Acta Informatica 34, 411–428 (1997)

Acta Informatica

Performance analysis of parallel programs based on directed acyclic graphs

Georg Trogemann[1], Matthias Gente[2]

[1] Academy of Media Arts, Department of Computer Science/Audio-Visual Media, Peter-Welter-Platz 2, D–50676 Cologne, Germany
[2] University of Erlangen Nürnberg, Department of Computer Science IV, Martensstrasse 1, D–91058 Erlangen, Germany

Received: 26 August 1994 / 13 May 1996

Abstract. Task precedence graphs are widely used for modeling and evaluation of parallel applications. Their nodes represent the subtasks of the parallel program and the edges represent the precedence relations between the subtasks. The execution times of the subtasks are described by random variables and their distributions. In our paper we introduce a new class of distributions, particularly suited for the modeling and evaluation of parallel programs.

Exponential polynomials introduced by Sahner and Trivedi have the disadvantage that a large number of parameters is needed for the representation of realistic task execution times, which usually have a small value of variation. We extend this class to derive the class of truncated θ-exponential polynomials which allow the representation of realistic task execution times with fewer parameters. Additionally this class of distributions has the advantage that minimum as well as maximum execution times can be guaranteed.

Models with a large number of subtasks n can not be evaluated on a computer using exact analytical methods because of memory requirements and numerical inaccuracies, which accumulate, when the operations of analysis are applied. Using extreme value theory we derive approximate formulas for the parallel independent execution of n subtasks, a structure, which can be found in every parallel program. The obtained results for truncated and not truncated distributions show, that distributions with an infinite domain are not suitable, particularly for massively parallel structures.

1. Introduction

Performance modeling which can be used to predict the execution time of a parallel program is an important tool in the development phase of a parallel application. A well established approach is the representation of the functional structure of the program by precedence graphs ([11], [10], [5], [16], [4]). The nodes of these graphs represent

Correspondence to: G. Trogemann, e-mail: trogemann@khm.de

the subtasks of a parallel program, and the edges represent the precedence relations between them. Precedence graphs are very popular because of their comprehensibility.

The GIANT-Model[1] [1] which will be introduced in Sect. 2 is also based on precedence graphs. Random variables are assigned to the nodes of the graph to represent the temporal behavior of the subtasks of the parallel application. Additional parameters can be used to represent dynamic scheduling mechanisms and communication strategies.

Empirical distributions can be provided for the execution times of the tasks if the model is to be evaluated numerically [10]. Only distributions which can be represented by a class of functions, which is closed under the operations of analysis are appropriate for the evaluation on a computer, if the distribution function for the total completion time of the graph is to be calculated symbolically in the time parameter t. Martin [11] uses distributions which are represented piecewise by polynomials in the analysis of series-parallel structures. Exponential polynomials which are used by Sahner and Trivedi [17] to model task execution times and also to represent the reliability of redundant systems, are also closed under the analytic operations for series-parallel graphs. We will demonstrate in Sect. 3 that exponential polynomials have the disadvantage that many parameters are required to represent distributions with small variation and large mean value; these often occur in the modeling of the execution times of parallel programs. The number of parameters also increases greatly if the operations of analysis are used.

θ-exponential polynomials as introduced in Sect. 3.1 are derived from common exponential polynomials by translation and allow the representation of such distributions with significantly fewer parameters. In Sect. 3.1 it is proven that the class of θ-exponential polynomials has the properties which are required, if the overall distribution function is to be calculated symbolically in the time parameter t. These properties include closure under the operations multiplication, convolution, integration, differentiation and the ability to approximate arbitrary distributions to any degree of accuracy.

In Sect. 3.2 we introduce the class of truncated θ-exponential polynomials. These are created by truncating the right-hand end of θ-exponential polynomials, which makes them suitable for modeling the bounded execution time of subtasks of parallel programs. The useful closure property of the θ-exponential polynomials remains intact. We show in Sect. 4 that the right-hand end-point of the distributions of the subtasks has a decisive effect on the total execution time, when modeling massively parallel systems. In reality the run-times of the subtasks are bounded and thus we need truncated distributions to represent the run-time behavior correctly.

The accuracy with which the operations of analysis are performed is impacted by rounding errors, due to the finite accuracy of the representation of numbers in a digital computer. These accumulate as the analysis progresses and can lead to completely false results for task structures containing a large number of nodes. It is therefore practically impossible to determine the exact distribution of the total execution time of large task structures. Approximation formulas are derived in Sect. 4 using extreme value theory for the basic structure of parallel applications, the parallel execution of independent tasks. These formulas are based on the limit distributions of θ-exponential polynomials and truncated θ-exponential polynomials. The total execution time can thus be obtained exactly for small graphs and approximately for large graphs using the approximation method mentioned above. The validity of the approximation method

[1] **G**raphorientated **I**nteractive **AN**alysis **T**ool

is confirmed by comparing the results of extreme value analysis with results obtained by simulation.

2. The *GIANT*-Model

In the modeling and performance evaluation tool *GIANT* parallel programs are represented by precedence graphs. The nodes of the graphs represent the subtasks and the edges represent the precedence relations of the program. These graph models - referred to below as task structures - are easy to understand and so are widely used for modeling parallel programs ([12], [10], [15]). Figure 1 demonstrates that it is possible to represent system constraints such as a limited number of processors by simply adding nodes and arcs. Graph (a) represents the pure algorithm with inherent parallelism n. The execution of the program on a parallel system with p processors leads to structure (b). The communication between processor P_1 and the other processors $P_i, i = 2 \ldots p$ is indicated by the nodes K_{Ii} and K_{Oi} (c).

If stochastically independent random variables X_T, $T \in \{T_I, T_O, T_1 \ldots T_n, K_{I2} \ldots K_{Ip}, K_{O2} \ldots K_{Op}\}$ are assigned to the subtasks T of the graphs the following expression describes the total execution time $L(G)$ of the graph G[2]:

$$L(G) = \max_{\pi \in path(G)} \left(\sum_{T \in \pi} X_T \right) \tag{1}$$

For example the total execution time of (a) can be calculated as

$$\max_{\pi \in path(G)} \left(\sum_{T \in \pi} X_T \right) = \max_{i \in \{1 \ldots n\}} \{ X_{T_I} + X_{T_i} + X_{T_O} \} \tag{2}$$

If G is series-parallel, (1) can be transformed to an expression in which each random variable occurs only once. All graphs in Fig. 1 are series-parallel. For example expression (2) can be transformed to:

$$L(G) = X_{T_I} + \max_{i \in \{1 \ldots n\}} X_{T_i} + X_{T_O} \tag{3}$$

The resulting expression can be evaluated in linear or polynomial time [17] by recursively applying the operations convolution and multiplication on the mutually independent subexpressions.

3. Modeling the execution times of parallel programs

Continuous distribution functions are presented in this section, which are suitable for the description of the execution times of parallel programs if the total completion time of the graph is to be computed in a semi-symbolic form. The following should be taken into account when selecting such a class of distributions:

1. The evaluation is only possible if the class is closed with respect to the operations of analysis. In particular such a class of distributions must be closed with respect to multiplication, differentiation, integration and convolution.

[2] $path(G)$ is the set of all paths in the graph.

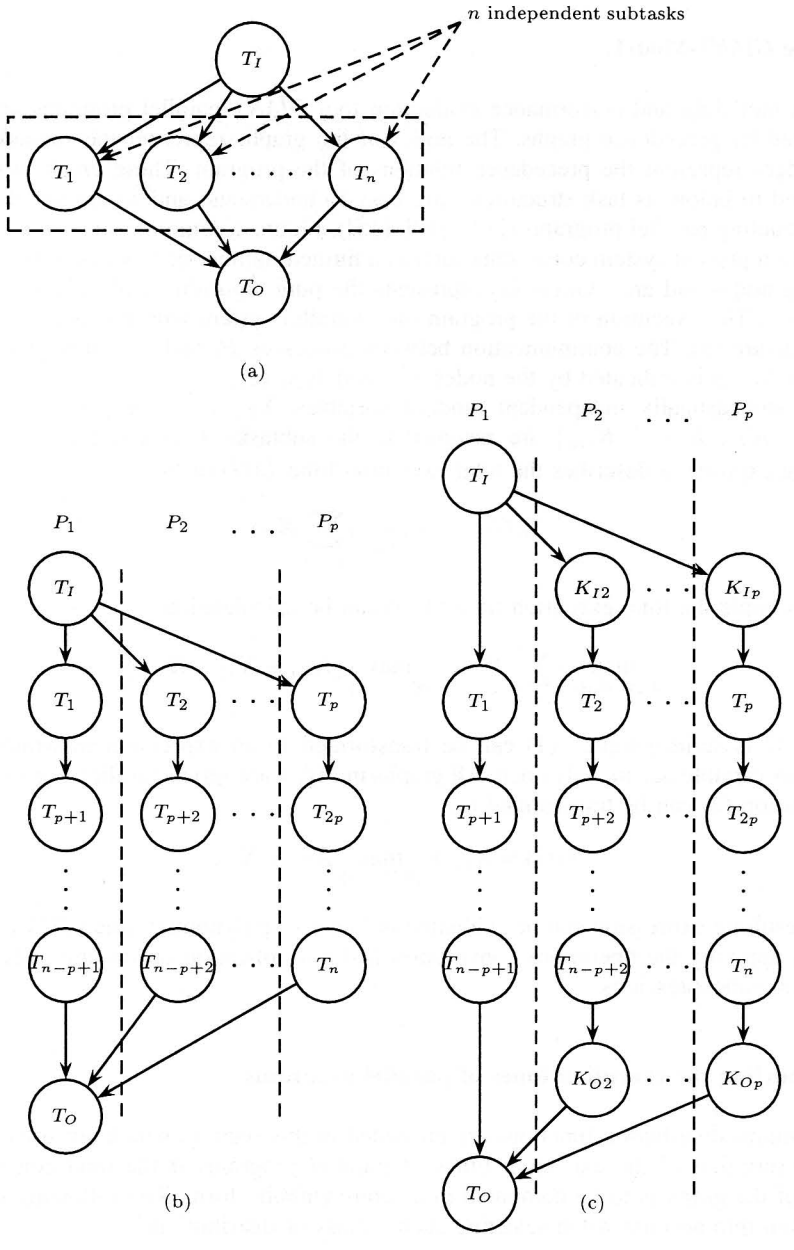

Fig. 1. Transformation of a graph to represent system constraints

2. The class must be capable of representing empirical distributions to any required degree of accuracy.
3. The execution of the particular operations of analysis must not require excessive computer time or memory.

Exponential polynomials $EP(t) = \sum_{i=1}^{n} a_i t^{k_i} e^{-\lambda_i t}$ with $a_i \in \mathbb{C}, \lambda_i \in \mathbb{C}, k_i \in \mathbb{N}$, introduced in [16] constitute a parameterized class of functions which satisfy requirements 1 and 2 above. Sahner and Trivedi [16],[17] use exponential polynomials to model the execution times of parallel programs and also the reliability of particular components in parallel computers. When using exponential polynomials to model execution times, the analysis involves a manipulation of the parameters (a_i, k_i, λ_i) of the exponential polynomials. [15] presents the results of series-parallel operations as a function of the parameter t. The number of parameter triples (a_i, k_i, λ_i) obtained as the result of these operations increases strongly with the number of stages $k_i + 1$. To satisfy the third requirement above it is therefore necessary to keep the number of stages as small as possible. However, execution times obtained in practice, often having coefficients of variation $c^2 = Var[X_i]/E[X_i]^2 \ll 1$, can only be represented by a large number of stages. For example, an Erlang distribution with mean 10 and variance 1 contains 100 stages. The distribution function $F(t)$ is $F(t) = 1 - e^{-\lambda t} \sum_{i=0}^{99} \frac{\lambda^i t^i}{i!}$. The multiplication of two such distributions with different rates λ produces 400 parameter triples; this is unacceptable as an operand in further operations on account of the memory requirement, processing time, and the numerical problems. Such distributions can be represented using significantly smaller values for k_i if the translation of individual terms along the t-axis is allowed. If the Erlang distribution with mean 10 and variance 1 above is translated by 8 units, then only 4 stages are required to represent it.

3.1. θ-Exponential polynomials

In this section we give a formal definition of the class of θ-exponential polynomials and prove that they have the same useful properties as exponential polynomials.

A function $H : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ of the form

$$H(t) = \sum_{i=1}^{n} H_i(t) \quad \text{with} \quad H_i(t) = \begin{cases} a_i(t-\theta_i)^{k_i} e^{-\lambda_i(t-\theta_i)} & \text{for} \quad t \geq \theta_i \\ 0 & \text{for} \quad t < \theta_i \end{cases}$$

with $a_i \in \mathbb{R}, \lambda_i \in \mathbb{R}_0^+, k_i \in \mathbb{N}_0$ is called a θ-exponential polynomial. The parameter θ_i is called the deterministic part, a_i the coefficient, λ_i the rate, and k_i the stage.

Figure 2 shows the phase diagram of θ-exponential polynomials. The parameters $c_i := a_i k_i! / \lambda_i^{k_i+1}$ can be interpreted as branching probabilities, if $c_i \in [0, 1]$. Every branch consists of a deterministic part followed by an Erlang distribution.

If $\forall_i : \theta_i = 0$ we obtain pure exponential polynomials. The translation of the individual terms ensures that typical program execution times can be represented using a small number of stages. In addition, the deterministic parts can be used to guarantee minimal execution times (i.e. startup phase of a subtask). In the following we will show that θ-exponential polynomials are suitable for the manipulation on a computer, because they are closed under the operations of analysis and they can approximate empirical distributions to any required degree of accuracy.

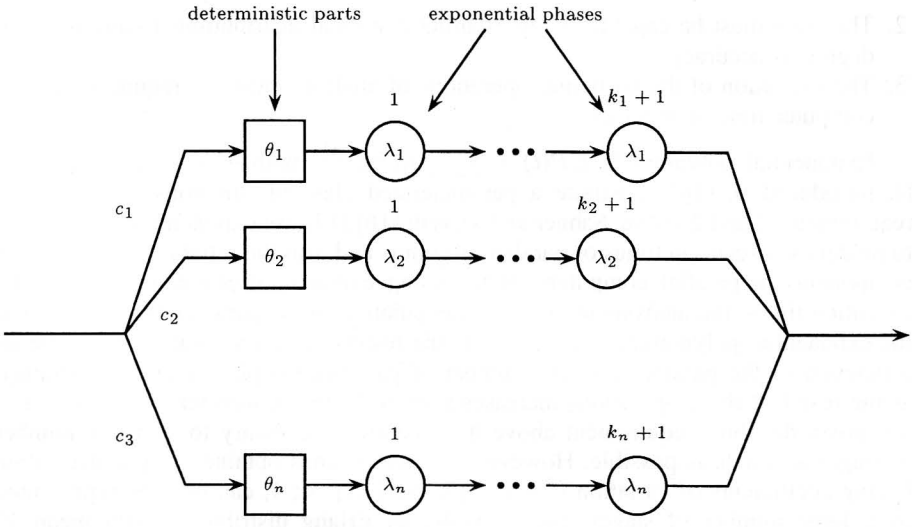

Fig. 2. Representation of θ-exponential polynomials by phase diagram

It is described in [6] how Erlang distributions can be used to approximate an arbitrary density function to any degree of accuracy. Any function $f(t)$ can be represented to any degree of accuracy by a step function. Each step of the step function corresponds to an Erlang-∞ function, which can be approximated to any degree of accuracy by an Erlang-n function with sufficiently large n. Thus any function $f(t)$ can be represented with any degree of accuracy by a weighted sum of Erlang density functions. Since θ-exponential polynomials are an extension of the class of mixed Erlang distributions, they can also approximate any distribution. In particular, the restriction of the parameters λ_i to the domain $\mathbb{R}_0^+$ and a_i to the range $\mathbb{R}$ instead of the domain $\mathbb{C}$ does not cause any loss of generality.

The function $I_i(t)$, which is used below, is defined as follows:

$$I_i(t) = \begin{cases} 0 & \text{if} \quad t \leq 0 \\ 1 & \text{if} \quad t > 0 \end{cases}$$

$E^l(f(t))$ is the l'th moment of a function $f(t)$, i. e. $E^l(f(t)) = \int_0^\infty t^l f(t)dt$. Let $H(t)$ be a density function from the class of θ-exponential polynomials.

Theorem 1. *The moments of the density term $H_i(t)$ are*

$$E^l(H_i(t)) = \frac{a_i k_i!}{\lambda_i^{k_i+1}} \sum_{j=0}^{l} \binom{l}{j} \frac{\prod_{m=1}^{j}(k_i+m)}{\lambda_i^j} \theta_i^{l-j}. \tag{4}$$

Proof.

$$\begin{aligned} E^l(H_i(t)) &= (-1)^l \frac{\partial^l}{\partial s^l}(L_s(H_i(t)))|_{s=0} \\ &= (-1)^l \frac{\partial^l}{\partial s^l}(a_i \frac{k_i!}{(s+\lambda_i)^{k_i+1}} \mathrm{e}^{-s\theta_i}) \mid_{s=0} \end{aligned}$$

$$= (-1)^l a_i k_i! \sum_{j=0}^{l} \binom{l}{j} \frac{\partial^j}{\partial s^j} \left(\frac{1}{(s+\lambda_i)^{k_i+1}}\right) \frac{\partial^{l-j}}{\partial s^{l-j}} (e^{-s\theta_i})|_{s=0}$$

$$= a_i \frac{k_i!}{\lambda_i^{k_i+1}} \sum_{j=0}^{l} \binom{l}{j} \frac{\prod_{m=1}^{j}(k_i+m)}{\lambda_i^j} \theta_i^{l-j}$$

Then the moments of θ-exponential polynomials are given by the summation formula:

$$E^l(H(t)) = \sum_{i=1}^{n} a_i \frac{k_i!}{\lambda_i^{k_i+1}} \sum_{j=0}^{l} \binom{l}{j} \frac{\prod_{m=1}^{j}(k_i+m)}{\lambda_i^j} \theta_i^{l-j}$$

□

The mean value of the θ-exponential polynomial is calculated as follows:

$$E(H(t)) = \sum_{i=1}^{n} a_i \frac{k_i!}{\lambda_i^{k_i+1}} \left(\frac{k_i+1}{\lambda_i} + \theta_i\right) = \sum_{i=1}^{n} (E(H_i^*) + a_i \frac{k_i!}{\lambda_i^{k_i+1}} \theta_i). \tag{5}$$

where $H_i^*(t) = H_i(t+\theta_i)$, i. e. H_i^* are terms without deterministic part. The variance of the individual terms is not altered by the translation along the t-axis; however, different θ_i alter the total variance.

It is shown below that the closure properties of the exponential polynomials are not affected by the translation of the exponential terms.

The operations necessary for the evaluation of series-parallel graphs, i. e. integration, differentiation, convolution, and multiplication, are presented below in symbolic form. The rules used to perform the calculations required for these operations can be obtained directly from the theorems proved below.

Theorem 2. *The class of θ-exponential polynomials is closed under integration. The definite integral of $H(t) = \sum_{i=1}^{n} H_i(t)$ with $\lambda_i > 0$ is given by:*

$$\int_0^t H(t)dt = \sum_{i=1}^{n} \hat{H}_i(t)$$

$$\hat{H}_i(t) = \begin{cases} \sum_{l=0}^{k_i} \frac{a_i k_i!}{\lambda_i^{k_i+1-l} l!} (t-\theta_i)^l e^{-\lambda_i(t-\theta_i)} + a_i \frac{k_i!}{\lambda_i^{k_i+1}} & \text{if } t > \theta_i \\ 0 & \text{if } t \le \theta_i \end{cases}$$

Proof. The expression $\hat{H}_i(t)$ must be derived after applying the summation rule. If the integral of the Erlang density is assumed to be known, then:

$$\int_0^t I_i(t) H_i(t) dt = I_i(t) a_i \frac{k_i!}{\lambda_i^{k_i+1}} \left(1 - e^{-\lambda_i(t-\theta_i)} \sum_{l=0}^{k_i} \frac{\lambda_i^l (t-\theta_i)^l}{l!}\right).$$

□

θ-exponential terms $a_i(t-\theta_i)^{k_i} e^{-\lambda_i(t-\theta_i)}$ with $k_i \in \{0, 1\}$ are discontinuous in θ_i. Before we can prove the closure property of the class under differentiation, we must explain the concept of the derivative for such terms.

Let $H_i(t)$ be a θ-exponential term with $k_i \in \{0, 1\}$. Then the derivative $H'(t)$ is defined as follows:

$$H_i'(t) = \begin{cases} 0 & \text{in} \quad (0, \theta_i) \\ a_i(e^{-\lambda_i(t-\theta_i)} - \lambda_i(t-\theta_i)e^{-\lambda_i(t-\theta_i)}) & \text{in} \quad [\theta_i, \infty) \quad \text{if} \quad k_i = 1 \\ -a_i\lambda_i e^{-\lambda_i(t-\theta_i)} & \text{in} \quad [\theta_i, \infty) \quad \text{if} \quad k_i = 0 \end{cases}$$

Theorem 3. *The class of θ-exponential polynomials is closed under differentiation. The first derivative of a θ-exponential polynomial $H(t) = \sum_{i=1}^{n} a_i(t-\theta_i)^{k_i} e^{-\lambda_i(t-\theta_i)}$ with $\forall_{i \in <1,n>} \lambda_i \neq 0$ is given by:*

$$\frac{dH(t)}{dt} = \sum_{i=1}^{n} \frac{d}{dt} H_i(t)$$

$$\frac{d}{dt} H_i(t) = \begin{cases} I_i(t)h(t) & \text{if} \quad k_i \neq 0 \\ -I_i(t)a_i\lambda_i e^{-\lambda_i(t-\theta_i)} & \text{if} \quad k_i = 0 \end{cases}$$

with

$$h(t) = a_i k_i (t-\theta_i)^{k_i - 1} e^{-\lambda_i(t-\theta_i)} - \lambda_i a_i (t-\theta_i)^{k_i} e^{-\lambda_i(t-\theta_i)}.$$

Proof. The theorem follows directly from the summation rule and the product rule for differentiation. □

Theorem 4. *The class of θ-exponential polynomials is closed under convolution. The convolution of two θ-exponential polynomials $h(t)$ and $H(t)$ is given by:*

$$h(t) \otimes H(t) = \left(\sum_{i=1}^{n} I_i(t)a_i(t-\theta_i)^{k_i} e^{-\lambda_i(t-\theta_i)}\right) \otimes \left(\sum_{j=1}^{m} I_j(t)A_j(t-\Theta_j)^{K_j} e^{-\Lambda_j(t-\Theta_j)}\right)$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{m} H_{ij}(t)$$

where

$$H_{ij}(t) = \begin{cases} 0 & \text{if} \quad t \leq \Theta_j + \theta_i \\ I_{ij}(t)\hat{H}_{ij}(t) & \text{if} \quad \lambda_i \neq \Lambda_j \\ I_{ij}(t)\frac{A_j a_i k_i! K_j!}{(k_i + K_j + 1)!}(t - \Theta_{ij})^{k_i + K_j + 1} e^{-\lambda_i(t-\Theta_{ij})} & \text{if} \quad \lambda_i = \Lambda_j \end{cases}$$

with $\Theta_{ij} = \Theta_j + \theta_i$ *and* $I_{ij}(t) = \begin{cases} 0 & \text{if} \quad t < \Theta_{ij} \\ 1 & \text{otherwise} \end{cases}$

$$\hat{H}_{ij}(t) = \sum_{l=0}^{k_i} \frac{(-1)^l \binom{K_j + l}{l} a_i k_i! A_j K_j!}{(k_i - l)!(\Lambda_j - \lambda_i)^{K_j + l + 1}} (t - \Theta_{ij})^{k_i - l} e^{-\lambda_i(t-\Theta_{ij})}$$
$$+ \sum_{l=0}^{K_j} \frac{(-1)^l \binom{k_i + l}{l} A_j K_j! a_i k_i!}{(K_j - l)!(\lambda_i - \Lambda_j)^{k_i + l + 1}} (t - \Theta_{ij})^{K_j - l} e^{-\Lambda_j(t-\Theta_{ij})}$$

Proof. Because of the linearity of the integration, it is sufficient to show the validity of the formula for $H_{ij}(t)$. A θ-exponential term can be regarded as a convolution of the unit impulse function u_0 at the position θ_i with the function $H_i(t + \theta_i) = H_i^*(t)$.

$$H_i(t) = u_0(t - \theta_i) \otimes H_i^*(t)$$

It follows that:

$$H_{ij}(t) = h_i(t) \otimes H_j(t) = u_0(t - \Theta_j) \otimes H_j^*(t) \otimes u_0(t - \theta_i) \otimes h_i^*(t).$$

Because of the commutativity of the convolution:

$$\begin{aligned} H_{ij}(t) &= u_0(t - \Theta_j) \otimes u_0(t - \theta_i) \otimes H_j^*(t) \otimes h_i^*(t) \\ &= u_0(t - (\Theta_j + \theta_i)) \otimes H_{ij}^*(t) \end{aligned}$$

We use the Laplace transformation to calculate the expression $H_{ij}^*(t)$.

$$\begin{aligned} L\{H_{ij}^*(t)\} &= L\{H_j^*(t)\} L\{h_i^*(t)\} \\ &= \frac{A_j K_j!}{(s + \Lambda_j)^{K_j+1}} \frac{a_i k_i!}{(s + \lambda_i)^{k_i+1}} \end{aligned}$$

Case 1: $\lambda_i \neq \Lambda_j$

$$L\{H_{ij}^*(t)\} = \frac{A_j a_i K_j! k_i!}{(s + \Lambda_j)^{K_j+1}(s + \lambda_j)^{k_i+1}}$$

We expand $L\{H_{ij}^*(t)\}$ as a partial fraction:

$$\begin{aligned} L\{H_{ij}^*(t)\} &= \sum_{l=0}^{K_j} \left(\frac{A_j a_i K_j! k_i!}{l!(s + \Lambda_j)^{K_j+1-l}} \right) \left(\frac{\partial^l}{\partial s^l} \frac{1}{(s + \lambda_i)^{k_i+1}} \right) |_{s=-\Lambda_j} \\ &+ \sum_{l=0}^{k_i} \left(\frac{A_j a_i K_j! k_i!}{l!(s + \lambda_i)^{k_i+1-l}} \right) \left(\frac{\partial^l}{\partial s^l} \frac{1}{(s + \Lambda_j)^{K_j+1}} \right) |_{s=-\lambda_i}. \end{aligned}$$

$H_{ij}^*(t)$ is obtained by differentiation and inverse transformation.

Case 2: $\lambda_i = \Lambda_j$

$$L\{H_{ij}^*(t)\} = A_j a_i K_j! k_i! \frac{1}{(s + \Lambda_j)^{K_j+k_i+2}}$$

We obtain by inverse transformation:

$$H_{ij}^*(t) = \frac{A_j a_i K_j! k_i!}{(K_j + k_i + 1)!} t^{k_i+K_j+1} e^{-\Lambda_j t}.$$

The translation of the functions $H_{ij}^*(t)$ by the amount $\Theta_{ij} = \Theta_j + \theta_i$, i. e. convolution with the unit impulse $u_0(t - (\Theta_j + \theta_i))$, produces the required result. □

Theorem 5. *The class of θ-exponential polynomials is closed under multiplication. The product of two θ-exponential polynomials is given by:*

$$\begin{aligned} H(t)h(t) &= \left(\sum_{j=1}^{m} I_j(t) A_j (t - \Theta_j)^{K_j} e^{-\Lambda_j(t-\Theta_j)}\right)\left(\sum_{i=1}^{n} I_i(t) a_i (t - \theta_i)^{k_i} e^{-\lambda_i(t-\theta_i)}\right) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{n} H_{ij}^*(t) \end{aligned}$$

where

$$H^*_{ij}(t) = \begin{cases} 0 & \text{if } t < \max(\theta_i, \Theta_j) \\ h^{\theta}_{ij}(t) & \text{if } \theta_i > \Theta_j \\ h^{\Theta}_{ij}(t) & \text{if } \Theta_j > \theta_i \\ h_{ij}(t) & \text{if } \Theta_j = \theta_i \end{cases}$$

$$h^{\theta}_{ij} = \sum_{l=0}^{K_j} \binom{K_j}{l} A_j a_i (\theta_i - \Theta_j)^{K_j - l} \mathrm{e}^{-\Lambda_j(\theta_i - \Theta_j)} (t - \theta_i)^{k_i + l} \mathrm{e}^{-(\lambda_i + \Lambda_j)(t - \theta_i)}$$

$$h^{\Theta}_{ij} = \sum_{l=0}^{k_i} \binom{k_i}{l} A_j a_i (\Theta_j - \theta_i)^{k_i - l} \mathrm{e}^{-\lambda_i(\Theta_j - \theta_i)} (t - \Theta_j)^{K_j + l} \mathrm{e}^{-(\Lambda_j + \lambda_i)(t - \Theta_j)}$$

$$h_{ij}(t) = A_j a_i (t - \Theta_j)^{K_j + k_i} \mathrm{e}^{-(\Lambda_j + \lambda_i)(t - \Theta_j)}$$

Proof. First we prove the validity of $h^{\Theta}_{ij}(t)$ and $h^{\theta}_{ij}(t)$. Without loss of generality we can assume that $\Theta_j > \theta_i$. We perform a translation of the terms along the t-axis by the amount $-\Theta_j$, i. e. $H^*_i(t) = H_i(t + \Theta_j)$ and multiply the resulting functions H^*_i, H^*_j.

$$\begin{aligned} H^*_i H^*_j &= (A_j t^{K_j} \mathrm{e}^{-\Lambda_j t})(a_i(t + (\Theta_j - \theta_i))^{k_i} \mathrm{e}^{-\lambda_i(t + (\Theta_j - \theta_i))}) \\ &= \sum_{l=0}^{k_i} \binom{k_i}{l} A_j a_i (\Theta_j - \theta_i)^{k_i - l} \mathrm{e}^{-\lambda_i(\Theta_j - \theta_i)} t^{K_j + l} \mathrm{e}^{-(\lambda_i + \Lambda_j)t} \end{aligned}$$

The required result is obtained by translation by the amount $+\Theta_j$. The formula $h_{ij}(t)$ for $\Theta_j = \theta_i$ can easily be derived in the same way. □

3.2. *Truncated θ-exponential polynomials*

In reality the execution times of the subtasks of a parallel program are bounded. We will show in Sect. 4 that this behavior must be taken into account, when modeling massively parallel programs. Thus, we need a class of distributions, which are suitable for the manipulation on a computer and also have finite domain. Truncated θ-exponential polynomials have these properties as we will show. Truncated θ-exponential polynomials (H_ω) are functions of the following form:

$$H_\omega(t) = \xi \sum_{i=1}^{n} H_i(t), \qquad H_i(t) = \begin{cases} 0 & \text{if } t < \theta_i \\ a_i(t - \theta_i)^{k_i} \mathrm{e}^{-\lambda_i(t - \theta_i)} & \text{if } \theta_i \le t \le \omega \\ \frac{1}{n\xi} & \text{if } t > \omega \end{cases}$$

with $\xi := \frac{1}{\sum_{i=1}^{n} H_i(\omega)}$, $a_i \in \mathbb{R}, \lambda_i \in \mathbb{R}^+_0, k_i \in \mathbb{N}_0, \theta_i \in \mathbb{R}$ and $\forall i : \omega > \theta_i$.

The integration and differentiation of individual terms has of course, no effect on the domain of these terms. The combination of two truncated θ-exponential polynomial terms produces a quantity with the domain $[\theta, \omega]$:

convolution:	$\theta = \theta_1 + \theta_2,$	$\omega = \omega_1 + \omega_2.$
multiplication:	$\theta = \max\{\theta_1, \theta_2\},$	$\omega = \min\{\omega_1, \omega_2\}$

The values of the terms are calculated inside the interval $[\theta, \omega]$ using the relations which were derived in Sect. 3.1 for non-truncated exponential terms. It is also possible to approximate any empirical density function by a truncated θ-exponential polynomial

to any required degree of accuracy. A proof of this statement is only given here in outline. We consider an empirical density function D_1 which takes non-zero values only in the interval $[d_1, d_2]$. The problem of fitting a θ-exponential polynomial to the density function D_1 is reduced to the fitting of a normal exponential polynomial to the empirical density function D_2 in the interval $[0, d_2 - d_1]$ by the assigning $\forall i : \theta_i = d_1$ as well as the translation of the density function D_1 to the interval $[0, d_2 - d_1]$. After fitting an exponential polynomial EP to the density function D_2 with the required accuracy, we consider the difference surface ΔD in the interval $(d_2 - d_1, \infty]$. ΔD is calculated as $\Delta D = \int_{d_2-d_1}^{\infty} EP(t)dt$. This surface is distributed over the domain $[0, d_2 - d_1]$ by the normalization when truncating the exponential polynomial. This certainly does not lead to an increase in the total error. The maximum deviation $|\max\{EP(t) - D_2(t) | t \in [0, d_2 - d_1]\}|$ is at most increased by the factor $\frac{1}{1-\Delta D}$. Thus an arbitrarily small error boundary also can be satisfied for a truncated θ-exponential polynomial by a suitable choice of the error boundary when fitting the exponential polynomial.

4. Evaluation of a large number n of parallel subtasks

Many parallel applications consist of the iteration of the parallel execution of n independent subtasks. Examples include the solution of linear equations of partial differential equations, search and optimization problems and simulation [8]. In our investigation we focus on the case of global synchronization, where all tasks on an iteration step have to wait for the termination of all tasks on the previous step.

We examine the core of this fundamental structure, i.e. the parallel processing of n independent subtasks, which is included in Fig. 1a. Assuming that the distribution function of the execution time is the same for all subtasks and processors, and that there are sufficient processors for all subtasks to be executed in parallel, we have to calculate the distribution function of a random variable $M_n := X_{\max} := \max\{X_i, \ldots, X_n\}$ where X_i are identically distributed random variables with distribution function $F(t)$[3]. Let

$$F_{\max}(t) = (F(t))^n.$$

The exact calculation of $F_{max}(t)$ is not possible for a large value of n because of numerical inaccuracies and memory requirements. Thus, we use extreme value theory to derive approximate formulas. We base our investigations on the following theorem, which is proven in [14][4].

Theorem 6. *If $\alpha_n > 0$, $\beta_n \in \mathbb{R}$, $n \geq 1$ exist, such that*

$$P\left[\frac{(M_n - \beta_n)}{\alpha_n} \leq x\right] = F^n(\alpha_n x + \beta_n) \to G(x)$$

for $n \to \infty$, then G is of the form (I), (II) or (III).

[3] $X_{\max}$ is also represented by M_n here, in accordance with the literature used, e.g. [14]. Also μ and σ are used to represent the mean and variance, respectively

[4] [14] is given as the source of various theorems below. However, the theorems are not always to be found there in exactly the same formulation. Some theorems have been reformulated for our purposes, to avoid introducing an extensive mathematical apparatus as Resnick does, for example, including von Mises functions. However the fundamental contents of the theorems have been kept, and we have adhered to the notation from [14] as closely as possible.

$$(I) \quad \Lambda(x) = \exp(-e^{-x}) \quad \text{for} \quad x \in \mathbb{R}$$

$$(II) \quad \Phi_\alpha(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ exp(-x^{-\alpha}) & \text{for} \quad x \geq 0 \end{cases} \quad \text{for one} \quad \alpha > 0$$

$$(III) \quad \Psi_\alpha(x) = \begin{cases} exp(-(-x)^\alpha) & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases} \quad \text{for one} \quad \alpha > 0.$$

We write below $F \in D(G)$ if there exist normalizing constants $\alpha_n > 0$, $\beta_n \in \mathbb{R}$ such that for $n \to \infty$:

$$F^n(\alpha_n x + \beta_n) = P[M_n \leq \alpha_n x + \beta_n] \to G(x).$$

$\Psi_\alpha(x)$ is a reflected Weibull distribution and is also called type-III extreme value distribution [14]. $\Lambda(x)$ is also called the Gumbel distribution or the type-I extreme value distribution (see [19]). x_0 is the "right-hand end" of the distribution function F; i.e. $x_0 = \sup\{y : F(y) < 1\}$. The following assertions can be made about the possible distributions for the limiting cases:

a) If $F \in D(\Psi)$, then $x_0 < \infty$.
b) If $F \in D(\Phi)$, then $x_0 = \infty$.
c) If $F \in D(\Lambda)$, then $x_0 < \infty$ or $x_0 = \infty$.

It is shown in Sects. 4.1 and 4.2 by comparing the limit distributions of θ-exponential polynomials and truncated θ-exponential polynomials that the choice of distribution class for the execution time of subtasks has a decisive influence on the distribution of the overall execution time of massively parallel structures, even when the initial distributions barely differ in the shape of the density.

4.1. Limit distribution of θ-exponential polynomials

The following theorem, which is also taken from [14], is useful for determining the limit distribution of θ-exponential polynomials and the values of α_n and β_n.

Theorem 7. *Let $z_0 < x_0$ be arbitrary but fixed. If*

a) F is absolute continuous in (z_0, x_0), possesses a negative second derivative F'' for all x in this range, and

b) $\lim_{x \to x_0} \frac{F''(x)(1-F(x))}{(F'(x))^2} = -1$

then $F \in D(\Lambda)$. The parameters can be determined by the formulae:

$$\beta_n = \left(\frac{1}{1-F}\right)^{-1}(n) \tag{6}$$

$$\alpha_n = \left(\frac{1-F}{F'}\right)(\beta_n) \tag{7}$$

Thus, it can be shown that limit distributions from the class of θ-exponential polynomials are Gumbel distributions, i.e:

Theorem 8. *If $F \in H$ then $F \in D(\Lambda)$.*

Proof. The proof of this statement is given in [3]. □

Note 1. Since in this section we are only interested in θ-exponential polynomials as a means of describing the execution times of parallel programs, we shall assume below that θ-exponential polynomials are distribution functions or density functions from the class of θ-exponential polynomials.

Thus, we know the limit distribution $F^n(t), n \to \infty$ of θ-exponential polynomials[5]

$$\lim_{n\to\infty} (F(\alpha_n t + \beta_n)^n) = G_n(\alpha_n t + \beta_n) = exp(-e^{-t}).$$

We must next calculate the parameters α_n and β_n. The parameters α_n and β_n cannot be exactly determined analytically for a general $F(t) \in H$. Rearranging equation (6) as

$$(1 - F)(\beta_n) - \frac{1}{n} = 0$$

leads to the problem of determining the zeroes of θ-exponential polynomials. Numerical methods can be used to determine the values of β_n. Since the distribution function $F \in H$ increases monotonically in the range $[\min_i\{\theta_i\}, \infty]$, efficient numerical methods can be applied.[6] Besides determining the parameters α_n and β_n by finding the zeroes of equations, it is also possible to derive the parameters analytically. The following is proven in [3]:

Theorem 9. *If $F \in H$, then*

$$\beta_n = \frac{1}{\lambda^{(\min)}}(ln\, n + k^{(\max)} ln\, ln\, n + ln\, \alpha^m - (1 + k^{(\max)})(ln\, \lambda^{(\min)})) \tag{8}$$

is a suitable value for the normalization constant β_n, where

$$\alpha_{(m)} = \sum_{\substack{i=1 \\ k_i = k^{(\max)} \\ \lambda_i = \lambda^{(\min)}}}^{n} a_i e^{\theta_i \lambda^{(\min)}}$$

It is easy to calculate α_n using equation (7) when β_n is known.

$$\alpha_n = \frac{1 - F(\beta_n)}{f(\beta_n)}$$

There are simpler equations for calculating the parameters α_n and β_n in various special cases of the θ-exponential polynomials[7].

Exponential distribution:

$$\alpha_n = \frac{1}{\lambda} \qquad \beta_n = \frac{ln\, n}{\lambda} \tag{9}$$

[5] Since mixed Erlang distributions and Cox distributions with real coefficients are special cases of θ-exponential polynomials, we can deduce directly that all the phase-type distributions with real coefficients belong to class $D(\Lambda)$, i.e. their limit distribution is the Gumbel distribution.

[6] For our purposes the simple, though in general, slow method of bisection is adequate.

[7] Since it is sufficient to satisfy the asymptotic exactness of the determining equations, it is possible that there are other formulas which can produce tighter bounds; e.g. it has been found that for Erlang distributed random variables it is better to neglect the contribution of the term $(-ln(k!))$ which was originally present.

Erlang distribution:

$$\alpha_n = \frac{1}{\lambda} \qquad \beta_n = \frac{1}{\lambda}(ln\, n + k\, ln\, ln\, n) \tag{10}$$

θ-Exponential polynomials with $\forall(i = 1, \ldots, n) : \theta_i = \theta$:

$$\alpha_n = \frac{1}{\lambda^{(\min)}} \qquad \beta_n = \frac{1}{\lambda^{(\min)}}(ln\, n + k^{(\max)} ln\, ln\, n) + \theta \tag{11}$$

It follows from [14], proposition 1.1 and 2.1, that the extreme value distribution to which $F(t)^n$ converges is known, and also that the moments $E^j(F(t))^n)$ converge to the corresponding moments $E^j(G_n(t))$. The following formulas give the mean value $E(G_n) = \mu_n$ and the variance $Var(G_n) = \sigma^2$ of the Gumbel distribution (see [13]).

$$\mu_n = \beta_n + \alpha_n \gamma \tag{12}$$

$$\sigma_n^2 = \alpha_n^2 \delta \tag{13}$$

Here γ is the Euler constant

$$\gamma = \lim_{n\to\infty} \gamma_n \approx 0.577216, \qquad \gamma_n = \sum_{i=1}^{n} \frac{1}{i} - ln\, n \tag{14}$$

and

$$\delta = \lim_{n\to\infty} \delta_n = \frac{\pi^2}{6}, \qquad \delta_n = \sum_{i=1}^{n} \frac{1}{i^2}. \tag{15}$$

In the special case where the distribution is exponential ($F(t) = 1 - e^{-\lambda t}$) it follows from (12), (13), and (9) that

$$\mu_n = \frac{1}{\lambda}(ln\, n + \gamma), \qquad \sigma_n^2 = \frac{1}{\lambda^2}\delta. \tag{16}$$

It is interesting that the replacement of γ by γ_n and δ by δ_n leads to the exact results

$$\mu_n = \frac{1}{\lambda}\sum_{i=1}^{n} \frac{1}{i}, \qquad \sigma_n^2 = \frac{1}{\lambda^2}\sum_{i=1}^{n} \frac{1}{i^2}. \tag{17}$$

These are proven, for example, in [9] or [7].

The estimation functions 9 - 11 have been tested for particular distributions, to obtain an impression of the convergence behavior of the exact results and the asymptotic approximations. Simulation results are also presented for comparison.

In Fig. 3, μ_n is an upper boundary for the expectation of the maximum of n independent random variables with identical distributions. It can be seen that this formula provides a very good estimate of the mean value for small n ($n < 10$). This boundary, which is independent of the distribution, is derived, for example, in [2].

It emerges that the estimated values for the expectation agree very well with the values obtained by simulation when the numerical values of the parameters α_n, β_n are obtained by calculating the zeroes of the exponential polynomial. If the distribution is Erlang, the parameters α_n and β_n can be estimated successfully using formula (10). The general approximation formula (8) is only suitable for small n ($n < 100$) if there is a dominating $\lambda^{(\min)}$. This follows from the fact that all other λ_i, independent of the number of stages ($k_i + 1$), are neglected when calculating the parameters.

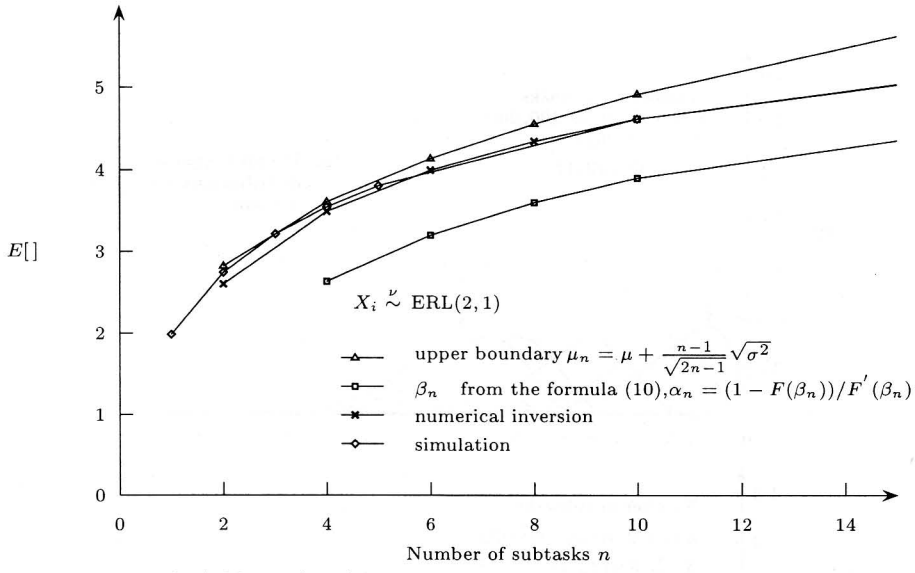

Fig. 3. Mean value of the total execution time for a parallel structure

4.2. *Limit distribution of truncated θ-exponential polynomials*

Because realistic run-times of parallel subtasks are finite, they are better represented by distributions with a finite domain. We use truncated θ-exponential polynomials to examine the effects of assigning distributions with a finite domain to the subtasks of massively parallel structures (Number of subtasks $n > 100$).

The following theorem provides information about the limit distribution of random variables with finite domain ($x_0 < \infty$). The proof is given in [14].

Theorem 10. *Let $D(\Psi_\alpha)$ be the set of distributions F whose extreme value distribution G is of type (III). If the distribution function F has a right-hand end point ($x_0 < \infty$), is absolutely continuous in a neighborhood at the left of x_0, has a positive density F' there, and there exists an $\alpha > 0$ such that*

$$\lim_{x \to x_0} \frac{(x_0 - x)F'(x)}{1 - F(x)} = \alpha, \tag{18}$$

then $F \in D(\Psi_\alpha)$. The parameters are determined by the equations

$$\alpha_n = x_0 - \left(\frac{1}{1 - F}\right)^{-1}(n),$$

$$\beta_n = x_0. \tag{19}$$

It is important that the asymptotic mean and the variance of distributions $F \in D(\Psi_a)$ be given by:

$$\mu_\infty = x_0$$

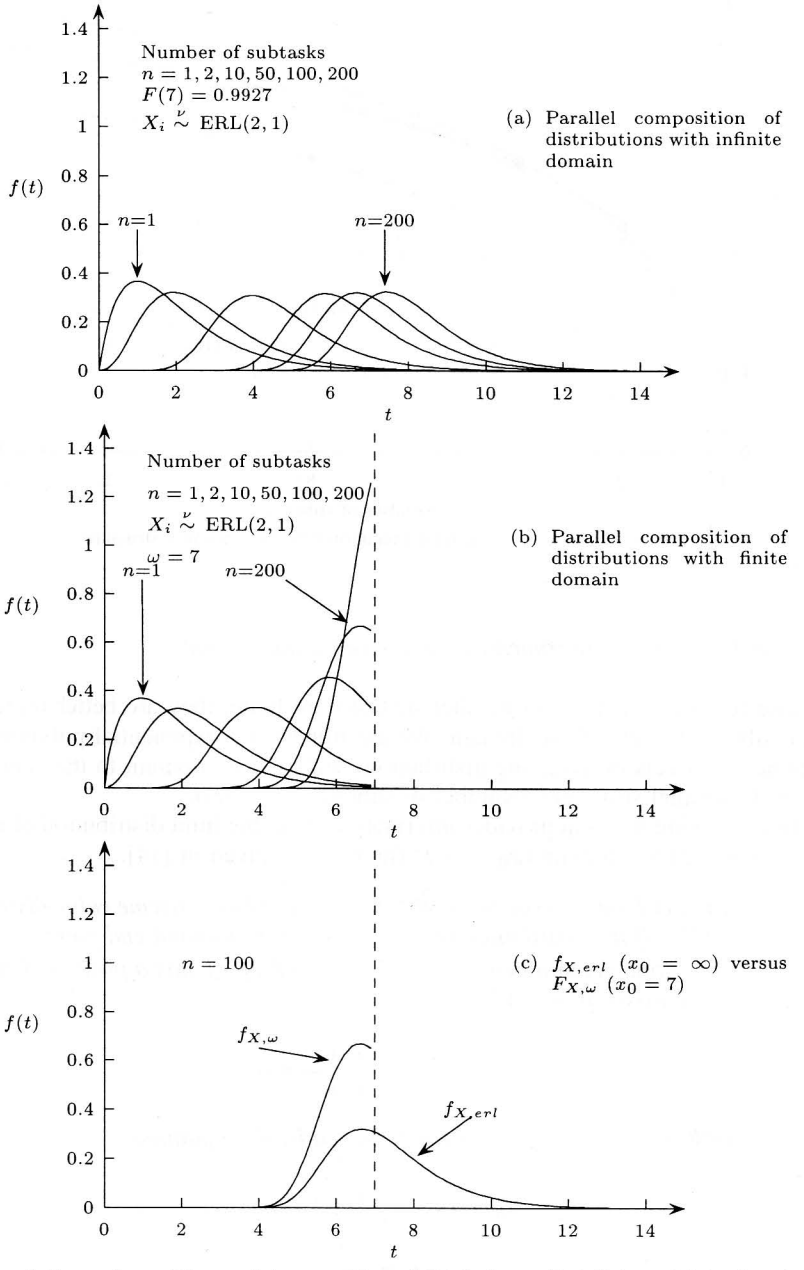

Fig. 4. Comparison of the parallel composition of distributions with infinite and finite domain

$$\sigma_\infty^2 = 0$$

In general it is true that

$$F \in D(\Psi_\alpha) \rightarrow \lim_{n\to\infty} \hat{\lceil} M_n = x_0,$$

where $lim\hat{\lceil}$ means the almost sure convergence [14].

Interesting consequences for the modeling and analysis of parallel programs follow from the fact that different extreme value distributions occur, depending on the domain of the initial distribution. We use the class of truncated θ-exponential polynomials to demonstrate the consequences more clearly. The limit distribution of the truncated θ-exponential polynomial is the Weibull distribution, i.e.:

Theorem 11. *If $F \in H_\omega$, then $F \in D(\Psi_1)$.*

This is proven in [3].

Figure 4a shows the densities for the maximum of random variables with Erlang-2 distributions with parameter $\lambda = 1$ for different numbers of parallel subtasks n. Figure 4b presents the results for corresponding truncated Erlang density functions with $\omega = 7$. Although the shapes of the Erlang and truncated Erlang densities, $f(t)$ and $f_\omega(t)$, are almost identical ($F_{ERL(2,1)}(7) = 0.9927$), the resulting density functions $f_{X,erl}$ and $f_{X,\omega}$ diverge even for small values of n. This effect is demonstrated in Fig. 4c in more detail.

This indicates, that the use of distributions with infinite domain is only acceptable, when the degree of parallelism (i. e. n) is very low. Distributions with finite domain are needed to model massively parallel systems.

5. Conclusion

In this paper we addressed the problem of accurate representation of the task execution times, when modeling parallel programs by task precedence graphs.

We introduced a generalization of exponential polynomials that enables the efficient representation of typical task run-times with a small number of stages k_i. This extension increases the efficiency of calculations and reduces the numerical problems for the evaluation of series - parallel graphs. Additional advantages of truncated θ-exponential polynomials are the possibility to combine deterministic and stochastic execution times and to guarantee minimum as well as maximum execution times. We showed that this new class of distributions has the closure and approximation properties required for accurate and efficient evaluation on a computer.

We then used θ-exponential polynomials and truncated θ-exponential polynomials together with theorems from extreme value theory to derive approximations for the distribution function and the mean and variance of the total computation time of massively parallel structures. The comparison of the approximations with results from simulation showed the accuracy of our formulas

Finally our survey provided the insight, that the use of distributions with infinite domain is only acceptable, when structures with a very small degree of parallelism are modeled. Distributions with finite domain are needed to model massively parallel systems.

References

1. Bolch, G., Fleischmann, G., Gente, M., Schiller. B.: GIANT: A tool for performance evaluation of parallel programs. Q-PASSPORT, Newsletter of the Dutch User Group Queueing/Performance AnalysiS Software (Q-PASS), (8), 1990.
2. David, H. A.: Order Statistics. New York: John Wiley 1981.
3. Fleischmann, G.: Leistungsbewertung paralleler Programme für MIMD-Architekturen: Modellbildung und mathematische Analyse. PhD thesis, Universität Erlangen-Nürnberg, 1990.
4. Fleischmann, G., Werner, G.: Modellierung und Bewertung paralleler Programme mit Hilfe von Markovketten. In Operations Research. Berlin: Springer 1989.
5. Fromm, H. J.: Multiprozessor-Rechenanlagen: Programmstrukturen, Maschinenstrukturen und Zuordnungsprobleme. PhD thesis, Universität Erlangen-Nürnberg, 1982.
6. Gelenbe, E., Pujolle, G.: Introduction to Queueing Networks. London: John Wiley & Sons 1987.
7. Herzog, U.: Performance characteristics for hierarchical organized multiprocessor systems with generally distributed processing times. In Archiv für Elektronik und Übertragungstechnik, volume 34. 1980.
8. Herzog, U., Hoffmann, W., Kleinöder, W.: Performance modeling for hierarchically organized multiprocessor computer systems. In Proceedings of the 1979 International Conference on Parallel Processing. University of South Florida, 1979.
9. Hoffmann, W.: Warteschlangenmodelle für die Parallelverarbeitung. PhD thesis, Universität Erlangen, 1978.
10. Kleinöder, W.: Stochastische Bewertung von Aufgabenstrukturen für hierarchische Mehrrechnersysteme. PhD thesis, Universität Erlangen-Nürnberg, 1982.
11. Martin, J. J.: Distribution of the time through a directed acyclic network. Operations Research, 13(1):46–66, 1965.
12. Mohan, J.: Performance of Parallel Programs: Model and Analyses. PhD thesis, Carnegie Mellon University, 1984.
13. Mood, A. M., Graybill, F. A., Boes, D. C.: Introduction to the theory of statistics. London: McGraw-Hill 1974.
14. Resnick, S. I.: Extreme values, regular variation, and point processes, volume 4 of Applied Probability. New York: Springer 1987.
15. Sahner, R.: A Hybrid, Combinatorial Markov Method of Solving Performance and Reliability Models. PhD thesis, Dep. Comput. Sci., Duke Univ., Durham, 1986.
16. Sahner, R. A., Trivedi, K. S.: SPADE: A tool for performance and reliability evaluation. In N. Abu El Ata, editor, Modelling Techniques and Tools For Performance Analysis, pages 147–163, 1986.
17. Sahner, R. A., Trivedi, K. S.: Performance and reliability analysis using directed acyclic graphs. IEEE Transactions on Software Engineering, SE-13(10):1105–1114, October 1987.
18. Schmickler, L.: Approximation von empirischen Verteilungsfunktionen mit Erlangmischverteilungen und Coxverteilungen. In U. Herzog, M. Paterok, editors, Informatik Fachberichte 154: Messung, Modellierung und Bewertung von Rechensystemen, GI/NTG-Fachtagung, pages 118–133. Berlin: Springer 1987.
19. Wilrich, P. T., Henning, H. J.: Formeln und Tabellen der angewandten mathematischen Statistik. Berlin: Springer 1987.

This article was processed by the author using the LaTeX style file *pljour1* from Springer-Verlag.