Probability in the Engineering and Informational Sciences, **28**, 2014, 473–488.
doi:10.1017/S0269964814000102

DOMAIN EXTENSIONS OF THE ERLANG LOSS FUNCTION: THEIR SCALABILITY AND ITS APPLICATIONS TO COOPERATIVE GAMES

Frank Karsten, Marco Slikker and Geert-Jan van Houtum
School of Industrial Engineering
Eindhoven University of Technology
PO Box 513, 5600 MB
Eindhoven
The Netherlands
E-mails: f.j.p.karsten@tue.nl m.slikker@tue.nl g.j.v.houtum@tue.nl

We prove that several extensions of the classic Erlang loss function to non-integral numbers of servers are scalable: the blocking probability as described by the extension decreases when the offered load and the number of servers s are increased with the same relative amount, even when scaling up from integral s to non-integral s. We use this to prove that when several Erlang loss systems pool their resources for efficiency, various corresponding cooperative games have a non-empty core.

1. INTRODUCTION

The Erlang loss function is a true classic in queueing theory. Derived by A. K. Erlang in 1917, it represents the steady-state blocking probability in an $M/G/s/s$ queue. It is often convenient to have an *extension* of the classic Erlang loss function, that is, a function that extends the domain of the Erlang loss function to non-integral values of s. For instance, techniques such as the equivalent random method and Hayward's approximation (Fredericks [5]) for performance approximations of loss systems with overflow layers require such an extension. Extensions also come in handy for the analysis of cooperative resource pooling games; cf. Karsten, Slikker and Van Houtum [11].

Various extensions are available in the literature, such as the *linear interpolation* (used in Kortanek, Slikker and Van Houtum [12]) and the integral representation in terms of the Gamma function (see, e.g., Jagerman [7]). The present paper will introduce a new extension with interesting properties that are helpful in analyzing the linear interpolation. Of course, we can concoct an infinite number of extensions. What is important is whether or not an extension satisfies appealing *properties*.

One property that we will be particularly interested in deals with the behavior of an extension when the arrival rate λ and the number of servers s are scaled up with the same relative amount. Smith and Whitt [14] showed that the classic Erlang loss probability always decreases when we scale up in this fashion (and stick to integral numbers for s). For instance, the blocking probability in a system with $\lambda = 2$ and $s = 1$ is higher than in a system with $\lambda = 4$ and $s = 2$ or a system with $\lambda = 6$ and $s = 3$. But if we would scale up to a system with, say, $\lambda = 5$ and $s = 2.5$, then we would have to interpret its blocking probability via an extension. A natural question is whether or not an extension retains the economies-of-scale behavior exhibited by the classic Erlang loss function when scaling up to any non-integral number of servers. To capture this concept formally, we introduce a property that we call *scalablility*.

The question of whether or not an extension is scalable is particularly interesting for the linear interpolation L, as this interpolation represents a long-run blocking probability that is actually achievable by "mixing" between two consecutive integer numbers of servers. That is, operating under each of those two numbers of servers during a desired percentage of time. This "mixing" (cf. Van Houtum and Zijm [16]) sometimes occurs in practice to meet a service constraint exactly. The first major contribution of this paper is that we prove scalability of L. So, the blocking probability in a system with $\lambda = 2$ and $s = 1$ is higher than the blocking probability in a system with $\lambda = 5$ that uses two servers half of the time and three servers the other half, assuming steady-state conditions are continually achieved under both.

Scalability of L does not come automatically. Indeed, the above-mentioned "mixing" yields a combination of the Erlang loss probability under $\lambda = 5$ and $s = 2$ and the Erlang loss probability under $\lambda = 5$ and $s = 3$, whereas Smith and Whitt [14] only told us something about the scaled system with $\lambda = 4$ and $s = 2$ and the scaled system with $\lambda = 6$ and $s = 3$. In fact, we will show that the linear interpolation of the closely related Erlang *delay* function (for the $M/M/s$ queue) is *not* scalable, in contrast to L.

The economies-of-scale behavior of the blocking probability encourages service providers to cooperate by pooling their $M/G/s/s$ systems. One may think of hospital departments that pool intensive care beds, chemical factories that share a specialized fire extinguishing device, or car rental agencies that pool luxurious rental cars. For such service providers, each associated with an exogenous Poisson stream of customer arrivals, holding a group of common servers rather than dedicated resources for each individual customer stream is beneficial to the system as a whole.

Yet, to achieve a sustainable collaboration between all service providers (the *players*), each individual player has to be guaranteed a lower cost than what he would face by acting independently or by pooling with only a few of the other players. To fairly allocate the joint costs of the common servers amongst the players, we apply concepts from cooperative game theory and look for a *stable* allocation. Under a stable allocation, no subset of players has an incentive to split off and form a separate pooling group. The set of all such allocations is called the core.

Several authors have recently analyzed cost allocation problems for shared queueing systems with infinite waiting room from the perspective of cooperative game theory. For example, Karsten, Slikker and Van Houtum [10] study a model wherein several $M/M/s$ queues join forces, and Anily and Haviv [1] focus on the $M/M/1$ model. Cost allocation problems for pooled queueing systems where waiting is *not* possible have been studied solely in the context of the $M/G/s/s$ system. For $M/G/s/s$ games where any coalition picks a number of common servers to minimize the sum of *linear* resource costs for servers and penalty costs for blocked customers, Özen, Reiman and Wang [13] and Karsten, Slikker and Van Houtum [9] have independently proven that the allocation of the collective costs proportional to players' arrival rates is stable. Karsten, Slikker and Van Houtum [11] extended this to non-identical resource cost parameters across players and additionally tackled $M/G/s/s$ game variants with exogenously given numbers of servers.

In the present paper, our second major contribution (next to the scalability of L) is that we prove stability of proportional allocations for a generalization of Özen et al.'s $M/G/s/s$ game and for a variant thereof: the generalization features concave rather than linear resource costs, and the variant has a service constraint in lieu of actual payments of penalty costs. To prove these stability results, the scalability of L is instrumental — this proof approach differs from the one used by Özen, Reiman and Wang [13].

2. THE ERLANG LOSS FUNCTION AND ITS EXTENSIONS

In this section, we define the classic Erlang loss function and subsequently introduce three functions that extend its domain to non-integral numbers of servers.

2.1. The Classic Erlang Loss Function

To describe the classic Erlang loss function, let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ denote the set of all non-negative integers and let $\mathbb{R}_{++} = \mathbb{R}_+ \setminus \{0\}$ denote the set of all (strictly) positive real numbers. Then, the *Erlang loss function* $B : \mathbb{N}_0 \times \mathbb{R}_{++} \to [0, 1]$ is defined by

$$B(s,a) = \frac{a^s/s!}{\sum_{y=0}^{s} a^y/y!} \qquad \text{for any} s \in \mathbb{N}_0 \quad \text{and} \quad a \in \mathbb{R}_{++}. \tag{1}$$

$B(s, a)$ may be interpreted as the blocking probability, that is, the steady-state probability that an arriving customer finds no free server, in an *Erlang loss system* ($M/G/s/s$ queue) with s servers and offered load a. In such a system, the service times are independent and identically distributed according to some general distribution function with finite mean $\tau > 0$, customers arrive according to a Poisson process with rate $a/\tau > 0$, and each newly arriving customer immediately goes into service if there is an unoccupied server available. All other customers are lost.

The Erlang loss function satisfies several useful properties. The properties that we will use in this paper's analysis are collected in the following theorem.

THEOREM 2.1: *Let $s, s' \in \mathbb{N}$ and $a, \tau, \lambda, \lambda' \in \mathbb{R}_{++}$. Then:*
(i) $B(s, \lambda/\mu)$ is decreasing and convex in μ for μ on $\mathbb{R}_{++}$.
(ii) $\dfrac{\partial B}{\partial a}(s, a) = [B(s, a) - 1 + \dfrac{s}{a}] \cdot B(s, a)$.
(iii) $B(s, a) = \dfrac{aB(s-1, a)}{aB(s-1, a) + s}$.
(iv) $a[B(s, a)]^2 - 1 - (a - s - 1)B(s, a) \leq 0$.
(v) $B(ts, ta)$ is decreasing in t for $t \in \{1/s, 2/s, \ldots\}$.

Property (i) corresponds to Proposition 3 in Harel [6]. The partial derivative, (ii), is due to Theorem 15 in Jagerman [7]. The recursive relation of property (iii) is well-known (see, e.g., Jagerman [7], p. 531). Property (iv) corresponds to Inequality (19) in Özen, Reiman and Wang [13]. They show that this inequality is valid by contradiction. We provide an alternative, independently derived, direct proof in the Appendix.

The final property, (v) captures the economies of scale in Erlang loss systems: as shown in the appendix of Smith and Whitt [14], when we increase the offered load a and the number of servers s with the same relative amount t, the blocking probability always decreases. This scaling occurs, for instance, when we combine the servers and arrival streams of two Erlang

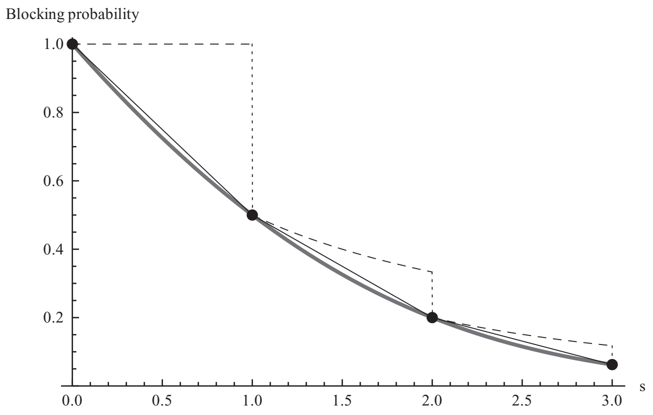

FIGURE 1. The extensions $X(s,a)$ (dashed), $L(s,a)$ (middle), and $\hat{B}(s,a)$ (thick) for $a = 1$ fixed.

loss systems with the same a/s value into a single joint system. Note that this is only meaningful when the scaling factor t is chosen such that the number of servers in the scaled system, ts, is an integer number. In Section 3, we will introduce a scaling concept that retains meaning for non-integral numbers of servers.

2.2. Extensions of the Erlang Loss Function

In this paper, we are interested in functions that extend the domain of the Erlang loss function to non-integral numbers of servers. Formally, we call any function $E : \mathbb{R}_+ \times \mathbb{R}_{++} \to [0,1]$ an *extension of the Erlang loss function* (or *extension* for short) if $E(s,a) = B(s,a)$ for all $s \in \mathbb{N}_0$ and $a \in \mathbb{R}_{++}$. We next introduce three different extensions; all are depicted in Figure 1.

First, the continuous function $\hat{B} : \mathbb{R}_+ \times \mathbb{R}_{++} \to [0,1]$ is defined by

$$\hat{B}(s,a) = \left(a \int_0^\infty e^{-ax}(1+x)^s dx \right)^{-1} \quad \text{for all } s \in \mathbb{R}_+ \text{ and } a \in \mathbb{R}_{++}. \tag{2}$$

This function, which is related to the Erlang loss function via the Gamma function, is indeed an extension (Jagerman [7]). It is one of the most commonly used extensions in the literature (Fredericks [5]).

Another extension is obtained by using linear interpolation (cf. Kortanek [12]). To be more precise, this piecewise linear function $L : \mathbb{R}_+ \times \mathbb{R}_{++} \to [0,1]$ is defined by

$$L(s,a) = (1 - (s - \lfloor s \rfloor)) \cdot B(\lfloor s \rfloor, a) + (s - \lfloor s \rfloor) \cdot B(\lceil s \rceil, a) \quad \text{for all } s \in \mathbb{R}_+ \text{ and } a \in \mathbb{R}_{++}, \tag{3}$$

where $\lceil s \rceil$ denotes the smallest integer larger than or equal to s, and $\lfloor s \rfloor$ denotes the largest integer smaller than or equal to s. By virtue of being a linear interpolation, this function is obviously an extension of the Erlang loss function.

Finally, we introduce a new extension of the Erlang loss function $X : \mathbb{R}_+ \times \mathbb{R}_{++} \to [0,1]$, which is defined by

$$X(s,a) = \begin{cases} B(\lfloor s \rfloor, a \cdot \lfloor s \rfloor / s) & \text{if } s \geq 1 \text{ and } a \in \mathbb{R}_{++}; \\ 1 & \text{if } s \in [0,1) \text{ and } a \in \mathbb{R}_{++}. \end{cases} \tag{4}$$

This function is not continuous, but it is clearly an extension by definition. We use this extension to prove scalability of L in the next section.

3. SCALABILITY OF THE EXTENSIONS

In this section, we aim to show that all three extensions, and the linear interpolation in particular, satisfy the scalability property. This property will prove instrumental in analyzing cooperative resource pooling games later on. We start by defining it formally.

Definition 3.1: *An extension E is said to be* scalable *if, for each $s \in \mathbb{N}_0$ and $a \in \mathbb{R}_{++}$, $E(ts, ta) \le E(s, a)$ for all $t \in (1, \infty)$ and $E(ts, ta) \ge E(s, a)$ for all $t \in (0, 1)$.*

In words, scalability means that if we take an Erlang loss system with an *integer* number of servers as our starting point, an increase (decrease) of the offered load and the number of servers with the same relative amount will result in a non-strictly decreased (increased) blocking probability. If this scaling would result in a system with a *non-integral* number of servers, then the blocking probability is described by the extension.

Not every extension is scalable. Nevertheless, one might expect that any "natural" extension, such as a linear interpolation, is scalable. Surprisingly, however, linear interpolations of elementary performance measures for canonical multi-server queueing models need not satisfy (the equivalent of) scalability in general, as shown in the following example.

Example 3.1: *Besides B, Erlang [4] derived another classic performance measure: the Erlang delay function C, defined for any $a > 0$ and $s \in \mathbb{N}$ with $s > a$ by*

$$C(s,a) = \left(1 + \sum_{y=0}^{s-1} \frac{s!(1 - a/s)}{y! a^{s-y}}\right)^{-1}.$$

$C(s, a)$ may be interpreted as the steady-state probability that an arrival must wait before beginning service in an Erlang delay system ($M/M/s$ queue) with s servers and offered load a. As the analogue of Part (v) of Theorem 2.1, it holds that $C(ts, ta)$ is decreasing in t for $t \in \{1/s, 2/s, \dots\}$, for each $a > 0$ and $s \in \mathbb{N}$ with $s > a$ (Calabrese [2]). We now consider the linear interpolation C^{lin}, defined for any $a > 0$ and $s \in \mathbb{N}$ with $\lfloor s \rfloor > a$ by

$$C^{lin}(s,a) = (1 - (s - \lfloor s \rfloor)) \cdot C(\lfloor s \rfloor, a) + (s - \lfloor s \rfloor) \cdot C(\lceil s \rceil, a).$$

Let $s = 1$, $a = 0.5$, and $t = 1.2$. We obtain that $C^{lin}(s,a) = C(1, 0.5) = 0.5$, whereas $C^{lin}(ts, ta) = 0.8 \cdot C(1, 0.6) + 0.2 \cdot C(2, 0.6) = 0.8 \cdot \frac{3}{5} + 0.2 \cdot \frac{9}{65} = \frac{33}{65}$. So, if we start with an Erlang delay system with 1 server and offered load 0.5, then scaling up by a factor of 1.2 results in an increased *delay probability.*

The same type of scaling, however, leads to a decreased *blocking probability in an Erlang loss system: $L(s,a) = B(1, 0.5) = \frac{1}{3}$, whereas $L(ts, ta) = 0.8 \cdot B(1, 0.6) + 0.2 \cdot B(2, 0.6) = 0.8 \cdot \frac{3}{8} + 0.2 \cdot \frac{9}{89} = \frac{57}{178}$.*

Given the tight link (see, e.g., p. 92 of Cooper [3]) between the Erlang loss function and the Erlang delay function and the negative result that Example 3.1 established for the latter, there is the question of whether L is actually scalable.

A sufficient condition for scalability of an extension E is subhomogeneity of degree zero, that is, the property that $E(ts, ta)$ is decreasing in t on $\mathbb{R}_{++}$, for each $s \in \mathbb{R}_+$ and $a \in \mathbb{R}_{++}$.

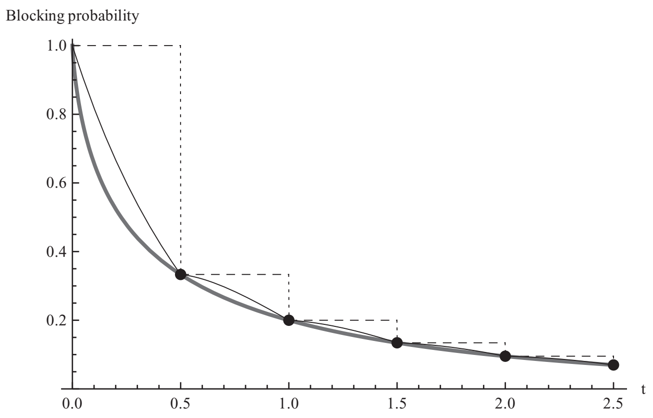

FIGURE 2. For $s = 2$ and $a = 1$ fixed, the extensions $X(ts, ta)$ (dashed), $L(ts, ta)$ (middle), and $\hat{B}(ts, ta)$ (thick) as functions of t.

(This goes beyond scalability by additionally comparing two systems that *both* have a real number of servers.) Scalability of the extensions $\hat{B}$ and X is readily shown.

PROPOSITION 3.1: *The extension $\hat{B}$ is scalable.*

PROOF: Follows from the appendix of Smith and Whitt [14]. ■

PROPOSITION 3.2: *The extension X is scalable.*

PROOF: Let $s \in \mathbb{N}_0$ and $a \in \mathbb{R}_{++}$. Then, $X(ts, ta)$ as a function of t is stepwise constant and non-increasing for t on $\mathbb{R}_{++}$. (See Figure 2 for an illustration.) More precisely, if $s > 0$, then for t between two successive values $t^- \in \{1/s, 2/s, \ldots\}$ and $t^+ = t^- + 1/s$, $X(ts, ta)$ equals $B(st^-, at^-)$. (And for any $t \in \mathbb{R}_{++}$ with $ts < 1$, we have $X(ts, ta) = 1$.) Combining this with Part (v) of Theorem 2.1 completes the proof. ■

To prove scalability of L, it will be convenient to show that the graph of L never dips below the graph of $\hat{B}$ and never jumps above the graph of X, as illustrated in Figures 1 and 2.

PROPOSITION 3.3: $\hat{B}(s, a) \leq L(s, a)$ *for all* $s \in \mathbb{R}_+$ *and* $a \in \mathbb{R}_{++}$.

PROOF: Let $a \in \mathbb{R}_{++}$. Since $\hat{B}(s, a)$ as a function of s is convex for s on $\mathbb{R}_+$ (Jagers and Van Doorn [8]), whereas $L(s, a)$ linearly interpolates between points on the graph of $B(s, a)$ at which $B(s, a) = \hat{B}(s, a)$, the desired inequality follows immediately from the definition of convexity. ■

In the process of proving that the graph of L always lies at or below the graph of X, we will present two lemmas that consider X and L as a function of the number of servers on a domain restricted to an interval between two consecutive integers. To describe this formally, it will be convenient to introduce two restricted functions: for any fixed $r = (S, a) \in \mathbb{N} \times \mathbb{R}_{++}$, we define the functions L_r and X_r, both mapping $[S, S + 1)$ to $[0, 1]$, by

$L_r(s) = L(s,a)$ and $X_r(s) = X(s,a)$ for all $s \in [S, S+1)$. So, these functions are described on their domain by

$$X_r(s) = B(S, aS/s); \tag{5}$$

$$L_r(s) = (1 + S - s) \cdot B(S,a) + (s - S) \cdot B(S+1, a). \tag{6}$$

Figure 1 on page 476 provides an illustration: there, the graph of $X_{(1,1)}$ corresponds to the dashed curve on $[1,2)$ and the graph of $L_{(1,1)}$ corresponds to the middle curve on $[1,2)$. The following lemma states two useful properties of X.

Lemma 3.4: *Let $r = (S,a) \in \mathbb{N} \times \mathbb{R}_{++}$. Then X_r is decreasing and convex on its domain $[S, S+1)$.*

Proof: By part (i) in Theorem 2.1, it holds for each fixed $\hat{s} \in \mathbb{N}$ and $\lambda \in \mathbb{R}_{++}$ that $B(\hat{s}, \lambda/\mu)$ is decreasing and convex in μ for $\mu \in [S, S+1)$. By substituting $\hat{s} = S$, $\lambda = aS$, and $\mu = s$, we conclude that $X_r(s) = B(S, aS/s)$ is decreasing and convex in s on $[S, S+1)$. ■

In contrast to X and L, the functions X_r and L_r are differentiable, which allows us to compare their derivatives, evaluated at S, in the following lemma.

Lemma 3.5: *Let $r = (S,a) \in \mathbb{N} \times \mathbb{R}_{++}$. Then $L'_r(S) \le X'_r(S)$.*

Proof: First of all, the derivative of L_r for any $s \in [S, S+1)$ is

$$L'_r(s) = B(S+1, a) - B(S,a). \tag{7}$$

To obtain the derivative of X_r, we combine part (ii) of Theorem 2.1 with Equation (5) to derive that for any $s \in [S, S+1)$:

$$\begin{aligned} X'_r(s) &= [B(S, aS/s) - 1 + S/(aS/s)] \cdot B(S, aS/s) \cdot \left(-aS/s^2\right) \\ &= [B(S, aS/s) - 1 + s/a] \cdot B(S, aS/s) \cdot \left(-aS/s^2\right). \end{aligned} \tag{8}$$

For notational ease, let $\mathfrak{B} = B(S,a)$. Evaluating the derivatives (7) and (8) at $s = S$, we obtain

$$\begin{aligned} L'_r(S) - X'_r(S) &= B(S+1,a) - \mathfrak{B} - \left[\mathfrak{B} - 1 + \frac{S}{a}\right] \cdot \mathfrak{B} \cdot \frac{-a}{S} \\ &= \frac{a\mathfrak{B}}{a\mathfrak{B} + S + 1} - \mathfrak{B} - \left[\mathfrak{B} - 1 + \frac{S}{a}\right] \cdot \mathfrak{B} \cdot \frac{-a}{S} \\ &= \mathfrak{B} \cdot \left[\frac{a}{a\mathfrak{B} + S + 1} + (\mathfrak{B} - 1) \cdot \frac{a}{S}\right] \\ &= \frac{a\mathfrak{B}}{S(a\mathfrak{B} + S + 1)} \cdot \left[S + (\mathfrak{B} - 1) \cdot (a\mathfrak{B} + S + 1)\right] \\ &= \frac{a\mathfrak{B}}{S(a\mathfrak{B} + S + 1)} \cdot \left[a\mathfrak{B}^2 - 1 - (a - S - 1)\mathfrak{B}\right] \\ &\le 0. \end{aligned}$$

The second equality holds by part (iii) of Theorem 2.1. The other equalities hold by rewriting. The inequality holds because $\mathfrak{B} > 0$, $a > 0$, $S(a\mathfrak{B} + S + 1) > 0$, and

$a\mathfrak{B}^2 - 1 - (a - S - 1)\mathfrak{B} \leq 0$, where the last-named inequality holds by Part (iv) of Theorem 2.1. We conclude that $L'_r(S) \leq X'_r(S)$. ■

We use these lemmas to prove that the graph of L never jumps above the graph of X.

PROPOSITION 3.6: *$X(s, a) \geq L(s, a)$ for all $s \in \mathbb{R}_+$ and $a \in \mathbb{R}_{++}$.*

PROOF: Let $s \in \mathbb{R}_+$ and $a \in \mathbb{R}_{++}$. We distinguish two cases.

Case 1: $s < 1$. Then, by definition $X(s, a) = 1$, whereas $L(s, a) \leq 1$.

Case 2: $s \geq 1$. Then, we denote $S = \lfloor s \rfloor$ and consider the functions $X_{(S,a)}$ and $L_{(S,a)}$, which are described on their domain $[S, S + 1)$ by Eqs. (5) and (6). First of all, observe that $X_{(S,a)}(S) = L_{(S,a)}(S)$ since both X and L are extensions of the Erlang loss function. Secondly, by Lemma 3.5, we observe that the derivative of $L_{(S,a)}$ at S does not exceed the derivative of $X_{(S,a)}$ at S. Third, $X_{(S,a)}$ is convex by Lemma 3.4, whereas $L_{(S,a)}$ is by definition a linear function, which (together with the second observation) implies that $X'_{(S,a)} \geq L'_{(S,a)}$ on $[S, S + 1)$. Combining these three observations yields $X_{(S,a)}(s) \geq L_{(S,a)}(s)$. We conclude that $X(s, a) \geq L(s, a)$. ■

With the various propositions derived so far, we can now prove the following theorem.

THEOREM 3.7: *The extension L is scalable.*

PROOF: Let $s \in \mathbb{N}_0$ and $a \in \mathbb{R}_{++}$. For any $t \in (1, \infty)$, we find that

$$L(ts, ta) \leq X(ts, ta) \leq X(s, a) = L(s, a),$$

where the first inequality holds by Proposition 3.6, the second inequality holds by Proposition 3.2, and the equality holds because both X and L are extensions. Analogously, for any $t \in (0, 1)$, we find that

$$L(ts, ta) \geq B(ts, ta) \geq B(s, a) = L(s, a),$$

where the first inequality holds by Proposition 3.3, the second inequality holds by Proposition 3.1, and the equality holds because both B and L are extensions. Hence, L is scalable. ■

We remark that it remains an open question whether or not L satisfies the stronger property of subhomogeneity of degree zero. We have not been able to find a counterexample, but at the same time our proof approach for Theorem 3.7 is not readily adaptable for subhomogeneity of degree zero. Nevertheless, scalability is sufficient for the purpose of the next section.

4. COOPERATIVE RESOURCE POOLING GAMES

For reasons of self-containedness, we first give a brief introduction to cooperative game theory. Subsequently, we introduce resource pooling situations and analyze two associated games.

4.1. Preliminaries on Cooperative Game Theory

A cooperative cost game with transferable utility, which we will simply refer to as *game*, is a pair (N, c) where N is a non-empty, finite set of *players* and $c : 2^N_- \to \mathbb{R}$ is the *characteristic cost function*. Here, $2^N_- = \{M \mid M \subseteq N, M \neq \emptyset\}$ represents the set of all (non-empty) *coalitions*. The value $c(M)$ is interpreted as the total costs of the joint cooperative effort if only the players in coalition M are involved in it. In particular, $c(N)$ represents the total costs for the *grand coalition* N when all players agree on working together.

In cooperative game theory, players are assumed to be able to draw up binding agreements, and side payments are allowed. A central problem is to allocate $c(N)$ to the individual players in a fair way. Formally, an *allocation* is a vector $x \in \mathbb{R}^N$ satisfying $\sum_{i \in N} x_i = c(N)$. The value x_i is interpreted as the costs assigned to player i. An allocation is called *stable* if $\sum_{i \in M} x_i \leq c(M)$ for all $M \in 2^N_-$. Under a stable allocation, no group of players has to pay more collectively than what they would face if they would split off and establish cooperation on their own. The set of all stable allocations is called the *core*.

An allocation scheme for a game (N, c) is a vector $y = (y_{i,M})_{i \in M, M \in 2^N_-}$, with $\sum_{i \in M} y_{i,M} = c(M)$ for all coalitions $M \in 2^N_-$, which specifies how to allocate the costs of every coalition to its members. Such a scheme is called a *population monotonic allocation scheme* (PMAS) if the amount that a player has to pay does not increase when the coalition to which he belongs grows. That is, $y_{i,M} \geq y_{i,R}$ for all coalitions $M, R \in 2^N_-$ with $M \subseteq R$ and $i \in M$. If the game admits a PMAS, say y, then $(y_{i,N})_{i \in N}$ is an element of its core (Sprumont [15]).

4.2. Resource Pooling Situations

Consider several players that require certain costly servers for their customer populations. Each player's customers arrive according to mutually independent Poisson processes. If a customer finds no free server upon arrival, he is lost and some penalty costs may have to be paid. We assume throughout that players are interested in their long-term average costs per time unit.

In a setting with more than one player, the players could benefit by pooling their servers and by jointly optimizing the number of servers in their shared service system (with the objective either of minimizing the sum of resource and penalty costs or of minimizing the number of servers subject to a constraint on the blocking probability). To analyze such settings, we define a *resource pooling situation* as a tuple of parameters $(N, \lambda, \tau, H, p, \beta)$, where

- N is the non-empty, finite set of players;
- $\lambda \in \mathbb{R}^N_{++}$ is the vector of arrival rates, that is, $\lambda_i > 0$ denotes the arrival rate of customers that belong to player $i \in N$;
- $\tau > 0$ is the mean service time for an arbitrary customer of any player;
- $H : \mathbb{N}_0 \to \mathbb{R}_+$ is a concave, increasing, unbounded function specifying that the resource costs for holding s servers are $H(s)$ per unit time;
- $p > 0$ is the expected penalty costs that are incurred whenever a customer is blocked;
- $\beta \in (0, 1)$ is a service level constraint indicating the maximal blocking probability.

This model covers a broad range of situations in which resource sharing can occur between separate Erlang loss systems. The concave nature of the resource costs represents additional economies of scale that may be exploited by acquiring and maintaining resources collaboratively. For our analysis, it will be convenient to extend the domain of the resource cost function H to all non-negative reals by a linear interpolation, that is, we define the function $H^{\text{lin}} : \mathbb{R}_+ \to \mathbb{R}_+$ by

$$H^{\text{lin}}(s) = (1 - (s - \lfloor s \rfloor)) \cdot H(\lfloor s \rfloor) + (s - \lfloor s \rfloor) \cdot H(\lceil s \rceil).$$

Given any resource pooling situation $\varphi = (N, \lambda, \tau, H, p, \beta)$, the players in a coalition $M \in 2^N_-$ are assumed to collaborate by complete pooling of their respective arrival streams and common servers into a joint system. Since the superposition of independent Poisson processes is also a Poisson process, this coalition now faces a Poisson arrival process with merged rate $\lambda_M = \sum_{i \in M} \lambda_i$. We assume that the common servers can handle all types of customers with equal ease and that all customers can effortlessly access the joint service facility.

Based on these assumptions, coalition M's joint service facility behaves as an Erlang loss system. For this type of resource pooling, we can construct two associated games, which are introduced in the next two subsections. The first is appropriate when penalty costs represent tangible monetary payments; the second is appropriate when they are intangible.

4.3. Games with Penalty Cost Payments

If the penalties for blocked customers represent tangible monetary costs, we can formulate a game corresponding to resource pooling situation $\varphi = (N, \lambda, \tau, H, p, \beta)$ in which any coalition picks a cost-minimizing number of servers. Here, bad performance is captured by p, and β becomes superfluous in the tuple. For any choice of the number of servers in the joint system $s \in \mathbb{N}_0$, the expected total relevant costs per unit time in steady state are given by

$$K_M(s) = H(s) + B(s, \lambda_M \tau) \cdot \lambda_M p. \tag{9}$$

Since the value of the Erlang loss function is confined to the interval $(0, 1]$, whereas the resource cost function H increases unboundedly, there exists an optimal number of servers, which can be found by, for example, an enumerative search procedure. There may still be multiple cost minimizing numbers of servers, so to avoid ambiguity we define a specific optimal number of servers by

$$S^*_M = \min\{S \in \mathbb{N}_0 : K_M(S) = \min_{s \in \mathbb{N}_0} K_M(s)\}. \tag{10}$$

We have suppressed the dependence of $K_M(s)$ and S^*_M on φ to avoid notational baggage.

We call the game (N, c^φ) with $c^\varphi(M) = \min_{s \in \mathbb{N}_0} K_M(s)$ for all coalitions $M \in 2^N_-$ the associated *penalty cost game*.

An easy way of allocating the total costs of the grand coalition — or any other coalition that may form — is by dividing these costs proportional to the arrival rate of each player. Formally, we define for any resource pooling situation $\varphi = (N, \lambda, \tau, H, p, \beta)$ this allocation scheme $\mathcal{A}(\varphi)$ by $\mathcal{A}_{i,M}(\varphi) = c^\varphi(M) \cdot \lambda_i / \lambda_M$ for any coalition $M \in 2^N_-$ and player $i \in M$. If costs are shared according to this scheme, then a player with more frequent customer arrivals pays a greater share of the costs, which seems reasonable. What's more, this scheme is always population monotonic, as stated by the following theorem.

THEOREM 4.1: *Let $\varphi = (N, \lambda, \tau, H, p, \beta)$ be a resource pooling situation. Then $\mathcal{A}(\varphi)$ is a PMAS of the associated penalty cost game (N, c^φ).*

PROOF: For any coalition $M \in 2^N_-$, we extend the domain of the cost function K_M, introduced in Eq. (9), by a linear interpolation, that is, we define the function $K_M^{\text{lin}} : \mathbb{R}_+ \to \mathbb{R}_+$ by

$$K_M^{\text{lin}}(s) = (1 - (s - \lfloor s \rfloor)) \cdot K_M(\lfloor s \rfloor) + (s - \lfloor s \rfloor) \cdot K_M(\lceil s \rceil) = H^{\text{lin}}(s) + L(s, \lambda_M \tau) \cdot \lambda_M p.$$

We now fix two arbitrary coalitions $M, R \in 2^N_-$ with $M \subseteq R$, and we let $i \in M$. Then,

$$\begin{aligned}
\mathcal{A}_{i,R}(\varphi) &= \frac{\lambda_i}{\lambda_R} \cdot K_R(S_R^*) \\
&\le \frac{\lambda_i}{\lambda_R} \cdot K_R^{\text{lin}}\left(\frac{\lambda_R}{\lambda_M} S_M^*\right) \\
&= \frac{\lambda_i}{\lambda_R} \cdot H^{\text{lin}}\left(\frac{\lambda_R}{\lambda_M} S_M^*\right) + L\left(\frac{\lambda_R}{\lambda_M} S_M^*, \lambda_R \tau\right) \cdot \lambda_i p \\
&\le \frac{\lambda_i}{\lambda_R} \cdot H^{\text{lin}}\left(\frac{\lambda_R}{\lambda_M} S_M^*\right) + L(S_M^*, \lambda_M \tau) \cdot \lambda_i p \\
&= \frac{\lambda_i}{\lambda_M} \cdot \frac{\lambda_M}{\lambda_R} \cdot H^{\text{lin}}\left(\frac{\lambda_R}{\lambda_M} S_M^*\right) + B(S_M^*, \lambda_M \tau) \cdot \lambda_i p \\
&\le \frac{\lambda_i}{\lambda_M} \cdot H(S_M^*) + B(S_M^*, \lambda_M \tau) \cdot \lambda_i p \\
&= \frac{\lambda_i}{\lambda_M} \cdot K_M(S_M^*) = \mathcal{A}_{i,M}(\varphi).
\end{aligned}$$

The first inequality holds because S_R^* is a cost minimizing number of servers for coalition R, so both $K_R(\lfloor \frac{\lambda_R}{\lambda_M} S_M^* \rfloor)$ and $K_R(\lceil \frac{\lambda_R}{\lambda_M} S_M^* \rceil)$ are no smaller than $K_R(S_R^*)$, and the same holds for the associated linear interpolation. The second inequality holds because L is scalable (Theorem 3.7) and $S_M^* \in \mathbb{N}_0$. The third inequality holds because H is a concave non-negative function, so the same holds for its linear interpolation H^{lin}, and thus

$$\frac{\lambda_M}{\lambda_R} \cdot H^{\text{lin}}\left(\frac{\lambda_R}{\lambda_M} S_M^*\right) \le \frac{\lambda_M}{\lambda_R} \cdot H^{\text{lin}}\left(\frac{\lambda_R}{\lambda_M} S_M^*\right) + \frac{\lambda_R - \lambda_M}{\lambda_R} \cdot H^{\text{lin}}(0) \le H^{\text{lin}}\left(S_M^*\right),$$

where the first inequality holds because $H^{\text{lin}}(0) \ge 0$ and the second inequality holds by concavity (since $S_M^* = \frac{\lambda_M}{\lambda_R} \cdot \frac{\lambda_R}{\lambda_M} \cdot S_M^* + \frac{\lambda_R - \lambda_M}{\lambda_R} \cdot 0$). We conclude that $\mathcal{A}(\varphi)$ is a PMAS. ■

We remark that the linear interpolation L could not have been replaced by the continuous extension $\hat{B}$ in this proof because then, as shown in the following example, an inequality corresponding to the proof's first inequality would not hold anymore.

EXAMPLE 4.1: *Consider the two-player resource pooling situation* $(N, \lambda, \tau, H, p, \beta)$ *with* $N = \{1, 2\}$, $\lambda_1 = 0.6$, $\lambda_2 = 0.4$, $\tau = 1$, $H(s) = 0.85s$ *for all* $s \in \mathbb{N}_0$, *and* $p = 3$. *For coalition* $M = \{1\}$, *the optimal number of servers is* $S_M^* = 1$. *For coalition* $R = \{1, 2\}$, *the optimal number of servers is* $S_R^* = 2$ *with associated costs* $K_R(S_R^*) = 0.85 S_R^* + \hat{B}(S_R^*, \lambda_R \tau) \cdot 3\lambda_R = 2.3$. *However, for* $s = S_M^* \lambda_R / \lambda_M = 1\frac{2}{3}$, *it holds that* $0.85s + \hat{B}(s, \lambda_R \tau) \cdot 3\lambda_R < 2.26 < 2.3 = K_R(S_R^*)$. *So, the costs for coalition R under the optimal (integer) number of servers are actually* larger *than the costs for this coalition under* $1\frac{2}{3}$ *servers when the associated blocking probability is interpreted via the continuous extension* $\hat{B}$. *This is graphically represented in Figure 3.*

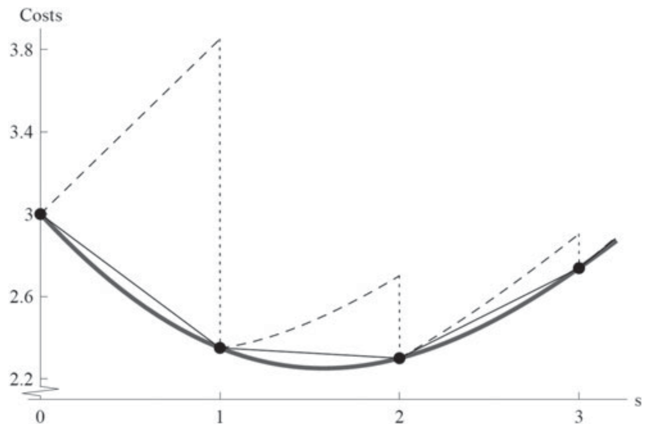

FIGURE 3. For each of the three extensions of the Erlang loss function and for the Erlang loss function itself, the corresponding costs $hs + B(s, \lambda_R\tau) \cdot \lambda_R p$ (thick curve), $hs + L(s, \lambda_R\tau) \cdot \lambda_R p$ (straight line segments), $hs + X(s, \lambda_R\tau) \cdot \lambda_R p$ (dashed, discontinuous), and $K_R(s)$ (big dots) as functions of s for the coalition R as described in Example 4.1.

Finally, by Sprumont [15], the following corollary follows immediately from Theorem 4.1.

COROLLARY 4.2: *Let $\varphi = (N, \lambda, \tau, H, p, \beta)$ be a resource pooling situation. Then $(\mathcal{A}_{i,N}(\varphi))_{i \in N}$ is in the core of the associated penalty cost game (N, c^φ).*

4.4. Games with a Service Constraint

If penalty costs are hard to quantify, a service constraint model may be preferable: we can let each player incur the consequences for its own blocked customers and look for an allocation of *only* the resource costs that ensures that no subcoalition can improve (achieve the service constraint at lower resource cost) by splitting off. This gives rise to an alternative game corresponding to any resource pooling situation $\varphi = (N, \lambda, \tau, H, p, \beta)$ where any subcoalition will pick the (possibly "mixed", cf. Van Houtum and Zijm [16]) number of servers that yields a blocking probability of exactly β. That is, any coalition $M \in 2^N_-$ will pick the unique $s \in \mathbb{R}_{++}$ such that $L(s, \lambda_M) = \beta$, and we will denote this number of servers by σ_M. This σ_M is well defined because $L(s, \lambda_M)$ as a function of s on $\mathbb{R}_+$ is strictly decreasing and continuous, with image or range $(0, 1]$. We call the game (N, d^φ), with $d^\varphi(M) = H^{\text{lin}}(\sigma_M)$ for all $M \in 2^N_-$ the associated *service constraint game.*

We propose for any resource pooling situation $\varphi = (N, \lambda, \tau, H, p, \beta)$ a proportional allocation $\mathcal{P}(\varphi)$, defined by $\mathcal{P}_i(\varphi) = H^{\text{lin}}(\sigma_N) \cdot \lambda_i/\lambda_N$ for each player $i \in N$. The following theorem states that this allocation is always stable provided that there is an integer in the interval $[\sigma_N \lambda_M/\lambda_N, \sigma_N]$ for all subcoalitions M. This is a technical requirement that simultaneously illustrates the limitation and allure of scalability. This requirement is met if σ_N is a sufficiently large real number and each player is big enough. It is also met if σ_N is any integer number.

THEOREM 4.3: *Let $\varphi = (N, \lambda, \tau, H, p, \beta)$ be a resource pooling situation. If $\lfloor \sigma_N \rfloor \geq \sigma_N \lambda_M/\lambda_N$ for all $M \in 2^N_- \setminus \{N\}$, then $\mathcal{P}(\varphi)$ is in the core of the associated service constraint game (N, d^φ).*

PROOF: Let $M \in 2^N_- \setminus \{N\}$. Since $\beta \in (0,1)$, we must have $\sigma_N > 0$. Assume that $\lfloor \sigma_N \rfloor \geq \sigma_N \lambda_M / \lambda_N$. As $\sigma_N > 0$ and $\lambda_M/\lambda_N > 0$, this assumption implies that $\lfloor \sigma_N \rfloor > 0$. Then,

$$L(\sigma_M, \lambda_M) = \beta = L(\sigma_N, \lambda_N) \leq L\left(\lfloor \sigma_N \rfloor, \lambda_N \frac{\lfloor \sigma_N \rfloor}{\sigma_N}\right) \leq L\left(\sigma_N \frac{\lambda_M}{\lambda_N}, \lambda_M\right). \tag{11}$$

The equalities hold by definition of σ_M and σ_N, respectively. The inequalities hold because L is scalable (Theorem 3.7), which we can employ because $\lfloor \sigma_N \rfloor \in \mathbb{N}$ and because $\sigma_N \geq \lfloor \sigma_N \rfloor \geq \sigma_N \lambda_M / \lambda_N$.

Since $L(s, \lambda_M)$ is decreasing in s on $\mathbb{R}_+$, Inequality (11) implies that $\sigma_N \lambda_M / \lambda_N \leq \sigma_M$. This yields

$$\sum_{i \in M} \mathcal{P}_i(\varphi) = H^{\text{lin}}(\sigma_N) \cdot \lambda_M / \lambda_N \leq H^{\text{lin}}(\sigma_N \cdot \lambda_M / \lambda_N) \leq H^{\text{lin}}(\sigma_M) = d^{\varphi}(M).$$

In the first inequality, we use that H^{lin} is concave and non-negative. In the second inequality, we use that H^{lin} is increasing. We conclude that $\mathcal{P}(\varphi)$ is stable. ■

Acknowledgments

Parts of results of this paper appeared in the preliminary working paper version, Karsten, Slikker and Van Houtum [9].

References

1. Anily, S. & Haviv, M. (2010). Cooperation in service systems. *Operations Research* 58(3): 660–673.
2. Calabrese, J. (1992). Optimal workload allocation in open networks of multi-server queues. *Management Science* 38(12): 1792–1802.
3. Cooper, R. (1981). *Introduction to queueing theory.* New York: North-Holland.
4. Erlang, A. (1917). Løsning af nogle problemer fra sandsynlighedsregningen af betydning for de automatiske telefoncentraler. *Electroteknikeren* 13: 5–13. Translation: Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. In E. Brockmeyer, H.L. Halstrøm, and A. Jensen, editors, *The Life and Works of A.K. Erlang*, pages 138–155. Transactions of the Danish Academy of Technical Sciences, 1948.
5. Fredericks, A. (1980). Congestion in blocking systems — a simple approximation technique. *Bell System Technical Journal* 59(6): 805–827.
6. Harel, A. (1990). Convexity properties of the Erlang loss formula. *Operations Research* 38(3): 499–505.
7. Jagerman, D. (1974). Some properties of the Erlang loss function. *Bell System Technical Journal* 53(3): 525–551.
8. Jagers, A. & Van Doorn, E. (1986). On the continued Erlang loss function. *Operations Research Letters* 5(1): 43–46.
9. Karsten, F., Slikker, M. & Van Houtum, G. (2011). Analysis of resource pooling games via a new extension of the Erlang loss function. BETA Working Paper 344, Eindhoven University of Technology.
10. Karsten, F., Slikker, M. & Van Houtum, G. (2011). Resource pooling and cost allocation among independent service providers. BETA Working Paper 352, Eindhoven University of Technology.
11. Karsten, F., Slikker, M. & Van Houtum, G. (2012). Inventory pooling games for expensive, low-demand spare parts. *Naval Research Logistics* 59(5): 311–395.
12. Kortanek, K., Lee, D. & Polak, G. (1981). A linear programming model for design of communications networks with time varying probabilistic demands. *Naval Research Logistics Quarterly* 28(1): 1–32.
13. Özen, U., Reiman, M. & Wang, Q. (2011). On the core of cooperative queueing games. *Operations Research Letters* 39(5): 385–389.
14. Smith, D. & Whitt, W. (1981). Resource sharing for efficiency in traffic systems. *Bell System Technical Journal* 60(1): 39–55.
15. Sprumont, Y. (1990). Population monotonic allocation schemes for cooperative games with transferable utility. *Games and Economic Behavior* 2(4): 378–394.

16. Van Houtum, G. & Zijm, W. (2000). On the relationship between cost and service models for general inventory systems. *Statistica Neerlandica* 54(2): 127–147.

APPENDIX

In this appendix, we prove Part (iv) of Theorem 2.1: $a[B(s,a)]^2 - 1 - (a-s-1)B(s,a) \leq 0$ for all $s \in \mathbb{N}$ and $a \in \mathbb{R}_{++}$.

PROOF: (Proof of Part (iv) of Theorem 2.1.) Let $S \in N$ and $a \in \mathbb{R}_{++}$. By Eq. (1), what we aim to show is that

$$a\left[\frac{a^S/S!}{\sum_{y=0}^{S} a^y/y!}\right]^2 - 1 - (a-S-1)\frac{a^S/S!}{\sum_{y=0}^{S} a^y/y!} \leq 0. \tag{A.1}$$

Re-arranging, combining all terms into a single fraction, and multiplying both sides of the resulting inequality with $-\left(\sum_{y=0}^{S} a^y/y!\right)^2 < 0$, we obtain that (A.1) is equivalent to

$$\left(\sum_{y=0}^{S} \frac{a^y}{y!}\right)^2 - a \cdot \left(\frac{a^S}{S!}\right)^2 + (a-S-1) \cdot \left(\sum_{y=0}^{S} \frac{a^y}{y!}\right) \cdot \frac{a^S}{S!} \geq 0. \tag{A.2}$$

For all $s \in \mathbb{N}$, define

$$f(s) = \left(\sum_{y=0}^{s} \frac{a^y}{y!}\right)^2 - \frac{a^{2s+1}}{s! \cdot s!} + \sum_{y=0}^{s} \frac{a^{y+s} \cdot (a-s-1)}{y! \cdot s!}. \tag{A.3}$$

Note that (A.2) corresponds to $f(S) \geq 0$. To complete the proof, we show that $f(s) \geq 0$ for all $s \in \mathbb{N}$ by induction. First,

$$f(1) = (1+a)^2 - a^3 + (a+a^2)(a-2) = (1+a)^2 - a^3 + a^2 - 2a + a^3 - 2a^2 = 1 \geq 0.$$

To avoid empty summations later on, it is convenient to treat the case $s = 2$ separately:

$$\begin{aligned} f(2) &= \left(\frac{1}{2}a^2 + a + 1\right)^2 - \frac{1}{4}a^5 + \left(\frac{1}{4}a^4 + \frac{1}{2}a^3 + \frac{1}{2}a^2\right) \cdot (a-3) \\ &= \frac{1}{4}a^4 + a^3 + 2a^2 + 2a + 1 - \frac{1}{4}a^5 + \frac{1}{4}a^5 + \frac{1}{2}a^4 + \frac{1}{2}a^3 - \frac{3}{4}a^4 - \frac{3}{2}a^3 - \frac{3}{2}a^2 \\ &= \frac{1}{2}a^2 + 2a + 1 \geq 0. \end{aligned}$$

For the induction step, let $s \in \{3, 4, \ldots\}$ and assume that $f(s-1) \geq 0$.

Then,

$$
\begin{aligned}
f(s) &= \left(\sum_{y=0}^{s}\frac{a^y}{y!}\right)^2 - \frac{a^{2s+1}}{s!\cdot s!} + \sum_{y=0}^{s}\frac{a^{y+s}\cdot(a-s-1)}{y!\cdot s!}\\
&= \left(\sum_{y=0}^{s}\frac{a^y}{y!}\right)^2 - \frac{a^{2s+1}}{s!\cdot s!} + \frac{a^{2s+1}}{s!\cdot s!} - \frac{a^{2s}\cdot s}{s!\cdot s!} - \frac{a^{2s}}{s!\cdot s!} + \sum_{y=0}^{s-1}\frac{a^{y+s}\cdot(a-s-1)}{y!\cdot s!}\\
&= \frac{a^{2s}}{s!\cdot s!} + 2\cdot\sum_{y=0}^{s-1}\frac{a^{y+s}}{y!\cdot s!} + \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 - \frac{a^{2s}\cdot s}{s!\cdot s!} - \frac{a^{2s}}{s!\cdot s!} + \sum_{y=0}^{s-1}\frac{a^{y+s}\cdot(a-s-1)}{y!\cdot s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 - \frac{a^{2s}\cdot s}{s!\cdot s!} + \sum_{y=0}^{s-1}\frac{a^{y+s}\cdot(a-s+1)}{y!\cdot s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[-\frac{a^s\cdot s}{s!} + \sum_{y=1}^{s}\frac{a^y}{(y-1)!} + \sum_{y=0}^{s-1}\frac{a^y\cdot(-s+1)}{y!}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[\sum_{y=1}^{s-1}\frac{a^y}{(y-1)!} + \sum_{y=0}^{s-1}\frac{a^y\cdot(-s+1)}{y!}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[\sum_{y=1}^{s-2}\frac{a^y}{(y-1)!} + \sum_{y=0}^{s-2}\frac{a^y\cdot(-s+1)}{y!}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[\left(\sum_{y=1}^{s-2}\frac{a^y}{y!}\cdot(y-s+1)\right) - s + 1\right]\cdot\frac{a^s}{s!}\\
&\geq \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[\sum_{y=1}^{s-2}\frac{a^y}{y!}\cdot\left(\frac{-s^2}{y+1}+s\right) - s + 1\right]\cdot\frac{a^s}{s!}\\
&\geq \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[\sum_{y=1}^{s-2}\frac{a^y}{y!}\cdot\left(\frac{-s^2}{y+1}+s\right) - s^2 + s - \frac{s^2}{a}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[\sum_{y=0}^{s-2}\frac{a^y}{y!}\cdot\left(\frac{-s^2}{y+1}+s\right) - \frac{s^2}{a}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[-\frac{a^{s-1}\cdot s}{(s-1)!} - \sum_{y=0}^{s-2}\frac{a^y\cdot s^2}{(y+1)!} - \frac{s^2}{a} + \sum_{y=0}^{s-2}\frac{a^{y-1}\cdot a\cdot s}{y!} + \frac{a^{s-1}\cdot s}{(s-1)!}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[-\frac{a^{s-1}\cdot s}{(s-1)!} - \sum_{y=1}^{s-1}\frac{a^{y-1}\cdot s^2}{y!} - \frac{s^2}{a} + \sum_{y=0}^{s-1}\frac{a^{y-1}\cdot a\cdot s}{y!}\right]\cdot\frac{a^s}{s!}\\
&= \left(\sum_{y=0}^{s-1}\frac{a^y}{y!}\right)^2 + \left[-\frac{a^{s-1}\cdot s}{(s-1)!} + \sum_{y=0}^{s-1}\frac{a^{y-1}\cdot(a-s)\cdot s}{y!}\right]\cdot\frac{a^s}{s!}
\end{aligned}
$$

$$= \left(\sum_{y=0}^{s-1} \frac{a^y}{y!}\right)^2 - \frac{a^{2(s-1)+1}}{(s-1)! \cdot (s-1)!} + \sum_{y=0}^{s-1} \frac{a^{y+s-1} \cdot (a-(s-1)-1)}{y!(s-1)!}$$

$$= f(s-1) \geq 0.$$

In the first couple of steps, we split up summations, split off terms from summations, and cancel out common terms. The first inequality is valid because, for all $y \in \mathbb{N}_0$, it holds that $s^2 - 2s(y+1) + (y+1)^2 = (s-(y+1))^2 \geq 0$, and thus $y - s + 1 \geq -s^2/(y+1) + s$. The second inequality holds because $-s+1 \geq -s^2 + s \geq -s^2 + s - s^2/a$, since $s > 1$. In the final steps, we rearrange our expression to show that it is equal to $f(s-1)$, which was non-negative by the induction hypothesis. Hence, by the principle of mathematical induction we have for all $s \in \mathbb{N}$ that $f(s) \geq 0$. This completes the proof. ■