f_c \neq 0$, for all $c'' \neq c$ where $f_{c''} \neq 0$, by Lemma 35,$1 + a_c^k(w,f-1^c) - a_c^k(w,f-1^{c''}) \geq 0$. Thus, each summand in the sum is a product of two non-negative numbers, meaning the sum is non-negative. So:

$$a_c^k(w,f) - a_c^k(v,f) \geq 0 \tag{82}$$

$$a_c^k(w,f) \geq a_c^k(v,f). \tag{83}$$

■

Now, we prove that increasing the weight of a contract does not increase the mass too much.

Lemma 37 *For any $w \in (\mathbf{R}^+)^n$, for any $f, f' \in \mathcal{F}^n$, if $f' \geq f$ but $f_c = f'_c$, then $a_c^k(w,f) \geq a_c^k(w,f')$. In other words, increasing the frequency cap of c' will not increase the number of impressions received by c.*

Proof: The proof is via recursion on recursion. First, it is obvious for a^0, because $a^0 = 0$. We prove it for all k. However, in the recursive step it is sufficient to prove (given the result for $k-1$), $a_c^k(w,f) \geq a^k(w,f+1^{c'})$ on some $c' \in C$ where $c' \neq c$. Then, by induction one can prove it for any $f' \geq f$ where $f'_c = f_c$. Consider two cases:

1. $f_{c'} \neq 0$. Then $p(w,f) = p(w, f+1^{c'})$. By the definition of a^k:

$$a_c^k(w, f+1^{c'}) = p_c(w,f) + \sum_{\{c'':f_{c''}\neq 0\}} p_{c''}(w,f) a_c^{k-1}(w, f+1^{c'} - 1^{c''}). \tag{84}$$

By induction, $a_c^{k-1}(w, f+1^{c'} - 1^{c''}) \leq a_c^{k-1}(w, f-1^{c''})$, so:

$$a_c^k(w, f+1^{c'}) \leq p_c(w,f) + \sum_{\{c'':f_{c''}\neq 0\}} p_{c''}(w,f) a_c^{k-1}(w, f-1^{c''}) \tag{85}$$

$$a_c^k(w, f+1^{c'}) \leq a_c^k(w,f). \tag{86}$$

2. $f_{c'} = 0$. Then $p(w,f) \neq p(w, f+1^{c'})$. In particular, define $S = \sum_{c'':f_{c''}\neq 0} w_{c''}$. Then, $p_{c'}(w, f+1^{c'}) = \frac{w_{c'}}{S+w_{c'}}$, and for $c'' \neq c'$, $p_{c''}(w, f+1^{c'}) = \frac{S}{S+w_{c'}} p_{c''}(w,f)$. Then, it is clear that:

$$a_c^k(w, f+1^{c'}) = p_c(w, f+1^{c'}) + p_{c'}(w, f+1^{c'}) a_c^{k-1}(w,f) \tag{87}$$

$$+ \sum_{\{c'':f_{c''}\neq 0\}} p_{c''}(w, f+1^{c'}) a_c^{k-1}(w, f+1^{c'} - 1^{c''}) \tag{88}$$

$$a_c^k(w, f+1^{c'}) \leq \frac{S}{S+w_{c'}} p_{c'}(w,f) + \frac{w_{c'}}{S+w_{c'}} a_c^{k-1}(w,f) \tag{89}$$

$$+ \sum_{\{c'':f_{c''}\neq 0\}} \frac{S}{S+w_{c'}} p_{c''}(w,f) a_c^{k-1}(w, f-1^{c''}) \text{ (by induction)} \tag{90}$$

$$a_c^k(w, f+1^{c'}) \leq \frac{S}{S+w_{c'}} \left(p_{c'}(w,f) + \sum_{\{c'':f_{c''}\neq 0\}} p_{c''}(w,f) a_c^{k-1}(w, f-1^{c''})) \right) \tag{91}$$

$$+ \frac{w_{c'}}{S+w_{c'}} a_c^{k-1}(w,f) \tag{92}$$

$$a_c^k(w, f+1^{c'}) \leq \frac{S}{S+w_{c'}} \left(a_c^k(w,f) \right) + \frac{w_{c'}}{S+w_{c'}} a_c^{k-1}(w,f) \tag{93}$$

Therefore, we are left with a mix of $a_c^k(w,f)$ and $a_c^{k-1}(w,f)$. Note that $a_c^k(w,f) \geq a_c^{k-1}(w,f)$, so:

$$a_c^k(w, f+1^{c'}) \leq \frac{S}{S+w_{c'}} \left(a_c^k(w,f) \right) + \frac{w_{c'}}{S+w_{c'}} a_c^k(w,f) \tag{94}$$

$$a_c^k(w, f+1^{c'}) \leq a_c^k(w,f) \tag{95}$$

■

Lemma 38 *For any $w \in (\mathbf{R}^+)^n$, for any $f \in \mathcal{F}^n$, $a_c^k(w, f+1^c) \geq a_c^k(w,f)$. In other words, increasing the frequency cap of c will not decrease the number of impressions received by c.*

Proof: We prove this by induction on k. First, observe that $a_c^0(w, f+1^c) = a_c^0(w,f) = 0$. Now, assume that the result holds for $k-1$. If $f_c = 0$, then $a^k(w, f+1^c) \geq 0 = a^k(w,f)$. Therefore, we can assume that $f_c \neq 0$. This implies $(f+1^c)_{c'} \neq 0$ if and only if $f_{c'} \neq 0$, and $p(w,f) = p(w, f+1^c)$, so:

$$a_c^k(w, f+1^c) = p(w, f+1^c) + \sum_{c':(f+1^c)_{c'}\neq 0} p(w, f+1^c) a_c^{k-1}(w, f+1^c - 1^{c'}) \tag{96}$$

$$a_c^k(w, f+1^c) = p(w,f) + \sum_{c':f_{c'}\neq 0} p(w,f) a_c^{k-1}(w, f+1^c - 1^{c'}) \tag{97}$$

$$a_c^k(w, f+1^c) \geq p(w,f) + \sum_{c':f_{c'}\neq 0} p(w,f) a_c^{k-1}(w, f-1^{c'}) \text{ (by induction)} \tag{98}$$

$$a_c^k(w, f+1^c) \geq a^k(w,f) \text{ (by definition)} \tag{99}$$

■

Lemma 39 *For any $v \in (\mathbf{R}^+)^n$, for any $f \in \mathcal{F}^n$, for any $\alpha \geq 1$, for any $c \in C$, given $w \in (\mathbf{R}^+)^n$ where $w_c = \alpha v_c$, $a_c^k(w,f) \leq \alpha a_c^k(v,f)$.*

Proof: We prove this by induction on k. First, observe that $a^0 = 0$, so $a_c^0(w,f) \leq \alpha a^0(v,f)$.

$$a_c^k(w,f) = p_c(w,f) + \sum_{c'':f_{c''}\neq 0} p_c(w,f) a_c^{k-1}(w, f - 1^{c''}) \text{ (by definition)} \tag{100}$$

$$a_c^k(w,f) \leq p_c(w,f) + \sum_{c'':f_{c''}\neq 0} p_c(w,f) \alpha a_c^{k-1}(v, f - 1^{c''}) \text{ (by induction)} \tag{101}$$

$$\alpha a_c^k(v,f) - a_c^k(w,f) \geq \alpha p_c(v,f) - p_c(w,f) \tag{102}$$

$$+ \sum_{c'':f_{c''}\neq 0} (\alpha p_{c''}(v,f) - \alpha p_{c''}(w,f)) a_c^{k-1}(v, f - 1^{c''}) \tag{103}$$

$$\alpha a_c^k(v,f) - a_c^k(w,f) \geq \alpha p_c(v,f) - p_c(w,f) \tag{104}$$

$$+ \alpha \sum_{c'':f_{c''}\neq 0} (p_{c''}(v,f) - p_{c''}(w,f)) a_c^{k-1}(v, f - 1^{c''}) \tag{105}$$

Define $S = \sum_{c' \neq c: f_{c'} \neq 0} w_{c'}$. Then, for $c' \neq c$, $p_{c'}(w,f) = \frac{v_c + S}{w_c + S} p_{c'}(v,f)$, and $p_c(w,f) = \frac{w_c}{w_c + S}$.

$$\begin{aligned}\alpha a_c^k(v,f) - a_c^k(w,f) \geq{}& \alpha p_c(v,f) - p_c(w,f) \\ &+ \alpha(p_c(v,f) - p_c(w,f)) a_c^{k-1}(v, f - 1^c) \\ &+ \alpha \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f)) a_c^{k-1}(v, f - 1^{c''}) \end{aligned} \tag{106}$$

$$\tag{107}$$

Note that since $\sum_{c' \in C} p_c(w,f) = 1 = \sum_{c' \in C} p_c(v,f)$, we can write $p_c(w,f) - p_c(v,f) = \sum_{c' \neq c} (p_c(v,f) - p_c(w,f))$. Since $p_{c''}(v,f) = p_{c''}(w,f) = 0$ if $f_{c''} = 0$, $p_c(w,f) - p_c(v,f) = \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f)))$.

$$\begin{aligned}\alpha a_c^k(v,f) - a_c^k(w,f) \geq{}& \alpha p_c(v,f) - p_c(w,f) \\ &+ \alpha \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(w,f) - p_{c''}(v,f)) a_c^{k-1}(v, f - 1^c) \\ &+ \alpha \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f)) a_c^{k-1}(v, f - 1^{c''}) \end{aligned} \tag{108}$$

$$\begin{aligned}\alpha a_c^k(v,f) - a_c^k(w,f) \geq{}& \alpha p_c(v,f) - p_c(w,f) \\ &+ \alpha \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f)) (a_c^{k-1}(v, f - 1^{c''}) - a_c^{k-1}(v, f - 1^c)) \end{aligned} \tag{109}$$

By Lemma 37 and Lemma 38, $a_c^{k-1}(v, f - 1^{c''}) - a_c^{k-1}(v, f - 1^c) \geq 0$. Since $c'' \neq c$, $p_{c''}(v,f) - p_{c''}(w,f) \geq 0$, so $\alpha \sum_{c'':f_{c''}\neq 0, c''\neq c} (p_{c''}(v,f) - p_{c''}(w,f))(a_c^{k-1}(v, f - 1^{c''}) - a_c^{k-1}(v, f - 1^c)) \geq 0$, which implies:

$$\alpha a_c^k(v,f) - a_c^k(w,f) \geq \alpha p_c(v,f) - p_c(w,f) \tag{110}$$

By definition:

$$p_c(w,f) = \frac{w_c}{w_c + S} \tag{111}$$

$$p_c(w,f) \leq \frac{w_c}{v_c + S} \text{ since } v_c < w_c, \tag{112}$$

$$p_c(w,f) \leq \alpha p_c(v,f) \tag{113}$$

Plugging this into Equation 110:

$$\alpha a_c^k(v,f) - a_c^k(w,f) \geq 0. \tag{114}$$

Thus, $\alpha a_c^k(v,f) \geq a_c^k(w,f)$. ■

D Proving Increasing Your Weight Does Not Help Others

In this section, we prove the last property of well-behavedness (property 1) for frequency capped contracts without property targeting. This analysis is radically different than the recursive analysis presented elsewhere, and it is where the no-property targeting constraint is actually leveraged. In particular, we consider how adding (or removing) one contract effects the other contracts.

Define H_f^k to be the set of all histories of length $j = \min(k, \sum_{c' \in C} f_{c'})$ where $h \in C^j$ is in H_f^k if and only if for all $c \in C$, $\text{count}(h,c) \leq f_c$. As we discussed when we defined a_c^k, in general $k \leq \sum_{c' \in C} f_{c'}$. If this is the case, we can establish that:

Lemma 40 $\sum_{h \in H_f^k} p_h(w,f) = 1$.

Corollary 41 $a_c^k(w,f) = \sum_{h \in H_f^k} \text{count}(h,c) p_h(w,f)$.

Proof: Define $j = \min(k, \sum_{c' \in C} f_{c'})$. By definition,$H_f^j \subseteq C^j$. Since $j \leq \sum_{c' \in C} f_{c'}$, by Lemma 27:

$$1 = \sum_{h \in C^j} p_h(w,f) \tag{115}$$

$$1 = \sum_{h \in H_f^k} p_h(w,f) + \sum_{h \in C^j \setminus H_f^k} p_h(w,f) \tag{116}$$

Note that if $h \in C^j \setminus H_f^k$, then there exists a $c \in C$ such that $f_c < \text{count}(h,c)$. Therefore, $p_h(w,f) = 0$, so:

$$1 = \sum_{h \in H_f^k} p_h(w,f). \tag{117}$$

■

Define $H_{f,a}^k = H_{f'}^k$ where $f'_a = 0$ and $f'_c = f_c$ if $c \neq a$. In what follows, we will connect $H_{f,a}^k$ to H_f^k by inserting as.

Define $\text{last}(h,c)$ to be the largest i such that $h_i = c$.

For $c \in C$, $h \in C^*$, $1 \leq i \leq |h|+1$, Define $\text{Ins}(h,c,i) \in C^{|h|+1}$ such that $\text{Ins}(h,c,i)_j = h_j$ if $j < i$, $\text{Ins}(h,c,i)_j = c$ if $i = j$, and $\text{Ins}(h,c,i)_j = h_{j-1}$ if $j < i$. In other words, $\text{Ins}(h,a,k)$ is a function from a history h, an ad a, and a position k, to a new history where that ad is inserted before that position.

Then, define $\text{Ins}^*(h,a)$ to be a function from a history h, an ad a, to a set of histories where we insert a before each position. Formally,

$$\text{Ins}^*(h,a) = \bigcup_{i=k}^{|h|} \text{Ins}(h,a,k) \tag{118}$$

Then, define ${\text{Ins}^*}_k(h,a)$ to be a function from a history h, an ad a, and an non-negative integer n to a set of histories, where:

$${\text{Ins}^*}_0(h,a) = \{h\} \tag{119}$$

$${\text{Ins}^*}_{k+1}(h,a) = {\text{Ins}^*}_k(h,a) \cup \bigcup_{h' \in {\text{Ins}^*}_k(h,a)} \text{Ins}^*(h',a) \tag{120}$$

Define $G_{f,a}^k = \bigcup_{h \in H_{f,a}^k} {\text{Ins}^*}_{f_a}(h,a)$. Note that for any $h, h' \in H_{f,a}^k$ where $h \neq h'$, ${\text{Ins}^*}_{f_a}(h,a)$ and ${\text{Ins}^*}_{f_a}(h',a)$ are disjoint.

Lemma 42 $\sum_{h \in G^k_{f,a}} p_h(w, f) = 1$.

Proof: Define $j = \sum_{c \in C \setminus a} f_c$. Note that for $k > j$, $G^k_{f,a} = G^j_{f,a}$. Thus, without loss of generality, assume $k \leq j$. We prove the result inductively on k. Observe that for $k = 0$, $H^0_{f,a} = \{\emptyset\}$, a set consisting of one zero-length vector, and $G^0_{f,a} = \{\emptyset\}$. So, $\sum_{h \in G^0_{f,a}} p_h(w, f) = p_\emptyset(w, f) = 1$.

Now, we assume the result holds for $k - 1$. We now have to do induction on f_a. If $f_a = 0$ and $k \geq 1$, then:

$$G^k_{f,a} = \bigcup_{c' \in C \setminus a : f_{c'} \neq 0} \left(c' \circ G^{k-1}_{f-1^{c'},a} \right). \tag{121}$$

Therefore:

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = \sum_{c' \in C \setminus a : f_{c'} \neq 0} \sum_{h' \in G^{k-1}_{f-1^{c'},a}} p_{c' \circ h'}(w, f) \tag{122}$$

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = \sum_{c' \in C \setminus a : f_{c'} \neq 0} p_{c'}(w, f) \sum_{h' \in G^{k-1}_{f-1^{c'},a}} p_{h'}(w, f - 1^{c'}) \tag{123}$$

By induction, $\sum_{h' \in G^{k'-1}_{f-1^{c'},a}} p_{h'}(w, f - 1^{c'}) = 1$, so:

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = \sum_{c' \in C \setminus a : f_{c'} \neq 0} p_{c'}(w, f) \tag{124}$$

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = 1 \tag{125}$$

Thus, we have established the base case for f_a. Now, if $f_a \geq 1$, and the inductive hypothesis holds for $k - 1$ and $f_a - 1$:

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = \left(\sum_{h \in G^{k-1}_{f-1^a,a}} p_{a \circ h}(w, f) \right) + \sum_{c' \in C \setminus a : f_{c'} \neq 0} \sum_{h' \in G^{k-1}_{f-1^{c'},a}} p_{c' \circ h'}(w, f) \tag{126}$$

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = \left(\sum_{h \in G^k_{f-1^a,a}} p_a(w, f) p_h(w, f - 1^a) \right) + \sum_{c' \in C \setminus a : f_{c'} \neq 0} \sum_{h' \in G^{k-1}_{f-1^{c'},a}} p_{c'}(w, f) p_{h'}(w, f - 1^{c'}) \tag{127}$$

$$\sum_{h \in G^k_{f,a}} p_h(w, f) = \left(p_a(w, f) \sum_{h \in G^k_{f-1^a,a}} p_h(w, f - 1^a) \right) + \sum_{c' \in C \setminus a : f_{c'} \neq 0} p_{c'}(w, f) \sum_{h' \in G^{k-1}_{f-1^{c'},a}} p_{h'}(w, f - 1^{c'}) \tag{128}$$

By the inductive hypothesis:

$$\sum_{h\in G^k_{f,a}} p_h(w,f) = p_a(w,f) + \sum_{c'\in C\setminus a: f_{c'}\neq 0} p_{c'}(w,f) \tag{129}$$

$$\sum_{h\in G^k_{f,a}} p_h(w,f) = 1. \tag{130}$$

■

Theorem 43 *If $c \neq a$, and $k' > k$, then:*

$$a^k_c(w,f) = \sum_{h\in H^{k'}_{f,a}} \sum_{h'\in \mathrm{Ins^*}_{f_a}(h,a)} \mathrm{count}(h'(k),c) p_{h'}(w,f) \tag{131}$$

where $h'(k) = h'$ if $k \geq |h'|$.

Proof: Define $j = \sum_{c'\in C} f_{c'}$. If $k > j$, then $a^k_{c'} = a^j_{c'}$. So, therefore, without loss of generality, assume $k \leq j$.

Define $G^{k'}_{f,a} = \bigcup_{h\in H^{k'}_{f,a}} \mathrm{Ins^*}_{f_a}(h,a)$. Note that $H^0_{f,a} = \{\emptyset\}$, a set with one entry which is a sequence of length zero. Also, $G^0_{f,a} = \{\emptyset\}$.

Now, we prove the theorem by induction on k. First, if $k = 0$:

$$\sum_{h\in H^{k'}_{f,a}} \mathrm{count}(h(0),c) p_h(w,f) = \sum_{h\in H^{k'}_{f,a}} 0 \tag{132}$$

$$\sum_{h\in H^{k'}_{f,a}} \mathrm{count}(h(0),c) p_h(w,f) = 0 \tag{133}$$

$$\sum_{h\in H^{k'}_{f,a}} \mathrm{count}(h(0),c) p_h(w,f) = a^0_c(w,f) \tag{134}$$

Now, we assume that the result holds for $k-1$. We now have to do induction on f_a. If $f_a = 0$ and $k \geq 1$, then $k' \geq 1$ and:

$$G^{k'}_{f,a} = \bigcup_{c'\in C\setminus a: f_{c'}\neq 0} \left(c' \circ G^{k'-1}_{f-1^{c'},a}\right). \tag{135}$$

Therefore:

$$\sum_{h\in G^{k'}_{f,a}} \mathrm{count}(h(k),c) p_h(w,f) = \sum_{c'\in C\setminus a: f_{c'}\neq 0} \sum_{h\in G^{k'-1}_{f-1^{c'},a}} \mathrm{count}((c'\circ h)(k),c) p_{c'\circ h}(w,f) \tag{136}$$

$$= \sum_{c'\in C\setminus a: f_{c'}\neq 0} \sum_{h\in G^{k'-1}_{f-1^{c'},a}} (\mathrm{count}(c',c) + \mathrm{count}(h(k-1),c)) p_{c'}(w,f) p_h(w,f-1^{c'}) \tag{137}$$

If $f_c = 0$, then $\sum_{c'\in C\setminus a: f_{c'}\neq 0} \sum_{h\in G^{k'-1}_{f-1^{c'},a}} \mathrm{count}(c',c) p_{c'}(w,f) p_h(w,f-1^{c'}) = 0 = p_c(w,f)$. Otherwise:

$$\sum_{c'\in C\setminus a: f_{c'}\neq 0} \sum_{h\in G^{k'-1}_{f-1^{c'},a}} \mathrm{count}(c',c) p_{c'}(w,f) p_h(w,f-1^{c'}) = p_c(w,f) \sum_{h\in G^{k'-1}_{f-1^c,a}} p_h(w,f-1^c) \tag{138}$$

$$\sum_{c'\in C\setminus a: f_{c'}\neq 0} \sum_{h\in G^{k'-1}_{f-1^{c'},a}} \mathrm{count}(c',c) p_{c'}(w,f) p_h(w,f-1^{c'}) = p_c(w,f) \sum_{h\in H^{k'-1}_{f-1^c}} p_h(w,f-1^c) \tag{139}$$

$$\sum_{c'\in C\setminus a: f_{c'}\neq 0} \sum_{h\in G^{k-1}_{f-1^{c'},a}} \mathrm{count}(c',c) p_{c'}(w,f) p_h(w,f-1^{c'}) = p_c(w,f) \tag{140}$$

Thus, plugging this back into Equation 137:

$$\sum_{h\in G^{k'}_{f,a}} \text{count}(h(k),c)p_h(w,f) = p_c(w,f) + \sum_{c'\in C\backslash a: f_{c'}\neq 0} \sum_{h\in G^{k'-1}_{f-1^{c'},a}} \text{count}(h(k-1),c)p_{c'}(w,f)p_h(w,f-1^{c'}) \tag{141}$$

By the inductive hypothesis, $\sum_{h\in G^{k'-1}_{f-1^{c'},a}} \text{count}(h(k-1),c)p_h(w,f-1^{c'}) = a^{k-1}_{c'}(w,f-1^{c'})$.

$$\sum_{h\in G^{k'}_{f,a}} \text{count}(h(k),c)p_h(w,f) = p_c(w,f) + \sum_{c'\in C\backslash a: f_{c'}\neq 0} p_{c'}(w,f)a^{k-1}_{c'}(w,f-1^{c'}) \tag{142}$$

$$\sum_{h\in G^{k'}_{f,a}} \text{count}(h(k),c)p_h(w,f) = a^k_c(w,f) \tag{143}$$

Observe that if $k>1$ and $f_a>1$, and we have established the fact for $k-1$ and f_a-1, then:

$$G^{k'}_{f,a} = \left(a\circ G^{k'}_{f-1^a,a}\right) \cup \bigcup_{c'\in C\backslash a: f_{c'}\neq 0} \left(c'\circ G^{k-1}_{f-1^{c'},a}\right). \tag{144}$$

Thus, following a similar pattern to before:

$$
\begin{aligned}
\sum_{h \in G^{k'}_{f,a}} \text{count}(h(k), c) p_h(w, f) = &\left(\sum_{h \in G^{k'}_{f-1^a,a}} \text{count}((a \circ h)(k), c) p_{a \circ h}(w, f) \right) \\
&+ \sum_{c' \in C \setminus a : f_{c'} \neq 0} \sum_{h \in G^{k'-1}_{f-1^{c'},a}} \text{count}((c' \circ h)(k), c) p_{c' \circ h}(w, f) && (145) \\
= &\left(p_a(w, f) \sum_{h \in G^{k'}_{f-1^a,a}} \text{count}(h(k-1), c) p_h(w, f - 1^a) \right) \\
&+ \sum_{c' \in C \setminus a : f_{c'} \neq 0} p_{c'}(w, f) \sum_{h \in G^{k'-1}_{f-1^{c'},a}} (\text{count}(c', c) + \text{count}(h(k-1), c)) p_h(w, f - 1^{c'}) && (146) \\
= &\left(p_a(w, f) \sum_{h \in G^{k}_{f-1^a,a}} \text{count}(h(k-1), c) p_h(w, f - 1^a) \right) \\
&+ p_c(w, f) \sum_{h \in G^{k'-1}_{f-1^c,a}} p_h(w, f - 1^c) \\
&+ \sum_{c' \in C \setminus a : f_{c'} \neq 0} p_{c'}(w, f) a_c^{k-1}(w, f - 1^{c'}) && (147) \\
= &\left(p_a(w, f) a^{k-1}(w, f - 1^a) \right) \\
&+ p_c(w, f) \sum_{h \in G^{k'-1}_{f-1^c,a}} p_h(w, f - 1^c) \\
&+ \sum_{c' \in C \setminus a : f_{c'} \neq 0} p_{c'}(w, f) a_c^{k-1}(w, f - 1^{c'}) && (148) \\
= &p_c(w, f) \sum_{h \in G^{k'-1}_{f-1^c,a}} p_h(w, f - 1^c) \\
&+ \sum_{c' \in C : f_{c'} \neq 0} p_{c'}(w, f) a_c^{k-1}(w, f - 1^{c'}) && (149)
\end{aligned}
$$

By Lemma 42:

$$
\begin{aligned}
\sum_{h \in G^{k'}_{f,a}} \text{count}(h(k), c) p_h(w, f) = &p_c(w, f) + \sum_{c' \in C : f_{c'} \neq 0} p_{c'}(w, f) a_c^{k-1}(w, f - 1^{c'}) && (150) \\
\sum_{h \in G^{k'}_{f,a}} \text{count}(h(k), c) p_h(w, f) = &a_c^k(w, f) && (151)
\end{aligned}
$$

■

Assume that h is a subsequence of h'. Define $C_{h,i}(h')$ to indicate whether the ith element of h is before or at the vth position of h' (1 if true, 0 if false).

Theorem 44 *For any $a' \neq a$, if A is the set of pairs $h, i \in (H^k_{f,a} \times \mathbf{Z})$ where $h_i = a'$, then the expected number of impressions of a' can be written:*

$$a^k_{a'}(S) = \sum_{(h,i)\in A} \sum_{h'\in \mathrm{Ins}^*{}_{f_a}(h,a)} p_{h'}(S) C_{h,i}(h') \tag{152}$$

Proof: By Theorem 43,

$$a^k_{a'}(S) = \sum_{h\in G^k_{f,a}} p_h(S)\mathrm{count}(h(k), a') \tag{153}$$

$$a^k_{a'}(S) = \sum_{h\in H^k_{f,a}} \sum_{h'\in \mathrm{Ins}^*{}_{f_a}(h,a)} p_{h'}(S)\mathrm{count}(h'(k), a') \tag{154}$$

Define $A(h) = \{i : h_i = a'\}$. Note that $\mathrm{count}(h', a') = \sum_{i\in A(h)} C_{h,i}(h')$. Thus:

$$a^k_{a'}(S) = \sum_{h\in H^k_{f,a}} \sum_{i\in A(h)} \sum_{h'\in \mathrm{Ins}^*{}_{f_a}(h,a)} p_{h'}(S) C_{h,i}(h') \tag{155}$$

$$a^k_{a'}(S) = \sum_{h,i\in A} \sum_{h'\in \mathrm{Ins}^*{}_{f_a}(h,a)} p_{h'}(S) C_{h,i}(h') \tag{156}$$

■

Suppose we were to change the weight of a, but only on the first visit. In particular, we fixed it such that $p_a(S') = \alpha p_a(S)$. If we leave the other masses the same, then for $a' \neq a$, $p_{a'}(S') = \beta p_a(S')$, where $\beta(1 - p_a(S')) + \alpha p_{S'}(a) = 1$. Define

$$\Delta_{a'} = a^k_{a'}(S') - a^k_{a'}(S), \tag{157}$$

$$\Delta_{h,i} = \sum_{h'\in \mathrm{Ins}^*{}_{f_a}(h,a)} (p_{h'}(S') - p_h(S)) C_{h,i}(h'). \tag{158}$$

Thus, given A defined as in Theorem 44, by that theorem:

$$\Delta_{a'} = \sum_{(h,i)\in A} \Delta_{h,i} \tag{159}$$

We wish to prove that $\Delta_{a'}$ is non-positive if $\alpha > 1$. We can do this by proving that $\Delta_{h,i}$ is non-positive. In particular, what we will do is show that for any history $h \in H^k_{f,a}$, for any $h' \in \mathrm{Ins}^*{}_{f_a}(h,a)$, that begins with a, we can find a set of histories in $H' \subseteq \mathrm{Ins}^*{}_{f_a}(h,a)$ that do not begin with a such that for all $i \in \{1 \ldots |h|\}$, for all $h'' \in H'$, $C_{h,i}(h') \leq C_{h,i}(h'')$.

Define $M_{i,j}(h)$ such that if $\mathrm{count}(h,a) < j - i$, then $M_i(h) = \{h\}$. If $\mathrm{count}(h,a) = j - i$, then if $h = h' \circ h''$, where $\mathrm{count}(h'', a) = 0$ and either $|h'| = 0$ or $h'_{|h'|} = a$, then $M_{i,j}(h) = h' \circ \mathrm{Ins}^*{}_i(h'', a)$.

Lemma 45 *Given $h \in H^k_{f,a}$, and $h' = a \circ \ldots \circ a$, where $|h'| = i$, and $h'' \in H^k_{f-i1^a}$, and $h''' = h' \circ h''$, then for any $h^{iv} \in M_{i,f_a}(h''')$, for all $j \in \{1 \ldots |h|\}$, $C_{h,j}(h''') \leq C_{h,j}(h^{iv})$.*

Proof: For any $j \in \{1 \ldots |h|\}$, for any $h^v \in \mathrm{Ins}^*{}_{f_a}(h,a)$ define $X_{h,j}(h^v)$ to be the position of the jth element of h in h^v. For any j, $X_{h,i}(h'') = X_{h,i}(h''') - i$. Since less than or equal to i as were inserted into h''' to form h^{iv}, then $X_{h,i}(h^{iv}) \leq X_{h,i}(h'') + i$. Therefore, $X_{h,i}(h^{iv}) \leq X_{h,i}(h''')$. This implies that if $X_{h,i}(h''') \leq v$, $X_{h,i}(h^{iv}) \leq v$, so $C_{h,i}(h''') \leq C_{h,i}(h^{iv})$. ■

Secondly, we must show that most often, the elements of $M_{i,j}(h)$ do not begin with a.

Lemma 46 *Given $h \in H^k_{f,a}$, and $h' = a \circ \ldots \circ a$, where $|h'| = i < f_a$, and $h'' \in H^k_{f-i1^a}$, where $h''_1 \neq a$, and $h''' = h' \circ h''$, then for any $h^{iv} \in M_{i,f_a}(h''')$, $h^{iv}_1 \neq a$.*

Proof: This can be broken down into two cases. First, if $\text{count}(h'', a) = 0$, then $M_{i,f_a}(h'') = h''$. and so $h^{iv}_1 = h''_1 \neq a$. Secondly if $\text{count}(h'', a) > 0$, then no as will be inserted before (at least) the first a. In this case as well, $h^{iv}_1 \neq a$. ∎

Of course, if $i = f_a$, there *is* an element of $M_{i,f_a}(h''')$ which starts with a. Therefore, for (h, i) where $i + f_a \leq v$, we need a special theorem (Theorem 57) which does not leverage $M_{i,j}$ directly. Now, if $i + f_a > v$, then all histories h' that start with f_a as will have $C_{h,i}(h') = 0$, so there will not be a problem. But, before we begin, we have to understand $M_{i,j}$ and the probability mass on various sets of histories.

For $h \in H^k_{f,a}$, for all $h' \in \text{Ins}^*{}_{f_a}(h, a)$, define $\text{split}_i(h', a)$ to be:

1. 0 if $h' = h$ and there are no zeros.
2. index of last a in h' if $\text{count}(h', a) < i$.
3. index of ith occurrence of a if $\text{count}(h', a) \geq i$.

Lemma 47 *If $f_c = 1$, then: $p_{c'}(w, f - 1^c) = p_c(w, f)p_{c'}(w, f - 1^c) + p_{c'}(w, f)$*

Proof: Define $S = \sum_{c'' \neq c : f_{c''} \neq 0} w_{c'}$. Then:

$$p_c(w, f)p_{c'}(w, f - 1^c) + p_{c'}(w, f) = \frac{w_c}{w_c + S}\frac{w_{c'}}{S} + \frac{w_{c'}}{w_c + S} \tag{160}$$

$$p_c(w, f)p_{c'}(w, f - 1^c) + p_{c'}(w, f) = \frac{w_{c'}}{w_c + S}\left(\frac{w_c}{S} + 1\right) \tag{161}$$

$$p_c(w, f)p_{c'}(w, f - 1^c) + p_{c'}(w, f) = \frac{w_{c'}}{w_c + S}\left(\frac{w_c + S}{S}\right) \tag{162}$$

$$p_c(w, f)p_{c'}(w, f - 1^c) + p_{c'}(w, f) = \frac{w_{c'}}{S} \tag{163}$$

$$p_c(w, f)p_{c'}(w, f - 1^c) + p_{c'}(w, f) = p_{c'}(w, f - 1^c). \tag{164}$$

∎

Lemma 48 *If $|h| \geq 1$, $\text{count}(h, c) = 0$ and $f_c = 1$, then $p_h(w, f - 1^c) = \sum_{h' \in M_{1,1}(h)} p_{h'}(w, f)$, or expanding the expression, $p_h(w, f) + \sum_{i=1}^{|h|} p_{\text{Ins}(h,c,i)}(w, f) = p_h(w, f - 1^c)$.*

Proof: We prove this by induction on the length of h. First, if $|h| = 1$, then by Lemma 47, we can handle the base case. Assume that the result holds for $k - 1$. Choose an arbitrary $h' \in C^{k-1}$ where $\text{count}(h, c) = 0$, and a $c' \in C \backslash c$. We wish to prove the result holds for $h = c' \circ h' = \text{Ins}(h', c', 1)$. First, if $f'_c = 0$, then $p_h(w, f) = 0$ and for all $i \in 1, \ldots |h|$, $p_{\text{Ins}(h,c,i)} = 0$. So, without loss of generality, assume $f_{c'} \neq 0$.

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_h(w,f) + p_{\mathrm{Ins}(h,c,1)}(w,f) + \sum_{i=2}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) \quad (165)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_{c'\circ h'}(w,f) + p_{c\circ h}(w,f) + \sum_{i=2}^{|h|} p_{c'\circ \mathrm{Ins}(h',c,i-1)}(w,f) \quad (166)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_{c'\circ h'}(w,f) + p_{c\circ h}(w,f) + \sum_{i=1}^{|h|-1} p_{c'\circ \mathrm{Ins}(h',c,i)}(w,f) \quad (167)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_{c'}(w,f)p_{h'}(w,f-1^{c'}) + p_c(w,f)p_h(w,f-1^c) + \sum_{i=1}^{|h|-1} p_{c'}(w,f)p_{\mathrm{Ins}(h',c,i)}(w,f-1^{c'}) \quad (168)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_c(w,f)p_h(w,f-1^c) + p_{c'}(w,f)\left(p_{c'}(w,f) + \sum_{i=1}^{|h|-1} p_{\mathrm{Ins}(h',c,i)}(w,f-1^{c'})\right) \quad (169)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_c(w,f)p_h(w,f-1^c) + p_{c'}(w,f)p_{h'}(w,f-1^{c'}) \quad (170)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_c(w,f)p_{c'}(w,f-1^c)p_{h'}(w,f-1^c-1^{c'}) + p_{c'}(w,f)p_{h'}(w,f-1^{c'}) \quad (171)$$

$$p_h(w,f) + \sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = (p_c(w,f)p_{c'}(w,f-1^c) + p_{c'}(w,f))p_{h'}(w,f-1^c-1^{c'}) \quad (172)$$

$$(173)$$

By Lemma 47:

$$\sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_{c'}(w,f-1^c)p_{h'}(w,f-1^c-1^{c'}) \quad (174)$$

$$\sum_{i=1}^{|h|} p_{\mathrm{Ins}(h,c,i)}(w,f) = p_{c'\circ h'}(w,f-1^c) \quad (175)$$

■

Lemma 49 *For all $h \in H^k_{f,a}$, for all $h' \in {\mathrm{Ins}^*}_{f_a-1}(h',a)$:*

$$p_{ah'}(S) = p_a(S) \sum_{h'' \in M_{1,f_a}(h')} p_{h''}(S) \quad (176)$$

Proof: First, observe that:

$$p_{a\circ h'}(w,f) = p_a(w,f)p_{h'}(w,f-1^a) \quad (177)$$

However, if $\mathrm{count}(h',a) \leq f_a - 1$, $p_{h'}(w,f-1^a) = p_{h'}(w,f)$. Therefore:

$$p_{ah'}(w,f) = p_a(w,f)p_{h'}(w,f). \quad (178)$$

Thus, we can assume without loss of generality that $\text{count}(h', a) = f_a - 1$. Define $h'' = h'_1 \dots h'_{\text{last}(h',a)}$, and $h''' = h'_{\text{last}(h',a)+1} \dots h'_{|h'|}$. Then:

$$p_{h'}(w, f - 1^a) = p_{h''}(w, f - 1^a) p_{h'''}(w, f - 1^a - 1^{h''}) \tag{179}$$

Define $f' = f - 1^{h''}$. Note that $f'_a = 1$. Thus, by Lemma 48:

$$p_{h'''}(w, f' - 1^a) = \sum_{h^{iv} \in M_{1,1}(h''')} p_{h^{iv}}(w, f') \tag{180}$$

$$p_{h'''}(w, f - 1^a - 1^{h''}) = \sum_{h^{iv} \in M_{1,1}(h''')} p_{h^{iv}}(w, f - 1^{h''}) \tag{181}$$

$$p_{h''}(w, f - 1^a) p_{h'''}(w, f - 1^a - 1^{h''}) = p_{h''}(w, f - 1^a) \sum_{h^{iv} \in M_{1,1}(h''')} p_{h^{iv}}(w, f - 1^{h''}) \tag{182}$$

$$p_{h'}(w, f - 1^a) = p_{h''}(w, f - 1^a) \sum_{h^{iv} \in M_{1,1}(h''')} p_{h^{iv}}(w, f - 1^{h''}) \tag{183}$$

Now, observe that since h'' ends with an a and $\text{count}(h'', a) = f_a - 1$, $p_{h''}(w, f - 1^a) = p_{h''}(w, f)$. Therefore:

$$p_{h'}(w, f - 1^a) = p_{h''}(w, f) \sum_{h^{iv} \in M_{1,1}(h''')} p_{h^{iv}}(w, f - 1^{h''}) \tag{184}$$

$$p_{h'}(w, f - 1^a) = \sum_{h^{iv} \in M_{1,1}(h''')} p_{h'' \circ h^{iv}}(w, f) \tag{185}$$

$$p_{h'}(w, f - 1^a) = p_{h'' \circ h'''}(w, f - 1^{h''}) + \sum_{i=1}^{|h'''|} p_{h'' \circ \text{Ins}(h''', a, i)}(w, f) \tag{186}$$

$$p_{h'}(w, f - 1^a) = \sum_{h^{iv} \in M_{1,f_a}(h')} p_{h^{iv}}(w, f) \tag{187}$$

$$p_{h'}(w, f - 1^a) = \sum_{h^{iv} \in M_{1,f_a}(h')} p_{h^{iv}}(w, f) \tag{188}$$

■

Lemma 50 *For all $h \in H^k_{f,a}$, for all $h' \in \text{Ins}^*{}_{f_a}(h, a)$, there exists an $h'' \in \text{Ins}^*{}_{f_a - i}(h, a)$ where $h' \in M_i(h'')$.*

Proof: Define $h'' \in \text{Ins}^*{}_{f_a - i}(h, a)$ to be a history of length $\max(\text{count}(h', a), f_a - i) + |h|$ where:

$$h''_t = \begin{cases} h'_t & \text{if } t \leq \text{split}_{f_a - i}(h') \\ h_{t+|h|-|h'|} & \text{otherwise} \end{cases}. \tag{189}$$

Informally, h'' starts like h' and ends like h. Note that $\text{count}(h'', a) \leq f_a - i$. If $\text{count}(h'', a) < f_a - i$, then $\text{count}(h', a) = \text{count}(h'', a)$, so $h'' = h'$ and $h' \in M_i(h'')$. If $\text{count}(h'', a) = f_a - i$, then h' and h'' are equal up until $\text{split}_{f_a - i}(h')$, and then h' might have more as inserted. So in this case as well, $h' \in M_i(h'')$. ■

Lemma 51 *For all $h \in H^k_{f,a}$, for all $h', h'' \in \text{Ins}^*{}_{f_a - i}(h, a)$ where $h' \neq h''$, it is the case that $M_{i,f_a}(h') \cap M_{i,f_a}(h'') = \emptyset$.*

Corollary 52 *$\{M_{i,f_a}(h') : h' \in \text{Ins}^*{}_{f_a - i}(h, a)\}$ is a partition of $\text{Ins}^*{}_{f_a}(h, a)$.*

Proof: Note that if $v \in M_i(h')$, then we can recover h' by the construct in Lemma 50. ■

Finally, we can apply the corollary three times to get:

Lemma 53 *For all $h \in H^k_{f,a}$, for any function g,*

$$\sum_{h' \in \text{Ins}^*{}_i(h,a)} g(h') = \sum_{h''' \in \text{Ins}^*{}_{f_a - i}(h,a)} \sum_{h'' \in M_{i,f_a}(h''')} g(h'') \tag{190}$$

$$\sum_{h' \in \text{Ins}^*{}_i(h,a)} g(h') = \sum_{h''' \in \text{Ins}^*{}_{f_a - i}(h,a)} \sum_{h'' \in M_{i-1,f_a-1}(h''')} \sum_{h' \in M_{1,f_a}(h'')} g(h') \tag{191}$$

Proof: Equation 190 is a direct result of Corollary 52, and the fact that, if $\{X_i\}_i$ is a partition of Y, then $\sum_{y \in Y} g(y) = \sum_i \sum_{x \in X_i} g(x)$. Moreover, Equation (191) is a product of Corollary 52 and the partition property both applied twice. ∎

Lemma 54 *For $S = (w, f)$, if $h'' = a \dots a$, and $|h''| = i \leq f_a$, then for all $h \in H^k_{f,a}$, for all $h' \in \text{Ins}^*{}_{f_a - i}(h,a)$:*

$$p_{h'' \circ h'}(S) = p_a(S)^i \sum_{h''' \in M_i(h')} p_{h'''}(S) \tag{192}$$

Proof: If $\text{count}(h', a) < f_a - i$, then

$$p_{h'' \circ h'}(S) = p_{h''}(w, f) p_{h'}(w, f - i1^a) \tag{193}$$

$$= p_{h''}(w, f) p_{h'}(w, f) \tag{194}$$

$$= p_a(S)^i \sum_{h''' \in M_i(h')} p_{h'''}(()w, f). \tag{195}$$

Thus, without loss of generality, we can assume that $\text{count}(h', a) = f_a - i$. We prove the result by induction on i. Observe that for $i = 1$, the lemma is the same as Lemma 49. Thus, assume that it holds for $i - 1$. Then:

$$p_{ah'}(w, f - (i-1)1^a) = \sum_{h''' \in M_{i-1,f_a}(ah')} p_{h'''}(w, f) \tag{196}$$

$$p_a(w, f - (i-1)1^a) p_{h'}(w, f - i1^a) = \sum_{h''' \in M_{i-1,f_a}(ah')} p_{h'''}(w, f) \tag{197}$$

$$p_a(w, f - (i-1)1^a) p_{h'}(w, f - i1^a) = \sum_{h''' \in M_{i-1,f_a-1}(h')} p_{ah'''}(w, f) \tag{198}$$

$$p_a(w, f - (i-1)1^a) p_{h'}(w, f - i1^a) = p_a(w, f) \sum_{h''' \in M_{i-1,f_a-1}(h')} p_{h'''}(w, f - 1^a) \tag{199}$$

Because $f^a > i - 1$, $p_a(w, f - (i-1)1^a) = p_a(w, f)$, so:

$$p_{h'}(w, f - i1^a) = \sum_{h''' \in M_{i-1,f_a-1}(h')} p_{h'''}(w, f - 1^a) \tag{200}$$

By Lemma 49:

$$p_{h'}(w, f - i1^a) = \sum_{h''' \in M_{i-1,f_a-1}(h')} p_{h'''}(w, f - 1^a) \tag{201}$$

$$p_{h'}(w, f - i1^a) = \sum_{h''' \in M_{i-1,f_a-1}(h')} \sum_{h^{iv} \in M_{1,f_a}(h''')} p_{h^{iv}}(w, f) \tag{202}$$

By Lemma 53:

$$p_{h'}(w, f - i1^a) = \sum_{h''' \in M_{i,f_a}(h')} p_{h'''}(w, f) \tag{203}$$

∎

Theorem 55 *If S is a mass model, and if a does not occur in h, then:*

$$\sum_{h' \in \text{Ins}^*{}_{f_a-1}(h,a)} p_{a\circ h}(S) = p_a(S) \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)} p_{h'}(S). \tag{204}$$

Corollary 56

$$\frac{1-p_a(S)}{p_a(S)} \sum_{h' \in a\circ\text{Ins}^*{}_{f_a-1}(h,a)} p_{a\circ h'}(S) = \sum_{h' \in \text{Ins}^*{}_1(h,a)\setminus a\circ\text{Ins}^*{}_{f_a}(h,a)} p_{h'}(S) \tag{205}$$

Proof: By Lemma 49, for all $h' \in \text{Ins}^*{}_{f_a-1}(h,a)$:

$$p_{ah'}(S) = p_a(S) \sum_{h'' \in M_1(h')} p_{h''}(S) \tag{206}$$

$$\sum_{h' \in \text{Ins}^*{}_{f_a-1}(h,a)} p_{ah'}(S) = p_a(S) \sum_{h' \in \text{Ins}^*{}_{f_a-1}(h,a)} \sum_{h'' \in M_1(h')} p_{h''}(S) \tag{207}$$

By Corollary 52:

$$\sum_{h' \in \text{Ins}^*{}_{f_a-1}(h,a)} p_{ah'}(S) = p_a(S) \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)} p_{h'}(S) \tag{208}$$

With the lemma proven, the proof of the corollary is straightforward arithmetic.

$$\sum_{h' \in a\circ\text{Ins}^*{}_0(h,a)} p_{a\circ h'}(S) = p_a(S) \sum_{h' \in \text{Ins}^*{}_1(h,a)} p_{h'}(S) \tag{209}$$

$$\sum_{h' \in a\circ\text{Ins}^*{}_0(h,a)} p_{a\circ h'}(S) = p_a(S) \sum_{h' \in a\circ\text{Ins}^*{}_0(h,a)} p_{h'}(S) + p_a(S) \sum_{h' \in \text{Ins}^*{}_1(h,a)\setminus a\circ\text{Ins}^*{}_0(h,a)} p_{h'}(S) \tag{210}$$

$$\frac{1-p_a(S)}{p_a(S)} \sum_{h' \in a\circ\text{Ins}^*{}_0(h,a)} p_{a\circ h'}(S) = \sum_{h' \in \text{Ins}^*{}_1(h,a)\setminus a\circ\text{Ins}^*{}_0(h,a)} p_{h'}(S) \tag{211}$$

■

Theorem 57 *If $i + f_a \leq v$, then $\Delta_{h,i} = 0$.*

Proof: Note that if $i + f_a \leq v$, then for any $h' \in \text{Ins}^*{}_{f_a}(h,a) = \{h\}$, $C_{h,i}(h') = 1$. Thus, what is left to show is that $\sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)} p_{h'}(S)$ and $\sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)} p_{h'}(S')$ are equal:

So, we can write:

$$\begin{aligned}
\Delta_{h,i} &= \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)} (p_{h'}(S') - p_{h'}(S)) C_{h,i}(h') && (212)\\
&= \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)} (p_{h'}(S') - p_{h'}(S)) && (213)\\
&= \sum_{h' \in a\circ\text{Ins}^*{}_{f_a-1}(h,a)} (p_{h'}(S') - p_{h'}(S)) + \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)\setminus a\circ\text{Ins}^*{}_{f_a-1}(h,a)} (p_{h'}(S') - p_{h'}(S)) && (214)\\
&= \sum_{h' \in a\circ\text{Ins}^*{}_{f_a-1}(h,a)} (\alpha p_{h'}(S) - p_{h'}(S)) + \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)\setminus a\circ\text{Ins}^*{}_{f_a-1}(h,a)} (\beta p_{h'}(S) - p_{h'}(S)) && (215)\\
&= (\alpha - 1) \sum_{h' \in a\circ\text{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S) + (\beta - 1) \sum_{h' \in \text{Ins}^*{}_{f_a}(h,a)\setminus a\circ\text{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S) && (216)\\
& && (217)
\end{aligned}$$

By Corollary 56:

$$\Delta_{h,i} = (\alpha-1)\sum_{h'\in a\circ \mathrm{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S) + (\beta-1)\frac{1-p_a(S)}{p_a(S)}\sum_{h'\in a\circ \mathrm{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S) \tag{218}$$

$$\Delta_{h,i} = \left(\alpha-1+(\beta-1)\frac{1-p_a(S)}{p_a(S)}\right)\sum_{h'\in a\circ \mathrm{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S) \tag{219}$$

$$= \left(\alpha-1+\beta\frac{1-p_a(S)}{p_a(S)}-\frac{1-p_a(S)}{p_a(S)}\right)\sum_{h'\in a\circ \mathrm{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S) \tag{220}$$

By the definition of β, $\beta(1-p_a(S)) = 1-\alpha p_a(S)$, so:

$$\Delta_{h,i} = \left(\alpha-1+\frac{1-\alpha p_a(S)}{p_a(S)}-\frac{1-p_a(S)}{p_a(S)}\right)\sum_{h'\in a\circ \mathrm{Ins}^*{}_{f_a-1}(h,a)} p_{h'}(S). \tag{221}$$

$$\Delta_{h,i} = 0. \tag{222}$$

■

Theorem 58 *If $i + f_a > v$, then $\Delta_{h,i} \leq 0$.*

Proof: First, define $T = \{h' \in \mathrm{Ins}^*{}_{f_a}(h,a) : h'_1 = a, C_{h,i}(h') = 1\}$. Define $T' = \{h' \in \mathrm{Ins}^*{}_{f_a}(h,a) : h'_1 \neq a, C_{h,i}(h') = 1\}$. Therefore:

$$\Delta_{h,i} = \sum_{h'\in \mathrm{Ins}^*{}_{f_a}(h,a)} (p_{h'}(S') - p_{h'}(S))C_{h,i}(h') \tag{223}$$

$$= \sum_{h'\in T}(p_{h'}(S') - p_{h'}(S)) + \sum_{h'\in T'}(p_{h'}(S') - p_{h'}(S)) \tag{224}$$

Since each history in T begins with a, and each history in T' does not begin with a:

$$\Delta_{h,i} = \sum_{h'\in T}(\alpha p_{h'}(S) - p_{h'}(S)) + \sum_{h'\in T'}(\beta p_{h'}(S) - p_{h'}(S)) \tag{225}$$

$$\Delta_{h,i} = (\alpha-1)\sum_{h'\in T} p_{h'}(S) + (\beta-1)\sum_{h'\in T'} p_{h'}(S) \tag{226}$$

So, we need to measure the relative sizes of the probability mass in T and T'. First, we split T into T_1, T_2, T_3, such that $h' \in T$ is in T_m if $h_j = a$ if $j \leq m$ and $h_{j+1} \neq a$. Thus, for any $h' \in T_m$, we can break it down into $h'' = a \circ \ldots \circ a$, where $|h''| = m$, and $h''' = h'_{m+1} \ldots h'_{|h'|}$. Define $\mathrm{suffix}(h', j) = h'_j \ldots h'_{|h'|}$, so that $h''' = \mathrm{suffix}(h', m)$. Note that since $C_{h,i}(h''') = 1$ and $i > v - f_a$, $m < f_a$, so by Lemma 46, $M_{m,f_a}(h''') \subseteq T'$. Moreover, given a second history $\tilde{h}' \in T_m$, we know from 51 that $M_{m,f_a}(\mathrm{suffix}(h', m)) \cap M_{m,f_a}(\mathrm{suffix}(\tilde{h}', m)) = \emptyset$, so:

$$\sum_{h'\in T'} p_{h'}(S) \geq \sum_{h'\in T_m}\sum_{h''\in M_{m,f_a}(\mathrm{suffix}(h',m))} p_{h''}(S). \tag{227}$$

Now, from Lemma 54, $p_{h'}(S) = (p_a(S))^m \sum_{h''\in M_{m,f_a}(\mathrm{suffix}(h',m))} p_{h''}(S)$, so:

$$\sum_{h'\in T'} p_{h'}(S) \geq \sum_{h'\in T_m} \frac{p_{h'}(S)}{(p_a(S))^m} \tag{228}$$

$$(p_a(S))^m \sum_{h'\in T'} p_{h'}(S) \geq \sum_{h'\in T_m} p_{h'}(S) \tag{229}$$

Since $T_1 \dots T_{f_a-1}$ is a partition of T:

$$\Delta_{h,i} = (\alpha-1)\sum_{m=1}^{f_a-1}\sum_{h'\in T_m} p_{h'}(S) + (\beta-1)\sum_{h'\in T'} p_{h'}(S) \tag{230}$$

$$\Delta_{h,i} \leq (\alpha-1)\sum_{m=1}^{f_a-1}(p_a(S))^m \sum_{h'\in T'} p_{h'}(S) + (\beta-1)\sum_{h'\in T'} p_{h'}(S) \tag{231}$$

$$\Delta_{h,i} \leq \left((\beta-1) + (\alpha-1)\sum_{m=1}^{f_a-1}(p_a(S))^m\right)\sum_{h'\in T'} p_{h'}(S) \tag{232}$$

$$\Delta_{h,i} \leq \left((\beta-1) + (\alpha-1)\frac{p_a(S)-(p_a(S))^{m+1}}{1-p_a(S)}\right)\sum_{h'\in T'} p_{h'}(S) \tag{233}$$

$$\Delta_{h,i} \leq \left((\beta-1) + (\alpha-1)\frac{p_a(S)}{1-p_a(S)}\right)\sum_{h'\in T'} p_{h'}(S) \tag{234}$$

$$\Delta_{h,i} \leq \left((\beta-1) + \alpha\frac{p_a(S)}{1-p_a(S)} - \frac{p_a(S)}{1-p_a(S)}\right)\sum_{h'\in T'} p_{h'}(S) \tag{235}$$

From this point, we continue in a way similar to Equation (219). By the definition of β, $\alpha(p_a(S)) = 1-\beta(1-p_a(S))$, so:

$$\Delta_{h,i} \leq \left((\beta-1) + \frac{1-\beta(1-p_a(S))}{1-p_a(S)} - \frac{p_a(S)}{1-p_a(S)}\right)\sum_{h'\in T'} p_{h'}(S) \tag{236}$$

$$\Delta_{h,i} \leq \left((\beta-1) + \frac{(1-\beta)(1-p_S(a))}{1-p_a(S)}\right)\sum_{h'\in T'} p_{h'}(S) \tag{237}$$

$$\Delta_{h,i} \leq ((\beta-1)+(1-\beta))\sum_{h'\in T'} p_{h'}(S) \tag{238}$$

$$\Delta_{h,i} \leq 0 \times \sum_{h'\in T'} p_{h'}(S) \tag{239}$$

$$\Delta_{h,i} \leq 0 \tag{240}$$

■

Theorem 59 *If $a' \neq a$, then $a^k_{a'}(S') \leq a^k_{a'}(S)$.*

Proof: Since $a^k_{a'}(S') \leq a^k_{a'}(S) = \Delta_{a'} = \sum_{(h,i)\in A}\Delta_{h,i}$, and by Theorem 57 and Theorem 58, $\Delta_{h,i} \leq 0$, so therefore $\Delta_{a'} \leq 0$, and the theorem follows. ■

Theorem 60 *If $a, a' \in C$, $a' \neq a$, and for all $w_c = v_c$ when $c \neq a$, and $v_a = \alpha w_a$ with $\alpha \geq 1$, then $a^k_{a'}(v,f) \leq a^k(w,f)$.*

Proof: We prove this by induction on k. First observe that $a^0 = 0$, so it holds for $k = 0$. For $k > 0$:

$$a^k_{a'}(v,f) = p_{a'}(v,f) + \sum_{c\in C: f_c\neq 0} p_c(v,f)a^{k-1}_{a'}(v, f-1^c). \tag{241}$$

By induction:

$$a^k_{a'}(v,f) \leq p_{a'}(v,f) + \sum_{c\in C: f_c\neq 0} p_c(v,f)a^{k-1}_{a'}(w, f-1^c). \tag{242}$$

Observe that $p_{a'}(v,f) + \sum_{c \in C: f_c \neq 0} p_c(v,f) a^{k-1}_{a'}(w, f - 1^c) = a^k_{a'}(S')$. So therefore,

$$a^k_{a'}(v,f) \leq a^k_{a'}(S') \tag{243}$$

$$a^k_{a'}(v,f) \leq a^k_{a'}(S) \tag{244}$$

$$a^k_{a'}(v,f) \leq a^k_{a'}(w,f) \tag{245}$$

■

Proof (of Lemma 16): First, observe that the problem can be separated into individual users by Theorem 11. Moreover, by Lemma 25, a represents the model of a user, given some values of f and k. Therefore, we need only prove properties of a^k_c.

Theorem 60 establishes well-behaved property 1. Lemma 39 establishes well-behaved property 3. Lemma 36 establishes well-behaved property 2. Lemma 33 establishes well-behaved property 4.

■