INFORMATIQUE THÉORIQUE ET APPLICATIONS

F. AFRATI
A. STAFYLOPATIS

Performance considerations on a random graph model for parallel processing

Informatique théorique et applications, tome 27, n° 4 (1993), p. 367-388.

<http://www.numdam.org/item?id=ITA_1993__27_4_367_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 27, n° 4, 1993, p. 367 à 388)

PERFORMANCE CONSIDERATIONS ON A RANDOM GRAPH MODEL FOR PARALLEL PROCESSING (1) (*)

by F. AFRATI and A. STAFYLOPATIS (2)

Communicated by J. GABARRO

Abstract. – *Consider a random directed acyclic graph (dag) with nodes* 1, 2, . . . , *n, and an edge from node i to node j (only if i>j) with fixed probability p. Such a graph can be thought of as the task graph associated with a job and thus it serves as a parallel processing model; the vertices correspond to tasks and the edges correspond to precedence constraints between tasks. In this case, the length of the graph corresponds to the parallel processing time of the job (an infinite number of available processors is assumed) and the width of the graph corresponds to the parallelism of the job. We estimate here the average length of the random dag (that is, the average processing time of the job) as a function of the probability p and the number of tasks n by establishing tight lower and upper bounds. The lower (resp. upper) bound is determined as being equal to the average length of a random dag considerably simpler to manipulate than the original one. Furthermore, the asymptotic behaviour of the average length is studied and the results obtained improve previously published results. Finally, asymptotic results are obtained concerning the average width of the task graph; it is shown that the average width tends to* $1/p$ *as* $n \to \infty$.

Résumé. – *Un graphe dirigé acyclique aléatoire avec n nœuds est considéré, auquel une arête du nœud i vers le nœud j (i>j) existe avec une probabilité fixe p. Un tel graphe peut servir comme modèle de calcul parallèle, où les nœuds représentent les tâches composant un travail et les arêtes représentent des contraintes de précédence entre tâches. Ainsi, la longueur du graphe correspond au temps de traitement parallèle du travail (en supposant un nombre infini de processeurs) et la largeur du graphe correspond au parallélisme du travail. La longueur moyenne du graphe aléatoire (qui représente le temps moyen de traitement d'un travail) est estimée ici en fonction de la probabilité p et du nombre n de tâches, à travers le calcul d'une borne inférieure et d'une borne supérieure. Les bornes sont déterminées comme les longueurs moyennes de deux autres graphes aléatoires qui sont beaucoup plus faciles à manipuler que le graphe original. De plus, le comportement asymptotique de la longueur moyenne est étudié et les résultats obtenus améliorent des résultats publiés auparavant. A la fin, des résultats asymptotiques concernant la largeur moyenne du graphe montrent qu'elle tend vers* $1/p$ *pour* $n \to \infty$.

(*) Received May 1992, accepted July 1992.
(1) Part of this work was supported by the General Secretariat of Research and Technology of Greece under Grant No. 9707.

(2) Computer Science Division, Department of Electrical Engineering National Technical University of Athens, 15773 Zographou, Athens, Greece.

Informatique théorique et Applications/Theoretical Informatics and Applications
0988-3754/93/04 367 22/$4.20/

1. INTRODUCTION

Consider a set of tasks which are to be executed in parallel under certain precedence constraints. Let us number the tasks 1, 2, . . ., n, thus obtaining an ordered set; for example, a natural numbering of the tasks would be according to the order of their appearance in the system or to some strict priority rule. For each task i and each one of its preceding (in the numbering) tasks j there is a fixed probability p that task i cannot start execution until the execution of j is completed (because, for example, results computed by the latter are needed by task i, or because they both share some common resource). We are interested in performance measures for the parallel execution of a job consisting of the above defined set of tasks, in the case where task execution times are deterministic and there are infinitely many available processing units.

A straightforward application of this model is in parallel computation; consider a general program, which, depending on the input data, asks for the execution of several processes from an ordered set, where each process i needs with probability p the results of process j ($i>j$). In database applications, if there is a conflict between two transactions requiring access to the same entities, then the transaction that arrived later has to wait until the service of the other transaction has been completed (known as locking the entities for every other transaction except the current one). In general, the above described model can be useful in the representation of concurrent systems, in which conflicts arise during the execution of interdependent tasks [4]. In all cases, the probability p represents the very complicated phenomenon of interdependence between tasks. Of course, this is a simplified description with respect to realistic situations, but it provides a means of understanding the fundamental properties of this kind of parallel processing.

Several probabilistic models have been developed so far concerning random tree or graph structures for the performance evaluation of parallel processing schemes. In [2], a graph model related to the model considered here is analysed, in which vertices represent transactions and edges represent conflicts. The measure of interest is the effective concurrency, defined as the expected number of transactions that can run concurrently, assuming that there is always a fixed number of transactions present in the system. Bounds on the above quantity are obtained for three different general classes of concurrency control methods. An approximate analysis for the asymptotic behaviour of a model similar to the one studied here is carried out in [3] considering exponentially distributed task execution times. Also, maximum likelihood estimators are derived for the value of the probability p given

specific characteristics of a particular task graph. An analogous problem is studied in [5] where an $M/D/\infty$ queue is considered with the same probabilistic description of precedence relations among arrivals. The investigation of stability conditions for the queue leads to the study of the asymptotic length of a random directed acyclic graph, which is the same as the one in our model. Upper and lower bounds are obtained for the rate of increase of the length per unit arrival as a function of the interdependence probability p. The problem of the length of a similar graph model is studied in [1] in the special case where $p=1/2$ and a closed form asymptotic formula is derived by getting upper and lower bounds.

In this paper, we derive upper and lower bounds on the average length (mean processing time) of the random task graph, for all values of the number n of tasks in the graph and all values of the probability p. In particular, in order to estimate the lower (resp. upper) bound we define a simpler random dag and we show that its average length is less (resp. greater) than the average length of the original random dag. It turns out that the average length of the new random dag is much simpler to compute and it is a tight bound. Furthermore, we study the asymptotic behaviour of the average length and obtain bounds, which improve the ones obtained in [5]. Finally, we investigate the asymptotic behaviour of the mean width of the graph, which represents the expected number of tasks that can run in parallel. We prove that the asymptotic value of the average width (as $n \to \infty$) for any specific level is $1/p$, thus providing the exact relationship between p and the parallelism of the graph, which was suggested by the bounds obtained in [5].

In the next section the random graph model is discussed and in Section 3 bounds are derived on its average length by means of simpler graph models which can be obtained from the original one. In Section 4 we study the asymptotic behaviour of the average length, whereas in Section 5 we present results concerning the average width of the graph.

2. THE RANDOM DAG MODEL

Consider an ordered set of tasks indexed by the integers 1, 2, . . ., n. The precedence constraints governing the execution of this set are well described by a directed acyclic graph (dag) G with n nodes, the *task graph*. Each task forms a node of the graph and an edge from node i to node j denotes that task j must have completed execution before task i starts being executed. There is an edge from node i to node j with fixed probability p if $i>j$ and

with zero probability if $i \leq j$ (it is easy to see that acyclicity is guaranteed this way). An example of a task graph with $n=13$ is shown in Figure 1.

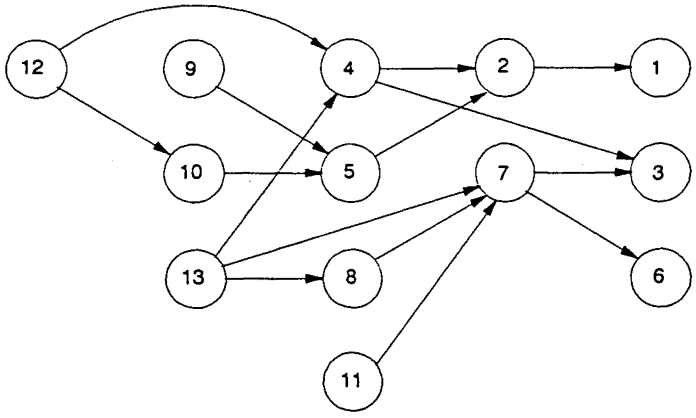

Figure 1. – A Task Graph Realisation.

We will assume that task execution times are deterministic and equal to 1 and that there are infinitely many available processing units which can operate in parallel. The *processing time* of a task graph is defined to be the minimum amount of time needed to execute all the tasks respecting the precedence constraints (and with an infinite number of available processors). It is easy to observe that the processing time of a given task graph equals the length of its maximum path (in terms of the number of nodes on the path); we will call it the length of the graph and will denote it by $L(G)$. Hereafter, we will use the terms processing time and length of the task graph interchangeably. We are interested in the average length $\overline{L(n)}$ of a family of random dags with n nodes, the edges of which are drawn stochastically according to the above described procedure.

For our analysis it is more convenient to consider the dag as being topologically sorted; that is, partition the set of nodes into the least possible number of classes, called levels, such that there are no edges leading from any node of level j to nodes of any level with index greater than or equal to j, and there is at least one edge from each node of level j to some node of level $j-1$. Nodes with outdegree 0 are put at level 1 and can immediately assume execution at the beginning of the processing of the graph, since they are free of precedence constraints. The task graph of Figure 1 is topologically sorted. Hereafter, when we refer to a task graph, we will consider that it is in topologically sorted form. The length of an acylic directed graph equals

the number of levels in the topological sorting. Then, the parallel execution of the task graph can be viewed as taking place level by level. For a given task graph G, let $W_k(G)$ denote the *width* of the k-th level of the graph, *i.e.*, the number of nodes at that level. Then $\overline{W_k(n)}$ is the *average width* of the k-th level over all random graphs with n nodes constructed as described above. The width represents the number of tasks that can run concurrently. The average width of a level expresses the *effective parallelism* of that level.

In order to compute the mean length of a random graph, it is necessary to consider the probabilities concerning the number of nodes at each level of the topologically sorted graph, which yields a state-space impossible to handle. Fortunately, we can introduce simpler models that can be shown to provide upper and lower bounds on the desired quantity.

Before proceeding with the study of the bounds, it would be helpful to recall how we can construct a random dag G with n nodes in terms of its partition into levels. Consider the ordered set of n tasks or nodes. Note 1 will be put at the first level. Node 2 will be put randomly with probability p at the second level and with probability $1-p$ at the first level, since edge (2, 1) appears with probability p. In general, if $i-1$ nodes constitute already the graph and are partitioned into levels, the position of the i-th node is determined as follows: For each node $k=1, 2, \ldots, i-1$, an edge is drawn at random from i to k with probability p. If there are no edges from i to nodes belonging to a level grater than j, and there is at least one edge from node i to a node of level j, then node i is put at level $j+1$. It is obvious that each node i either will be put at one of the already formed levels, or will initiate a new level. Each particular realisation of the graph, that is, each particular realisation of the random variables involved in its construction, is a *sample path* of the random dag.

3. BOUNDS ON THE MEAN LENGTH

The random dag G, introduced in the previous section, is too complicated to allow for an efficient computation of its mean length. Therefore, we introduce here two simpler random dags G_1 and G_2, with the same number of nodes as G, such that their mean lengths constitute a lower and an upper bound on the mean length of G, respectively.

3.1. Lower bound

We define the random dag G_1 as follows: for every sample path of G we construct a sample path of G_1 by applying Procedure 1.

PROCEDURE 1: Node 1 will be put at the first level and until some node initiates the second level the procedure is the same as for G. Once there is more than one level the following general step is performed for each node i added to the graph. Delete all the outgoing edges from node i, except the ones that lead from i to either a node of the last level or to the node that initiated the second to last level in the graph formed so far. In other words, if node i draws an edge to any node of the last level, then it forms a new level; otherwise, there are two possibilities: if node i draws an edge to the node that initiated the second to last level, then it is put at the last level, if not it is put at the first level.

The graph of Figure 2 is the graph corresponding to the graph of Figure 1 according to the above procedure.

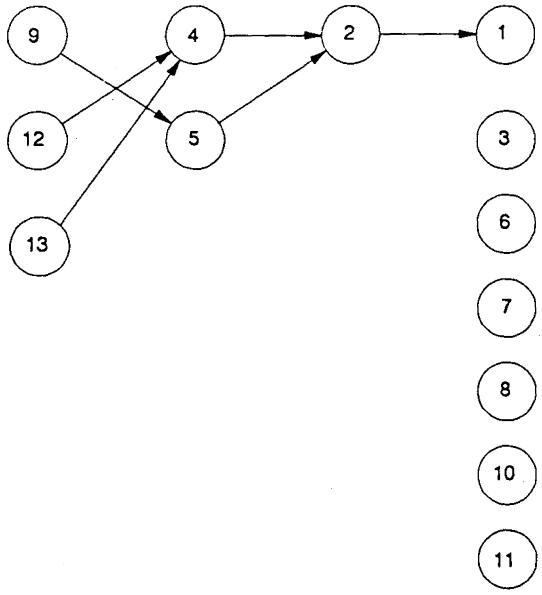

Figure 2. – Lower Bound Graph for the Graph of figure 1.

PROPOSITION 1: *The length $L_1(G_1)$ of a graph G_1 constructed by Procedure* 1 *is less than or equal to the length $L(G)$ of the corresponding original graph G, for every sample path.*

Proof: By induction on the number of nodes n. For $n=1$ the proposition is trivially true. We are going to prove the inductive assertion: *for every sample path, the node i ($i=1, 2, \ldots, n$) is put at a lower or the same level in graph G_1 than it is put in graph G.* Suppose this is true for graphs G and G_1 having n nodes. If we add a new node $n+1$ to G and G_1, we observe that the set of outgoing edges from node $n+1$ in G_1 is a subset of the set of outgoing edges from node $n+1$ in G. It is a consequence of this observation and the inductive hypothesis that node $n+1$ is also put at a lower or the same level in G_1 than it is put in G. Finally, it is a straightforward consequence of the inductive assertion that the length of G_1 is less than or equal to the length of G. ■

From Proposition 1 follows that the mean length $\overline{L_1(n)}$ constitutes a lower bound on the mean length $\overline{L(n)}$ of the original graph model.

We are going to derive the lower bound $\overline{L_1(n)}$. Define a sequence of indicator random variables X_k as follows:

$$X_k = \begin{cases} 1 & \text{if adding the } k\text{-th node increases the length of the graph} \\ 0 & \text{otherwise} \end{cases}$$

We will then have:

$$\overline{L_1(n)} = \sum_{k=1}^{n} \bar{X}_k \tag{1}$$

where $\bar{X}_k = \Pr[X_k = 1] = \Pr$ [node k initiates a new level]. We notice that the above probability depends directly only upon the number of nodes at the last level of the graph. We are thus led to define $P(n, j)$, $1 \leqq j \leqq n$, to be the probability that a graph G_1 with n nodes possesses exactly j nodes at the last level. In fact, the probabilities $P(n, j)$, $n \geqq 1$, constitute the transient-state solution of a Markov process representing the evolution of the graph, and must satisfy the following set of equations:

$$P(2, 1) = P(1, 1)p \tag{2}$$

$$P(n, 1) = \sum_{k=1}^{n-1} P(n-1, k)[1-(1-p)^k] + P(n-1, 1)(1-p)^2, \qquad n > 2 \tag{3}$$

$$P(n, j) = P(n-1, j-1)(1-p)^{j-1}p + P(n-1, j)(1-p)^{j+1}, \quad 2 \leqq j \leqq n-2 \tag{4}$$

$$P(n, n-1) = P(n-1, n-2)(1-p)^{n-2}p, \qquad n>2 \tag{5}$$

$$P(n, n) = P(n-1, n-1)(1-p)^{n-1} \tag{6}$$

The general step for the construction of the graph G_1 is mainly reflected by equations (3) and (4). Equation (3) refers to the case where the last level of the graph contains a single node, either because a new level has just been initiated or because the size of the last level was already equal to one and the newly added node was put at the first level. Accordingly, equation (4) refers to the case where the size of the last level is greater than one; again two possibilities must be considered depending on whether the newly added node was put at the last or at the first level of the graph.

Starting with $P(1, 1)=1$ the probabilities $P(n, j)$ can be easily computed from the above recursive equations.

To compute $\overline{L_1(n)}$ we need $\Pr[X_n=1]$, which can be expressed as follows:

$$\Pr[X_n=1] = \sum_{k=1}^{n-1} P(n-1, k)[1-(1-p)^k], \qquad n>1 \tag{7}$$

or by using (3):

$$\Pr[X_n=1] = P(n, 1) - P(n-1, 1)(1-p)^2, \qquad n>2 \tag{8}$$

Summation of (8) over all $n>2$ and substitution of the result in (1) yields (note that $\bar{X}_1 = P(1, 1)$ and $\bar{X}_2 = P(2, 1)$):

$$\overline{L_1(n)} = 1 + \sum_{k=2}^{n-1} P(k, 1)[1-(1-p)^2] + P(n, 1) \tag{9}$$

We do not know if a closed form solution can be obtained for $\overline{L_1(n)}$. However, equation (9) together with equations (2) to (6) provide for an efficient computation of the lower bound $\overline{L_1(n)}$, since at each step one only needs the values computed at the previous step.

3.2. Upper bound

Again we create a random graph G_2 as follows: for every sample path of G we construct a sample path of G_2 by applying Procedure 2.

PROCEDURE 2: Node 1 will be put at the first level and the following general step is performed for each node i ($i>1$) added to the graph. If node i has an outgoing edge to any node at the last level of the graph formed so far, then

it initiates a new level; otherwise, node i is put at the last level with a single outgoing edge to the node of the second to last level with higher task number. (Notice that until a second level is formed, the procedure is the same as for G.)

The graph of Figure 3 is the graph corresponding to the graph of Figure 1 according to the above procedure.

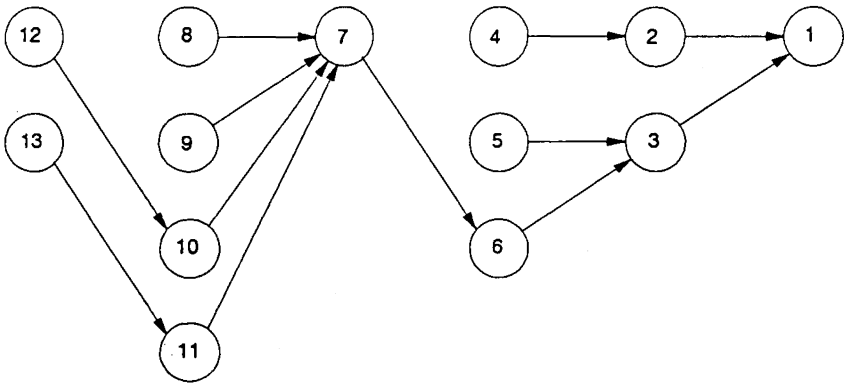

Figure 3. – Upper bound graph for the graph of figure 1.

PROPOSITION 2: *The length $L_2(G_2)$ of a graph G_2 constructed by Procedure 2 is greater than or equal to the length $L(G)$ of the corresponding original graph C, for every sample path.*

Proof: On the same lines with the proof of Proposition 1. ■

From Proposition 2 follows that the mean length $\overline{L_n(n)}$ constitutes an upper bound on the mean length $\overline{L(n)}$ of the original model.

To compute $\overline{L_2(n)}$ we proceed as in the case of the lower bound:

$$\overline{L_2(n)} = \sum_{k=1}^{n} \bar{X}_k \tag{10}$$

where the random variables X_k are defined exactly as before. Let us define the probability $P(n, j)$, $1 \leq j \leq n$, that a graph G_2 with n nodes possesses exactly j nodes at the last level. The probabilities $P(n, j)$, $n > 1$, satisfy the

set of equations:

$$P(n, 1) = \sum_{k=1}^{n-1} P(n-1, k)[1-(1-p)^k] \tag{11}$$

$$P(n, j) = P(n-1, j-1)(1-p)^{j-1} \tag{12}$$

and can be easily computed starting with $P(1, 1) = 1$.

We have for $\Pr[X_n = 1]$:

$$\Pr[X_n = 1] = \sum_{k=1}^{n-1} P(n-1, k)[1-(1-p)^k] = P(n, 1), \qquad n > 1 \tag{13}$$

and by substitution in (10) we finally obtain for the mean length:

$$\overline{L_2(n)} = \sum_{k=1}^{n} P(k, 1) \tag{14}$$

Note that, after some algebraic manipulation using (11) and (12), we can express $P(n, 1)$ as a function of the probabilities $P(l, 1)$ for $1 \leqq l < n$:

$$P(n, 1) = \sum_{k=1}^{n-1} P(n-k, 1)(1-p)^{(k-1)k/2}[1-(1-p)^k] \tag{15}$$

Again, we do not know if the recurrence above results in a closed form solution for $\overline{L_2(n)}$, but it can be the basis for an efficient computation.

Numerical examples concerning the lower and upper bounds on the mean length are plotted in figures 4 and 5, in comparison with simulation results for the original graph model. Simulation experiments were carried out using the independent replication method with a 95% confidence interval. For each case considered the simulation program was executed until the confidence interval was less than 5% of the estimated mean value or until 1 000 replications had been generated.

The lower bound seems to be closer to the simulation results than the upper one, especially for small values of the probability p. Experimental results show that the bounds are tight, although a theoretical justification of that would require a more involved argument. In general, we can say that the bounds behave well and that the corresponding computational algorithms are much faster than simulation.

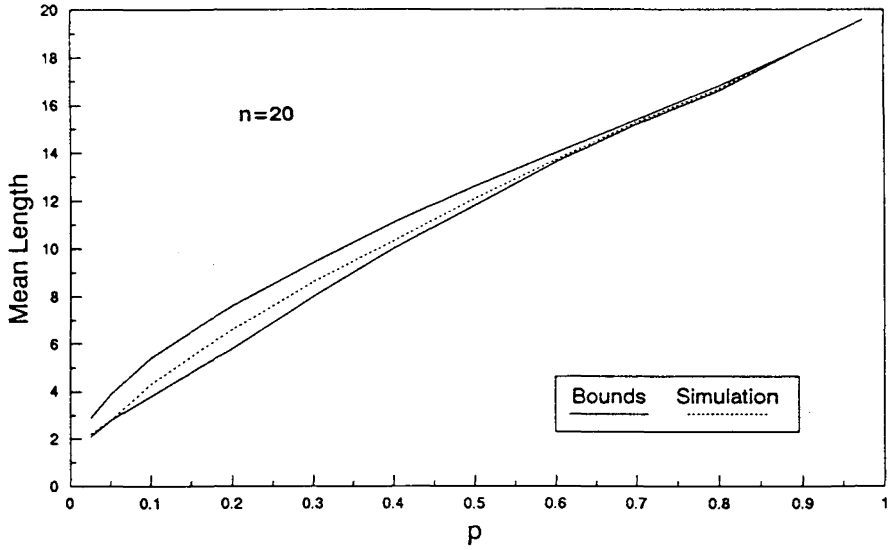

Figure 4. – Bounds and simulation results for a graph with 20 nodes.

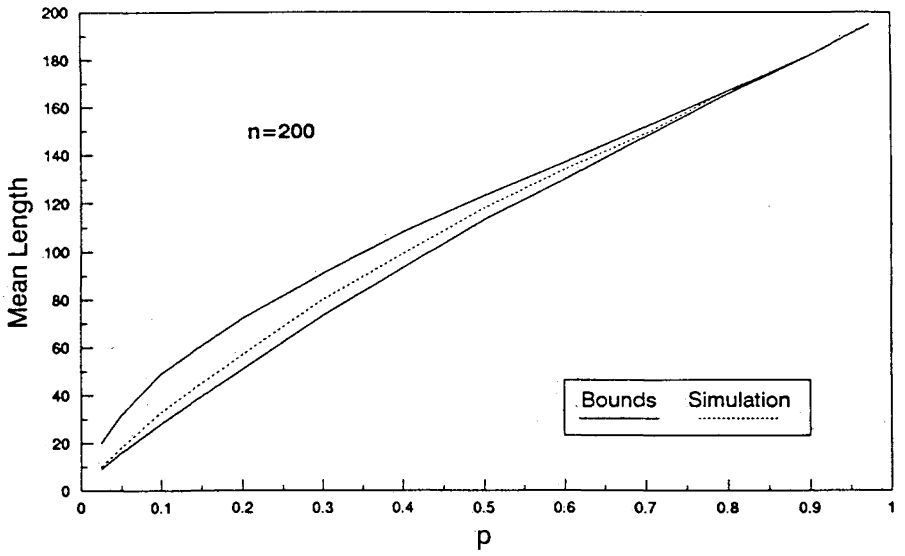

Figure 5. – Bounds and simulation results for a graph with 200 nodes.

4. ASYMPTOTIC BEHAVIOUR OF THE MEAN LENGTH

Let us now consider the behaviour of the mean length of the graph G as the number of nodes n becomes very large. We will be interested in the asymptotic rate α of increase of the length per node added to the graph:

$$\alpha = \lim_{n \to \infty} [\overline{L(n)} - \overline{L(n-1)}] \tag{16}$$

In an analogous manner, we define the asymptotic rates of increase α_1 and α_2 corresponding to the graph models G_1 and G_2, respectively, which will be shown to constitute bounds on α.

4.1. Lower bound

Consider the rate α_1 of increase of the length for a graph G_1 with n nodes as n becomes very large. The rate depends upon the distribution of the number of nodes at the last level of the graph. This latter is a random variable whose behaviour is described by a discrete-parameter Markov process with state-space $\{1, 2, \ldots\}$. Assume there exists a steady state distribution $\pi(j)$; then it must satisfy the following equations:

$$\pi(1) = \sum_{k=1}^{\infty} \pi(k)[1-(1-p)^k] + \pi(1)(1-p)^2 \tag{17}$$

$$\pi(j) = \pi(j-1)(1-p)^{j-1}p + \pi(j)(1-p)^{j+1}, \qquad j>1 \tag{18}$$

as well as the normalizing condition $\sum_{j=1}^{\infty} \pi(j) = 1$.

Equations (17) and (18) constitute the steady-state counterpart of equations (2) to (6), which express the transient behaviour of the Markov process. Actually, $\pi(j) = \lim_{n \to \infty} P(n, j)$, where n plays the role of the time parameter. From (18) we obtain:

$$\pi(j) = \frac{p^{j-1}(1-p)^{(j-1)j/2}}{\prod_{k=3}^{j+1}[1-(1-p)^k]}\pi(1), \qquad j \geqq 1 \tag{19}$$

and from the normalizing condition:

$$\pi(1)=\frac{1}{S_1} \tag{20}$$

where

$$S_1=\sum_{j=1}^{\infty}\frac{p^{j-1}(1-p)^{(j-1)j/2}}{\prod_{k=3}^{j+1}[1-(1-p)^k]} \tag{21}$$

The sum above converges, so the steady-state distribution exists and is given by (19) and (20).

The asymptotic rate α_1 can then be expressed as the steady-state probability that a node added to the graph initiates a new level:

$$\alpha_1=\sum_{k=1}^{\infty}\pi(k)[1-(1-p)^k] \tag{22}$$

By using equations (17) and (20) we finally obtain:

$$\alpha_1=\pi(1)[1-(1-p)^2]=\frac{1-(1-p)^2}{S_1} \tag{23}$$

Note that (22) and (23) can be also obtained from (7) and (8), respectively (by taking $n\to\infty$). But, we still need the steady-state equations (17) and (18) in order to obtain an expression for $\pi(1)$.

To prove that S_1 converges we argue as follows. Fix an $\varepsilon>0$ such that $p/(1-\varepsilon)<1$. Now $\exists j_0$ such that $(1-p)^j<\varepsilon$, $\forall j>j_0$:

$$\beta_j=\frac{p^{j-1}(1-p)^{(j-1)j/2}}{\prod_{k=3}^{j+1}[1-(1-p)^k]}<\frac{p^{j-1}}{\prod_{k=3}^{j_o}[1-(1-p)^k](1-\varepsilon)^{j+1-j_0}}$$

$$=\frac{1}{\prod_{k=3}^{j_0}[1-(1-p)^k](1-\varepsilon)^{2-j_0}}\left(\frac{p}{1-\varepsilon}\right)^{j-1}$$

Denoting by A the fraction in the last expression, we obtain:

$$S_1 < \sum_{j=1}^{j_0} \beta_j + A \left(\frac{p}{1-\varepsilon} \right)^{j_0 - 1} \frac{1}{1-p/(1-\varepsilon)} \tag{24}$$

Hence, if we denote by B_1 the right-hand side of the above inequality, we have $S_1 < B_1$, and finally

$$\alpha_1 > \frac{1-(1-p)^2}{B_1} \tag{25}$$

In fact, this bound can be as tight as we want by choosing sufficiently small ε and, accordingly, sufficiently large j_0 (observe that for $j_0 \to \infty$ the second term of the right-hand side of inequality (24) tends to zero).

4.2. Upper bound

Let us now consider the rate α_2 of increase of the length for a graph G_2 with n nodes as n becomes very large. Proceeding exactly as before, we consider the steady-state distribution $\pi(j)$ of the Markov process representing the number of nodes at the last level of the graph. $\pi(j)$ can be obtained from the equations:

$$\pi(1) = \sum_{k=1}^{\infty} \pi(k)[1-(1-p)^k] \tag{26}$$

$$\pi(j) = \pi(j-1)(1-p)^{j-1}, \qquad j > 1 \tag{27}$$

which constitute the steady-state counterpart of equations (11) and (12). We have:

$$\pi(j) = (1-p)^{(j-1)\,j/2}\,\pi(1), \qquad j \geqq 1 \tag{28}$$

and

$$\pi(1) = \frac{1}{S_2} \tag{29}$$

where

$$S_2 = \sum_{j=1}^{\infty} (1-p)^{(j-1)\,j/2} \tag{30}$$

Since $(1-p)^{(j-1)j/2} < (1-p)^{j-1}$, the sum S_2 also converges, so the steady-state distribution exists and is given by (28) and (29). Proceeding as before and using (26), we have for the rate α_2:

$$\alpha_2 = \pi(1) = \frac{1}{S_2} \tag{31}$$

If we set

$$B_2 = \sum_{j=1}^{j_0} (1-p)^{(j-1)j/2}$$

for fixed j_0, then $S_2 > B_2$ and

$$\alpha_2 < \frac{1}{B_2} \tag{32}$$

The above results can be put to use, after proving the following:

LEMMA 1: *Consider the two increasing sequences* u_n, v_n, $n=1, 2, \ldots$, *such that* $u_n \leqq v_n \ \forall n \in \mathbf{N}$ *and there exist the limits* $\lim_{n \to \infty} [u_{n+1} - u_n] = U$ *and* $\lim_{n \to \infty} [v_{n+1} - v_n] = V$. *Then* $U \leqq V$.

Proof: Suppose $V < U$. By definition of the limit, we have: $\forall \varepsilon > 0 \, \exists n_0 \in \mathbf{N}$ such that $\forall n \geqq n_0$ the following two inequalities hold:

$$U - \varepsilon < u_{n+1} - u_n < U + \varepsilon, \qquad n = n_0, n_0 + 1, \ldots \tag{33}$$

$$V - \varepsilon < v_{n+1} - v_n < V + \varepsilon, \qquad n = n_0, n_0 + 1, \ldots \tag{34}$$

Summing up inequalities (33) for values of n equal to $n_0, n_0+1, \ldots, n_0+k$ (k may be any positive integer) yields:

$$u_{n_0} + k\,U - k\,\varepsilon < u_{n_0+k} < u_{n_0} + k\,U + k\,\varepsilon \tag{35}$$

Similarly, from inequalities (34) we obtain:

$$v_{n_0} + k\,V - k\,\varepsilon < v_{n_0+k} < v_{n_0} + k\,V + k\,\varepsilon \tag{36}$$

If we choose ε such that $\varepsilon < (U-V)/2$, then there is an integer k_0 (actually there are infinitely many) such that $k_0 > (v_{n_0} - u_{n_0})/(U - V - 2\varepsilon)$. The last inequality can be written:

$$v_{n_0} + k_0\,V + k_0\,\varepsilon < u_{n_0} + k_0\,U - k_0\,\varepsilon \tag{37}$$

Inequalities (35) to (37) yield that

$$v_{n_0+k_0} < u_{n_0+k_0}$$

which is a contradiction. ■

As Lemma 1 applies to both

$$(u_n, v_n) = (\overline{L_1(n)}, \overline{L(n)}) \quad \text{and} \quad (u_n, v_n) = (\overline{L(n)}, \overline{L_2(n)}),$$

we are in a position to state the following:

THEOREM 1: *For all* $p \in (0, 1]$, $(1-(1-p)^2)/S_1 \leqq \alpha \leqq 1/S_2$, *where* S_1 *and* S_2 *are given by* (21) *and* (30) *respectively.*

It is not straightforward for the quantities α_1 and α_2 to be exactly computed. However, they can be bounded as shown by (25) and (32), so that finally

$$\frac{1-(1-p)^2}{B_1} < \alpha < \frac{1}{B_2}$$

where B_1 and B_2 can be exactly computed for any fixed value of ε.

4.3. Limits

It is obvious that the rate α tends to one as p tends to one and to zero as p tends to zero. In order to obtain a more detailed picture of the behaviour of α for small values of p, we study the quantity α/p; the latter turns out to converge to a finite number as $p \to 0$, indicating that α tends to zero with the same speed as p. Bounds and limiting values for this quantity are also obtained in [5], which can be compared with the ones obtained here, though derived in a different context and following a different approach. In particular, in [5] the inverse of α/p is refered to as the "efficiency" of a queueing system, in which the arrival process corresponds to forming a random dag like the one studied here.

Let us consider the limits of the bounds for $p \to 0$. We have for the lower bound:

$$\lim_{p \to 0} \frac{\alpha_1}{p} = \lim_{p \to 0} \frac{1-(1-p)^2}{S_1 p}$$

$$= \lim_{p \to 0} \lim_{j_0 \to \infty} \frac{1-(1-p)^2}{p \sum_{j=1}^{j_0} p^{j-1}(1-p)^{(j-1)j/2} / \prod_{k=3}^{j+1} [1-(1-p)^k]}$$

$$= \lim_{j_0 \to \infty} \lim_{p \to 0} \frac{1-(1-p)^2}{p \sum_{j=1}^{j_0} p^{j-1}(1-p)^{(j-1)j/2} / \prod_{k=3}^{j+1} [1-(1-p)^k]}$$

$$= \lim_{j_0 \to \infty} \lim_{p \to 0} \frac{2-p}{1+\sum_{j=2}^{j_0} p^{j-1}[1-p(j-1)j/2+o(p)]/p^{j-1}[(j+1)!/2+d(j)p+o(p)]}$$

$$= \lim_{j_0 \to \infty} \frac{1}{\sum_{j=1}^{j_0} 1/(j+1)!}$$

where $d(j)$ is a function of j and $o(p)$ represents any quantity that approaches zero faster than p as $p \to 0$; that is $o(p)/p \to 0$ as $p \to 0$. We finally obtain:

$$\lim_{p \to 0} \frac{\alpha_1}{p} = \frac{1}{e-2} \tag{38}$$

In what concerns the upper bound we can show that the quantity α_2/p diverges for $p \to 0$. We will actually study the limit of the quantity $F = \alpha_2/p - 1$. We have:

$$\lim_{p \to 0} F = \lim_{p \to 0} \frac{1/S_2 - p}{p} = \lim_{p \to 0} \frac{1/p - S_2}{S_2}$$

$$= \lim_{p \to 0} \frac{\sum_{j=1}^{\infty} [(1-p)^{j-1} - (1-p)^{(j-1)j/2}]}{\sum_{j=1}^{\infty} (1-p)^{(j-1)j/2}}$$

For the numerator of the above expression we have: $\forall j_0 \exists \delta$ such that $\forall p < \delta$ the following inequality holds:

$$\sum_{j=1}^{\infty} (1-p)^{j-1} [1-(1-p)^{(j-1)(j-2)/2}] \geqq \frac{p}{2} \sum_{j=1}^{j_0} (j-1)(j-2)(1-p)^{j-1} \quad (39)$$

Since

$$\lim_{j_0 \to \infty} \frac{p}{2} \sum_{j=1}^{j_0} (j-1)(j-2)(1-p)^{j-1} = \frac{(1-p)^2}{p^2}$$

we can write: $\forall \varepsilon > 0 \exists \delta$ such that $\forall p < \delta$

$$\sum_{j=1}^{\infty} (1-p)^{j-1} [1-(1-p)^{(j-1)(j-2)/2}] \geqq \frac{(1-p)^2}{p^2} - \varepsilon$$

Since the denominator of F is bounded from above by $1/p$ we obtain: $\forall \varepsilon > 0 \exists \delta$ such that $\forall p < \delta$

$$F \geqq \frac{(1-p)^2}{p} - p\varepsilon \xrightarrow{p \to 0} \infty$$

or finally

$$\lim_{p \to 0} \frac{\alpha_2}{p} = \infty \quad (40)$$

The above results are illustrated succinctly in Figure 6. The variation of α_1/p and α_2/p as function of p has been plotted in bold line. In dashed line we show the variation of the corresponding bounds obtained in [5] for the asymptotic behaviour of the same graph model. For small values of p, our upper bound diverges away from α/p; however, from [5] we have a good approximation of α/p for small values of p, since their upper bound is shown to converge to the actual limit of α/p as $p \to 0$. For moderate and large values of p our upper bound significantly improves that of [5], while our lower bound is overall slightly better than theirs.

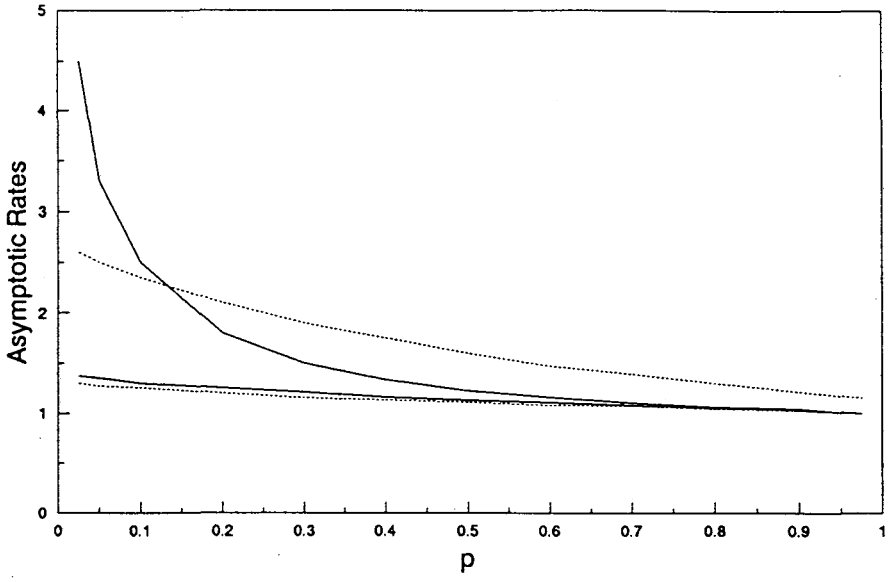

Figure 6. – Asymptotic rates for the bounds (dashed lines display previously published results).

5. AVERAGE WIDTH

We will now be interested in the asymptotic behaviour of the average width of each level of the graph, *i.e.*, the average number of nodes at that level. This is a measure of major importance, since it represents an evaluation of what is known as the *level of parallelism* or *level of concurrency* of the application.

We start by deriving the average number of nodes at the first level. Proceeding exactly as we did in Section 3, we define a sequence of indicator random variables Y_k as follows:

$$Y_k = \begin{cases} 1 & \text{if the } k\text{-th node is put at the first level} \\ 0 & \text{otherwise} \end{cases}$$

Then we have:

$$\overline{W_1(n)} = \sum_{k=1}^{n} \overline{Y_k} \tag{41}$$

where

$$\bar{Y}_k = \Pr[Y_k = 1] = (1-p)^{k-1} \tag{42}$$

since a node at the first level has outdegree 0. From (41), (42) we readily obtain:

$$\overline{W_1(n)} = \sum_{k=1}^{n} (1-p)^{k-1} = \frac{1-(1-p)^n}{p} \tag{43}$$

which tends to $\overline{W_1(\infty)} = 1/p$ as $n \to \infty$. The above quantity is derived in [2] as an upper bound for the expected level of concurrency in an analogous model, in which at any time there are n transactions present in the system (each transaction that completes execution is immediately replaced with a new transaction). It is also used in [3] to obtain a maximum likelihood estimator for the probability p from the number of nodes at the first level.

We shall now compute the average number of nodes at the second level of our random graph. Let us define the indicator random variables Y_k and Y'_k as follows:

$$Y_k = \begin{cases} 1 & \text{if the } k\text{-th node is put at the second level} \\ 0 & \text{otherwise} \end{cases}$$

$$Y'_k = \begin{cases} 1 & \text{if the } k\text{-th node added to levels 2, 3, . . .} \\ & (i.e., \text{ all but the first}) \\ & \text{is put at the second level} \\ 0 & \text{otherwise} \end{cases}$$

Recall that $\overline{W_2(n)}$ denotes the average number of nodes at the second level for a random dag with n nodes and let $\overline{W'_2(n)}$ denote the average number of nodes at the second level of a random dag when there are n nodes at levels 2, 3, . . . Arguing in the same way as before we have:

$$\overline{W_2(n)} = \sum_{k=1}^{n} \bar{Y}_k = \sum_{k=1}^{n} \Pr[Y_k = 1] \tag{44}$$

and

$$\overline{W'_2(n)} = \sum_{k=1}^{n} \bar{Y}'_k = \sum_{k=1}^{n} \Pr[Y'_k = 1] \tag{45}$$

The computation of $\overline{W'_2(n)}$ is straightforward, since $\Pr[Y'_k = 1] = (1-p)^{k-1}$, yielding exactly the expression obtained for $\overline{W_1(n)}$, namely

$$\overline{W'_2(n)} = [1-(1-p)^n]/p,$$

which tends to $\overline{W'_2(\infty)} = 1/p$ as $n \to \infty$. In the following, we will prove that $\overline{W_2(\infty)} = \overline{W'_2(\infty)}$. We first define the indicator random variable Z_k^n as follows:

$$Z_k^n = \begin{cases} 1 & \text{if the } n\text{-th node added to the dag} \\ & \text{is at the same time the } k\text{-th node} \\ & \text{added to levels 2, 3, } \ldots \\ 0 & \text{otherwise} \end{cases}$$

We then have:

$$\begin{aligned} \overline{W'_2(\infty)} &= \sum_{k=1}^{\infty} \Pr[Y'_k = 1] \\ &= \sum_{k=1}^{\infty} \sum_{n=k+1}^{\infty} \Pr[Y_n = 1] \Pr[Z_k^n = 1] \\ &= \sum_{n=1}^{\infty} \Pr[Y_n = 1] \sum_{k=1}^{n-1} \Pr[Z_k^n = 1] \\ &= \sum_{n=1}^{\infty} \Pr[Y_n = 1] = \overline{W_2(\infty)} \end{aligned}$$

The above reasoning holds for any given level of the graph. Hence, we can state the following:

THEOREM 2: *Given a probability p and an integer k, the average number of nodes at the k-th level of the random dag tends to* $1/p$ *as the number of nodes in the dag tends to infinity.*

This result suggests that *the expected level of parallelism is inversely proportional to the probability of interdependence.*

6. CONCLUSION

We have investigated in this paper the behaviour of a random task graph, which can serve as a general parallel processing model. In particular, we derived upper and lower bounds on the mean processing time of the graph and related asymptotic parameters. The results obtained constitute useful

extensions to previously published results on the same problem. We also examined the asymptotic behaviour of the average width of the graph. It would be interesting to investigate less general task graph models in what concerns the representation of precedence constraints, as well as the case of random task execution times. This type of model can also be applied in the performance evaluation of queueing systems with particular constraints concerning the servicing process.

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous referee for his/her careful and useful remarks.

REFERENCES

1. M. H. Albert and A. M. Frieze, Random Graph Orders, *Order*, Vol. 6, 1989, pp. 19-30.
2. P. Franaszek and J. T. Robinson, Limitations of Concurrency in Transaction Processing, *ACM Transactions on Database Systems, 10*, 1, 1985, pp. 1-28.
3. E. Gelenbe, R. D. Nelson, T. K. Philips and A. Tantawi, An Approximation of the Processing Time for a Random Graph Model of Parallel Computation, in: *Proceedings*, 1986 *Fall Joint Computer Conference*, Dallas, Texas, ACM/IEEE Computer Society, 1986, pp. 691-697.
4. C. H. Papadimitriou, *The Theory of Database Concurrency Control*, Computer Science Press, 1986.
5. J. N. Tsitsiklis, C. H. Papadimitriou and P. Humblet, The Performance of a Precedence-Based Queueing Discipline, *Journal of the ACM* 33, *3*, 1986, pp. 593-602.