Is Unequal Error Protection Useful?

Ozgun Y. Bursalioglu, Giuseppe Caire
Ming Hsieh Department of Electrical Engineering
University of Southern California
Email: {bursali, caire}@usc.edu

Abstract—When transmitting source-encoded data, not all information bits are equally important, due to the different sensitivity of the source decoder to errors. Unequal Error Protection (UEP) consists of allocating coding redundancy depending on the importance of the information bits. We consider progressive transmission of source-encoded data under three different packet formats where either number of source bits per packet is fixed (fixed- k approach), or packet block length is fixed (fixed-n approach) or both parameters allowed to vary for each block (variable-(n, k) approach). Most existing results are based on some chosen family of channel codes and consider a single-user setting. Thanks to the recent finite length error probability results by Polyanskiy et al., in this work we investigate the UEP concept using the new finite-length random coding bounds. In the single-user case, we show that when codes meeting Polyanskiy achievability bounds are used, UEP does not obtain significant advantages over Equal-Error Protection (EEP) (advantages disappear for the variable-(n, k) case). Based on these results, a low complexity optimization algorithm is proposed for the multiuser (multicast) scenario.

I. INTRODUCTION

While Shannon’s separation theorem makes of bits a universal currency into which any source can be converted in order to be transmitted reliably through a noisy channel, it is well-known that in certain lossy transmission settings “not all bits are created equal”. In particular, analog sources (e.g., video) are typically encoded via an “embedded” successive refinement scheme, and transmitted progressively: any additional packet of bits successfully decoded yields an improvement in the reconstruction distortion till the first block in error, after which the remaining portion of the sequence is useless to the embedded source decoder. Therefore, the first packets in the sequence (fundamental layer) are “more important” than the last ones (refinement).

In this context, Unequal Error Protection (UEP) consists of allocating coding redundancy over the sequence of progressively transmitted packets, depending on the importance of the information bits. The goal is to minimize the expected distortion at reconstruction. Since the minimization of the expected distortion is typically involved, and for most practical sources the embedded source coder operational rate-distortion function is unknown, a widely proposed approach consists of maximizing the expected number of correctly decoded packets up to the first packet in error. We shall refer to this optimization criterion as the maximum average rate case.

This problem has been widely studied in the literature [1], [2], [3], [4]. A packet of size n corresponds to n coded channel symbols that are obtained from a block of k information bits. We consider a memoryless noisy channel, operating on the coded symbols (not on the packets). In literature, mainly two different packet formats have been considered: variable-length channel packets with fixed-length information block (fixed-k approach) [1], fixed-length channel packets with variable length information block (fixed-n approach) [2]. Another version of the UEP problem, not so widely explored, considers variable k, n for different packets (variable-(n, k) approach) [5], see Fig. 1. Most existing results are based on some chosen family of codes (e.g., shortened Reed-Solomon codes). Thanks to recent results by Polyanskiy et al. [6], in this work we investigate the UEP concept using the new finite length random coding bounds and approximations. We show that when good finite length codes are used (i.e., codes that meet Polyanskiy achievability bounds), UEP after all does not obtain significant advantages over Equal-Error Protection.

![Packet structures of different UEP schemes considered](image)

Figure 1. Packet structures of different UEP schemes considered

We denote by qk,n the block error probability of codes of length n and rate k/n. 1 In addition to new, generally tighter than what previously known, achievability and converse bounds for the performance of codes at finite length, [6] also developed the accurate and easy to use approximation:

\[
\frac{k}{n} \approx C - \sqrt{\frac{V}{n}} Q^{-1}(q_{k,n}),
\]

where C denotes channel capacity and V denotes channel dispersion [6].

In [1], [2], [5] (as well as in several other works not referenced here for the sake of brevity), UEP is studied for the single-source single-destination case with the purpose of limiting the effects of residual post-decoding block errors due to finite block length coding. In fact, if arbitrarily large

1We use \(\Delta q_{k,n} = 1 - q_{k,n}\) throughout the rest of the paper.
block length and coding complexity could be afforded, coding arbitrarily close to Shannon’s separation limit would be optimal with no need of UEP. Different scenarios including fading channels, multiple sources, and parallel channels have been considered in [3], [7], and [8], respectively. In general, even for the maximum average rate case, optimal UEP rate allocation is obtained as the solution of a rather complicated dynamic program, which is hard to solve in an on-line real-time implementation [1], [2], [5]. Therefore, simplifications have been sought [3], [4]. Nosratinia et al. worked with an empirical model relating block error probability with Binary Symmetric Channel (BSC) capacity, somehow reducing the complexity of finding the optimal solution due to the structure of empirical error function. Simulations with fading channels have also been provided where the UEP scheme is designed according to the worst channel condition [3].

Among many variations of the UEP in literature, the multicast version, i.e., one source sent to different users through channels with different capacity, has not been considered in detail. In this work, for the fixed-\(k\) and fixed-\(n\) approaches, we show that a suitably optimized UEP scheme, where all packets size and rate are the same, is competitive with the high complexity optimal dynamic programming solution of UEP. We also show that, for the variable-\((n,k)\) case, using a single block is optimal. Based on these result for the single user case, we propose a near-optimal, yet simple, heuristic method for the much more involved UEP multicast problem.

II. THE MULTICAST VARIABLE-\((n,k)\) PROBLEM

UEP algorithms presented in the literature focus on rate-based and/or distortion-based optimization. It was observed that for progressive transmission schemes with fixed packet length (fixed-\(k\) or fixed-\(n\) approaches), the UEP optimization based on rate performs almost as good as the optimization based on distortion and yields lower complexity rate allocation algorithms [1], [2], [9]. This can be explained by observing that these schemes operate with a random number of correctly decoded packets till the first error. The number of decoded packets fluctuates around is average. With respect to these (small) local fluctuations, the source coder rate-distortion function is nearly linear, and therefore maximizing the average rate is nearly equivalent to minimizing the average distortion. This may not be true for the case of variable-\((k,n)\), where encoding over large blocks (up to a single large block) is allowed. In this case, for a sufficiently convex rate-distortion function with large “Jensen’s inequality penalty”, optimizing the average rate may be significantly suboptimal. For the sake of brevity, here we focus on the maximum average rate approach and investigate also the minimum average distortion case in [10].

The derivations and notations in this section are heavily based on [1]. We consider the general multicast variable-\((n,k)\) problem where a source is transmitted to \(L\) users. We assume that the \(L\) binary-input channels are degraded versions of the same channel “type” (e.g., they are BSCs with different cross-over probabilities, or Binary Erasure Channels (BECs) with different erasure probabilities). Furthermore, the channels are sorted in order of increasing capacity \(C_1 \leq \cdots \leq C_L\). Optimization is performed over all optimal finite length codes instead of a chosen code family as in previous literature.

Let \(\pi \triangleq \{ (k_1, n_1), \ldots, (k_{M(\pi)}, n_{M(\pi)}) \} \) denote a transmission policy with \(M(\pi)\) packets, where \(\sum_{i=1}^{M(\pi)} n_i = N\), and where \(N\) is the total number of channel symbols that we wish to transmit. By the results of [6], we know that there exist codes of block length \(n\) with \(k\) information bits and block error probability \(q_{k,n}\), approximately obeying relation (1). Let \(P_{\pi}^l(\pi)\) denote the probability that the first error happened at the \((i+1)^{\text{th}}\) packet given that \(t - 1\) packets are decoded correctly. For simplicity of exposition, we assume \(L = 2\) from now on. Then, \(P_{\pi}^l(\pi)\) can be calculated as follows for \(l \in \{1,2\}\):

\[
P_{\pi}^l(\pi) = \begin{cases} q_{k_i, n_i}^l \prod_{j=t}^{M(\pi)} q_{k_j, n_j} & i = t - 1; \\ \prod_{j=t}^{M(\pi)} q_{k_j, n_j} & i = M(\pi), \\ \prod_{j=t}^{\bar{\pi}} q_{k_j, n_j} & \text{else,} \end{cases}
\]

(2)

where \(q_{k,n}^l\) is the block error rate for channel \(l = 1, 2\). It is easy to see that

\[
P_{\pi}^l(\pi) = q_{k_i, n_i}^l P_{\pi}^i(\pi), \quad \sum_{i=t-1}^{M(\pi)} P_{\pi}^i(\pi) = 1, \forall l.
\]

(3)

Packet are transmitted sequentially. Some packets shall be used by both users and some other packets will be used only by the better user, depending on the UEP allocation. For the \(t^{\text{th}}\) channel, the average number of source symbols received before a packet decoding failure can be written as in (4):

\[
V_\pi^t = \sum_{i=0}^{M(\pi)} P_{\pi}^i(\pi) \left( \sum_{j=1}^{k_i} k_j \right)
\]

(4)

Then, we can state the multicast variable-\((n,k)\) optimization problem as:

\[
\max_\pi \alpha V_\pi^1 + \beta V_\pi^2, \text{ subject to } \sum_{i=1}^{M(\pi)} n_i \leq N,
\]

(5)

where \(\alpha\) and \(\beta\) are suitable nonnegative “priority” weights that parameterize the tradeoff between the two users.

Next, a dynamic programming framework is given, following in the footsteps of [1]. Define

\[
\Delta(t, \alpha, \beta, \pi) \triangleq \sum_{i=t}^{M(\pi)} \left( \sum_{j=t}^{i} k_j \right) \left[ \alpha P_{\pi}^1(t-1) + \beta P_{\pi}^2(t-1) \right],
\]

(6)

then \(\Delta(1, \alpha, \beta, \pi) = \alpha V_\pi^1 + \beta V_\pi^2\). Define the more general optimization problem:

\[
\max_\pi \Delta(t, \alpha, \beta, \pi), \text{ subject to } N_{t, \pi} = \sum_{i=t}^{M(\pi)} n_i \leq N.
\]

(7)

Then optimization of (6) is given by (7) for \(t = 1\).
\[ \Delta(t, \alpha, \beta, \pi) = \begin{cases} k_{M(\pi)} \left[ a_k^1 \Delta^1_{M(\pi)} + a_k^2 \Delta^2_{M(\pi)} \right] + \beta \Delta(t+1, \alpha, \beta, \pi), & t = M(\pi); \\ k_t \left[ a_{k_t, n_t}^1 + \beta a_{k_t, n_t}^2 \right] + \Delta(t + 1, \alpha, \beta, \pi), & \text{else}. \end{cases} \] (8)

\[ \Delta^*_t(a, b, Z) = \begin{cases} 0, & 0 \leq k \leq Z \leq n; \\ \max k \left[ a_{k,n}^1 + b a_{k,n}^2 \right] + \Delta^*_t+1(k, b_{k,n}, Z - n) & 0 \leq k \leq n < Z; \\ \Delta^* \in E & \text{else}. \end{cases} \] (9)

Using (3), the relationship between \( \Delta(t, \alpha, \beta, \pi) \) and \( \Delta(t+1, \alpha, \beta, \pi) \) is given in (8).

Let \( \Delta^*_t(a, b, Z) \) denote a sequence of recursive functions indexed by \( t \) as defined in (9). Note that neither \( \Delta(t, \alpha, \beta, \pi) \) nor \( N_i \) depend on \( n_t, k_t \) for \( i < t \). Hence, \( \Delta(t, \alpha, \beta, \pi) \) is optimized over only \( n_{M(\pi)}, \ldots, n_t \) and \( k_{M(\pi)}, \ldots, k_t \). The solution of the same optimization problem for \( t+1 \) does not depend on \( n_t, k_t \). Then the optimization of \( \Delta(t, \alpha, \beta, \pi) \) over \( n_t, k_t \) can be done independently from the optimization of \( \Delta(t+1, \alpha, \beta, \pi) \). This yields that the solution of (7) is equal to \( \Delta^*_t(t, a, b, Z) \) for \( a = \alpha, b = \beta \) and \( Z = N \). Also, note that the dependence on \( t \) in (9) is unnecessary due to the form of (9), hence we can actually consider (9) without the parameter \( t \) and use the notation \( \Delta^*(a, \beta, N) \) in the rest.

Dynamic programming solutions to fixed-\( k \) fixed-\( n \) single-user (\( L = 1 \)) problems as given in [1],[2] and can be obtained as special cases of (9).

### III. Threshold Approximation Method

We approximate \( q_{k,n} \) by a step function in the \((k, n)\) plane, such that \( q_{k,n} = 1_{\{k,n\} \in \mathcal{E}} \). Notice that this means that in the plane \((k, n)\), \( q_{k,n} \) is zero for \((k, n) \in \mathcal{E}\). The region \( \mathcal{E} \) is chosen such that the actual function \( q_{k,n} \) is very small for all pairs \((k, n) \in \mathcal{E}\). In Fig. 2, \( \mathcal{E} \) region’s boundary is plotted for various \( q_{k,n} \) threshold values for BEC with erasure rate \( \delta = 0.3 \). For a fixed threshold value, we define the boundary of the region as \( k = f(n) \), and its inverse function as \( n = g(k) \). Since it is possible to have better codes when larger block lengths are used, \( f(.) \) is superadditive i.e., \( f(k_1 + k_2) \geq f(k_1) + f(k_2) \) while \( g(.) \) is subadditive. Although the approximation depends on the threshold value of the true probability of error \( q_{k,n} \), to be approximated by zero. The usefulness of the threshold approximation comes from the fact that the resulting optimization of the UEP policy \( \pi \) is very simple. Then the threshold error probability is indeed a design parameter. This selection is a task much simpler than the full dynamic programming, and requires only a one-dimensional search.

Let’s start from the threshold approximation for variable-(\( n, k \)) single-user problem. The specialization to single user can be derived using \( \alpha = 1, \beta = 0, Z = N \) in (9) as follows where \( \Delta^*(Z) \) is short-hand notation for \( \Delta^*(1, 0, Z) \): For \( N < 1 \), \( \Delta^*(N) = 0 \) otherwise,

\[ \Delta^*(N) = \max_{0 \leq k \leq N} k \cdot \Delta^*(N-n) \] (10)

Next, the threshold approximation is applied to (10). Optimization solutions using threshold approximation is denoted by \( \Gamma^* \) instead of \( \Delta^* \). If \( n < g(k) \), then \( q_{k,n} \) becomes equal to 1 hence optimization reaches its lower bound. Instead when \( n \geq g(k) \) is considered, the error probability becomes zero at \( n = g(k) \). Obviously, any \( n \) larger than \( g(k) \) cannot be optimal. Then the optimal \( n \) is equal to \( g(k) \). \( \Delta^*(N) \) is simplified to \( \Gamma^*(N) \) in (11) for \( N \geq 1 \) (Note that \( \Gamma^*(N) = 0 \) for \( N < 1 \)).

\[ \Gamma^*(N) = \max_{1 \leq k \leq N} [k + \Gamma^*(N-g(k))] \] (11)

Using induction on \( N \), i.e. assuming \( \Gamma^*(N') = f(N') \) for \( N' < N \) and super additivity of \( f(.) \), we conclude that \( \Gamma^*(N) = f(N) \) which states single block coding is optimal according to the threshold approximation similar to exact analysis in Sec. V.

Next, we focus on the two-user (\( L = 2 \)) multicast scenario. We shall see that the threshold approximation yields that we should code using at most two blocks, one decodable by both users, and one decodable by the better user only. Similar to what we have seen with the threshold approximation for single user, here also for fixed \( k \) the optimization on \( n \) in (9) is solved by only comparing \( n \) values found according two points \( n_2 = g_2(k) \) or \( n_1 = g_1(k) \), where \( g_i(.) \) and \( f_i(.) \) functions are obtained with respect to \( i^{th} \) channel. Note that \( n_1 \geq n_2 \geq k \).

If \( n_2 \) is chosen, the recursive relation (9) sets the weight of the first user to zero hence, the dynamic programming can be converted into the single user case. When \( n_1 \) is used, both users can decode successfully and their weights are multiplied by \( 1 \). Then we write (12) for multicast case analogous to (11):

\[ \Gamma^*(\alpha, \beta, N) = \max \left\{ \beta f_2(N), \max_{0 \leq k \leq N} k(\alpha + \beta) + \Gamma^*(\alpha, \beta, N-n_1) \right\} \] (12)
The proof is again based on induction on \( N \). For \( N = 1 \), the only feasible solution is single block coding. Now for \( N' < N \) assume that the multicast solution \( \Gamma^*(\alpha, \beta, N') \) is given by a two-block structure where the first \( k \) source bits are encoded in the first package, again here \( k \) is a design parameter. Then the induction hypotheses for \( N' < N \) is as follows:

\[
\Gamma^*(\alpha, \beta, N') = \max_{0 \leq k \leq N'} \left( k(\alpha + \beta) + \beta f_2(N' - g_1(k)) \right)
\]

Let’s focus on the second term of the optimization in (12) since the first term is already in the single block format, and corresponds to coding only for the second user. Using the induction hypotheses for \( N' < N \),

\[
\max_{0 \leq k \leq N} \left[ (k' + k)(\alpha + \beta) + \beta f_2(N - g_1(k) - g_1(k')) \right]
\]

We notice that (14) is optimized when \( k' = 0 \). In order to see this, assume \( k^*, k^* \) are the optimal values, then compare the objective function evaluated at \( k^*, k^* \) and \( 0, k^* + k^* \). The objective function when \( k' = 0 \) is greater than the assumed optimal value at \( k^*, k^* \), using super additivity of \( f(.) \) and subadditivity of \( g(.) \). Hence also the second term in (12) results in two single block coding.

With similar methods one can easily show that threshold approximation gives very simple code structure also for fixed-\( k \) and fixed-\( n \) cases [10]. For example, in the fixed-\( k \) approach, the threshold approximation suggests that the optimal coding is achieved using block sizes of two different size either \( g_1(k) \) or \( g_2(k) \). Once a packet of size \( g_2(k) \) is coded, no following packets of size \( g_1(k) \) are produced.

### IV. Numerical Results

In this section, we consider BEC, further results with BSC can be found in [10]. Since the optimal probability of error \( q_{k,n} \) is not known in general, we will consider three sets of results, depending whether the achievability (upper bound), normal approximation or converse (lower bound) expression for the finite block length probability of error are used (see equations (184), (290), (187) of [6]). The corresponding curves shall be denoted with “Ach.”, “Norm.” and “Con.”, respectively. Note that ideal performance i.e. assuming capacity achieving code for every block length with zero error probability yields \( NC \) successfully decoded information bits in any single-user transmission of \( N \) channel uses. For multicast problems, ideal assumption results in simple linear programs as given in Table I.

Fig. 3 compares the exact dynamic programming with the results obtained by the threshold approximation for the single-user fixed-\( k \) case for BEC with \( C = 0.7 \). We observe that, there is no significant difference between the optimal dynamic programming solution (varying packet size \( n \), UEP) and the solution of the threshold approximation (uniform packet size equal to \( n = g(k) \), EEP). In Fig. 3, \( \Delta_k(N)/k \) which corresponds average number of correctly decoded packets is plotted for increasing \( N \). We fix \( k/N \) ratio to 1/20 and varied \( N \) between 500 – 20000. Notice that for \( N = 4000 \) (hence \( k = 200 \)) Ach., Norm. and Con. curves become very tight. Due to computational complexity of calculating Ach. and Con. curves for large \( N \), we plot only Norm. curves for \( N > 4000 \). The gap between the “ideal” performance (Shannon limit) and Norm. curve is due to the finite-\( k \) constraint. For example, for \( N = 20000 \) and \( k = 1000 \), (1) computes that \( q_{k,n} < 10^{-6} \) can be achieved for rate less than 0.65 obviously less than \( C = 0.7 \) (similar to rate vs. block length reports in Fig. 10 and Fig. 11 of [6]). Hence the finite length curve reaches \((N/k)0.65 = 13\) in Fig. 3. In Fig. 3, to avoid very small \( k \) values with short \( N \leq 4000 \) (as dictated by computational complexity of Ach. and Con. curves), we keep a large ratio of \( k/N \) in order to show that Ach., Con., and Norm. curves get tighter. But this ratio is very high with respect to sample experiments reported in previous UEP literature using digital images [1]. Next we consider another example with \( N = 256000 \) and \( k = 256 \) for the same channel where the \( N \) and \( k \) are selected considering the experiments in [1]. In this case we obtain \( \Delta_k(N)/k = 606.29 \) while \( \Gamma_k(N)/k = 604.51 \).

Dynamic programming of fixed-\( n \) and fixed-\( k \) approaches are not too complicated for the single user case. On the other hand, the exact dynamic programming UEP optimization for multicast has very high complexity (see [1], [2], and [10] for detailed complexity analysis) and it is not amenable for an on-line implementation. For this reason, we propose a low complexity suboptimal algorithm for multicast fixed-\( k \) problem inspired by solution structure of the threshold approximation. As discussed in Sec. III, for an \( L \)-user multicast the step analysis suggests that channel packets should be grouped in at most \( L \) many equal length packet groups. The optimal single-user value for the \( l^{th} \) user, \( \Delta_k(Z) \) is calculated recursively starting with \( Z = 1 \) to \( Z = N \) and saved for every user as a vector of length \( N \) only once at the beginning of the algorithm. Then the low complexity algorithm first partitions \( N \) bits into \( L \) blocks where \( l^{th} \) block is of.
size $N_l$. The coding strategies of each partition is decided using single-user dynamic programming independent of each other. The rationale behind this is the fact that optimization of $\Delta_k(N_l)$ does not depend on the allocations for previous blocks in average rate case but not in average distortion. Notice that $\Delta^\ast_{k,l}(N_l)$ is readily given by the initial calculation as explained but finding the solution itself requires re-running the optimization (10) for every partition. Concatenation of these individual block solutions forms the transmission policy $\pi$. Next for each user $V^l_\pi$ is calculated using (4) to obtain the weighted average. The algorithm then chooses the policy with highest weighted average. Notice that finding single-user optimal solution for every partitioning could be a daunting task due to re-running the optimization (10). Hence, as a surrogate for $V^l_\pi$ we propose using $\sum_{m=1}^L \Delta^m_k(N,n)$ which is actually its lower bound due to degradedness of the user channels.

In Fig. 4, we compare the performance of the dynamic programming approach finding the optimal solutions (curve labeled with Find. Sol.) and dynamic programming approach using the heuristic surrogate (curve labeled with Surr.). Similar algorithms can be also described for multicast-$(n,k)$ and fixed-$n$ problems (see [10] for further results).

![Figure 4](image-url)

**Figure 4.** $\alpha = 0.4$, $\beta = 1$ and $k/N = 1/20$ and $\delta_1 = 0.3$ and $\delta_2 = 0.1$ and curves are obtained only using Norm. approximation. Distribution of packet lengths are also given in the inset. Almost equal packet lengths can be observed for any partitioning.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\max \alpha_1 \alpha + \beta(a_1 + a_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s.t.$</td>
<td>$\frac{a_1}{a_2} + \frac{a_2}{a_3} \leq N$</td>
</tr>
<tr>
<td>$n$</td>
<td>$\max \alpha_1 C_1 + \beta(a_1 C_2 + a_2 C_3)$</td>
</tr>
<tr>
<td>$s.t.$</td>
<td>$a_1 + a_2 \leq N/n$</td>
</tr>
<tr>
<td></td>
<td>$\max \alpha_1 + \beta(k_1 + k_2)$</td>
</tr>
<tr>
<td>$s.t.$</td>
<td>$\frac{k_1}{k_2} + \frac{k_2}{k_3} \leq N$</td>
</tr>
</tbody>
</table>

**Table I**

Multicast solutions assuming capacity achieving codes. From left to right: fixed-$k$, fixed-$n$, variable-$(n,k)$

V. THE SINGLE-USER VARIABLE-$(n,k)$ PROBLEM

Define the function $S(N) \triangleq \max_{1 \leq k \leq N} k \bar{q}_{k,N}$, providing the largest average number of successfully decoded symbols, when we can design the coding rate $k/N$. In this section, we show that if $S(N)$ is super-additive, i.e. $S(a + b) \geq S(a) + S(b)$, then coding over a single block is optimal in the single-user variable-$(n,k)$ problem, with respect to the average rate criterion. This amounts to show that $\Delta^\ast(N)$ in (10) is equal to $S(N)$.

The super additivity of $S(N)$ can be observed numerically when $q_{k,n}$ is obtained using the finite block length results of [6]. By definition of optimum, we have that $\Delta^\ast(N) \geq S(N) = k_N \bar{q}_{k,N}$, where $k_N$ is the arg-max of the maximization w.r.t. $k$ that defines $S(N)$. Hence, assuming super-additivity, in the following we show that $\Delta^\ast(N) \leq S(N)$ using induction on $N$. For $N = 1$ we have trivially $k_1 = 1$ and $\Delta^\ast(1) = S(1)$. Next, assume that $\Delta^\ast(N') = S(N')$ for all $N' < N$. We rewrite (10) skipping the case of $N < 1$:

$$\Delta^\ast(N) = \max_{0 \leq n \leq N} \max_{0 \leq k \leq n} \bar{q}_{k,n} [k + k(N-n)\bar{q}_{(k,N)-n},N-n]$$

$$\leq \max_{0 \leq n \leq N} \max_{0 \leq k \leq n} \bar{q}_{k,n} k + \max_{0 \leq k \leq n} \bar{q}_{k(N-n),N-n}$$

$$\leq \max_{0 \leq n \leq N} \left( S(n) + k(N-n)\bar{q}_{(k,N)-n},N-n \right)$$

$$= \max_{0 \leq n \leq N} \left( S(n) + S(N-n) \right) \leq S(N). \quad (15)$$

**REFERENCES**


---

$^2$Letting $x = k/N$ and using the normal approximation (1), an approximation of $S(N)$ is given by $\tilde{S}(N) = \sup_{0 \leq x \leq N} N x Q \left( \sqrt{N} \frac{\bar{x}}{\sqrt{N}} \right)$. If one can show that $N x Q \left( \sqrt{N} \frac{\bar{x}}{\sqrt{N}} \right)$ is convex in $N$ for all $x$, then $\tilde{S}(N)$ would be convex in $N$. It is well known that convexity and $S(0) = 0$ implies super-additivity. Fortunately, the Hessian w.r.t. $N$ is not always non-negative for $x \in [0, 1]$. 

---

$\Delta^\ast(N)$ is super-additive, i.e. $S(a + b) \geq S(a) + S(b)$.