Analysis of a Randomized Dynamic Timetable Handshake Algorithm**

A. El Hibaoui [1], Y. Métivier [2], J.M. Robson [2], N. Saheb-Djahromi [2], A. Zemmari [2]

[1] FST de Tetouan, Université Abdelmalek Essaâdi, B.P. 2121, Mhannech II, Tetouan, Maroc
hibaoui@uae.ma

[2] LaBRI, Université de Bordeaux, 351 cours de la Libération, 33405 Talence, France
{metivier,robson,saheb,zemmari}@labri.fr

Abstract

In this paper we introduce and study a handshake algorithm based on random delays. This algorithm can also be considered as a probabilistic distributed algorithm to find a maximal matching. These delays are generated uniformly at random in the real interval $[0, 1]$ *and the handshakes take place between neighbour processors if both processors are free at the generated time. We study the distribution of the handshake number, and show that this algorithm is substantially more efficient than previous ones known to us, in that the expected number of handshakes per round is larger.*
Keywords: Distributed Algorithm, Handshake Algorithm, Matching, Random Graphs, Probabilistic Analysis.

1 Introduction

1.1 The Distributed Network Model

A distributed system (P, C) consists of a collection P of processes and a communication subsystem C (our definitions follow [13] (p. 45)). It is described by a simple undirected graph $G = (V, E)$, where the vertices represent the processes and the edges represent the bidirectional channels. Processes communicate by message passing and each process knows by which channel it receives a message or sends a message: an edge between two processes (or vertices v_1 and v_2) represents a channel connecting a port i of v_1 to a port j of v_2. Let δ be the port numbering function, we assume that for each vertex u and each adjacent vertex v, $\delta_u(v)$ is a unique integer belonging to $[1, deg(u)]$. Finally, the communication subsystem is described by $C = (V, E, \delta)$. The network is anonymous: unique identities are not available to distinguish processes. Furthermore we do not assume that the size (or an upper bound on the size) of the graph is known.

Our algorithm is synchronous and timer-based: the processes have access to a physical clock device and as in ([13] page 87):

1. we assume that the timer is a global continuous time to which processes have access: a real-valued variable whose value continuously increases in time,

2. we make the global time assumption, that is each event is said to take place at a certain time and each event itself is assumed to have duration 0.

We assume that communications take no time, thus the delay between the decision of a vertex and the notification to its neighbours is equal to 0.

**This work was supported by grant No ANR-06-SETI-015-03 awarded by Agence Nationale de la Recherche

1.2 The Problem

We are concerned with the establishment of communications through synchronisation signals or equivalently with a local distributed scheduler which finds matching pairs of processes in order to establish communications. This implementation may be done through a handshake as described by Reif and Spirakis in [12] (Appendix p. 93): *Suppose that each process has a special resource called channel which can be in one of two states* **open, closed**. *A handshake of two processes p, q in time t is a combination of process states at time t so that both channels of p and q are open at the same time.* Thus we are concerned by local signals so that each process indicates to at most one neighbour its readiness to send or receive data.

Let us recall the definition of a matching. A *matching* in $G = (V, E)$ is a subset of edges $\mathcal{M}$ of E, in which no two edges are adjacent (or, alternatively no vertex is adjacent to two edges in the matching). A matching is *maximal* if no edge can be added without violating the constraint. Clearly a matching of maximum cardinality is a maximal matching; the converse is not true. The maximum cardinality is called the *matching number*.

1.3 The Algorithm

Our solution is symmetric and fully distributed. We consider a variant of the communication protocol described in [5] (Section 3.1 p. 377-378). The same kind of control messages to negotiate communications are used in [2].

Procedure HS

Each process p executes forever the following steps during each unit time (round):

The process p generates $t_p(q)$: a random time chosen uniformly in the real interval $[0, 1]$ for each neighbouring process q;
p waits until one of the following three events occurs

- *p receives 1 from a neighbour r; p sends 0 to all its other neighbours.*
  (There is a handshake between p and r. *)*
- *$t = t_p(q)$ and p has not received any signal from q; p sends 1 to q and 0 to all its other neighbours.*
  (There is a handshake between p and q. *)*
- *$t = 1$.*
  (The round terminates without p taking part in a handshake. *)*

Remark 1.1. *Since time is continuous, with probability 1 one and only one of these events will occur.*

1.4 Results and Previous Works

The aim of the rest of this paper is the analysis of Procedure HS.

We consider first the average of the *handshake number*, that is the average size of the matching obtained. We give a recursive formula which could be used to compute this number in general and give closed forms or asymptotic values for some interesting families of graphs.

For random graphs with average degree a constant c, as the size tends to infinity, we show that the probability that a handshake takes place on a given edge tends to $1/(1 + c)$.

Then we consider the *efficiency ratio*, that is the ratio between the average handshake number and the matching number. For general graphs this ratio is at least $1/2$ and this bound cannot be increased.

We compare our algorithm with a previous one proposed by [12] (p. 225) and analysed in [10]. We prove that the probability of a handshake over an edge in our algorithm is greater than or equal to the probability of the same event in the algorithm in [12]. This implies that our matching size dominates that of [12] in expectation. For some graphs such as complete graphs, the difference is very considerable.

A similar algorithm has been introduced by [6] to handle the matching number in graphs. A more similar approach has been developed by [3]. We consider a distributed algorithm in which each processor has only local knowledge of the graph structure whereas in [3], the authors treat a centralised algorithm with complete knowledge of the graph. Our investigations, diverge in their goal and analysis. General considerations about randomised distributed algorithms may be found in [13] and some techniques used in the design and for the analysis of randomised algorithms are presented in [8] and in [7].

Our paper is organised as follows. Section 2 concerns the handshake number, the expected handshake number, the impact of a link insertion and of an extension and finally the expected handshake number in random graphs. Section 3 studies the efficiency ratio and gives the bounds for general graphs.

2 Handshake Number

Throughout this paper $G = (V, E)$ is a simple undirected graph. We refer to $|V|$ as the *size* (or *order*) of G. To avoid the triviality, we suppose that $|E| > 0$. The algorithm starts with the generation of $2|E|$ independent uniform real random variables (r.v.) in the interval $[0, 1]$: two r.v. for each edge. We can assume that all $(2|E|)!$ orderings on the set of these real numbers have the same probability. This is the main hypothesis in the sequel. In fact, for this assumption to be valid, one only has to postulate that, for each edge $e = \{u, v\}$ in G, the algorithm generates two continuous r.v. $X_e(u)$ and $X_e(v)$, corresponding to $t_u(v)$ and $t_v(u)$ in the algorithm, supposing that these r.v. associated with edges are *all* independent and identically distributed. The first handshake takes place on the edge $e = \{u, v\}$, if one of two associated r.v., $X_e(u)$ or $X_e(v)$, is minimal in the whole graph. Thus, for the first handshake on G, all edges have the same chance $1/|E|$ to be chosen.

The assignment of the first handshake to u and v on $\{u, v\}$, involves that these vertices are removed with their incident edges and, then, the process continues on the new graph (preserving the random generations for the remaining edges) until no edges remain in the set of edges. The *handshake number* in a round is simply the total number of edges to which a handshake is assigned. Let $R(G)$ be this number whenever our algorithm is applied to the graph G. This is an integer valued r.v. It takes the value 0 with probability 1 if $E = \emptyset$, the value 1 with probability 1 if $|E| = 1$. In general it takes a value ranging over the set of all cardinalities of maximal (in the inclusion sense) matchings in G, with some probability.

It seems useful to note the following fact which is easy to prove:

Fact. Let the first handshake be assigned to $e = \{u, v\}$. Let $G_e = (V', E')$ be the graph obtained by dropping vertices u and v and all incident edges from G. Then, all $(2|E'|)!$ orderings of generated r.v. (two per edge) in G_e, conditioned by the minimality $X_e(u)$ or $X_e(v)$, have the same probability $1/(2|E'|)!$. This means that keeping the $2|E|$ r.v. for the edges of G_e is equivalent to starting the algorithm on G_e with new random generation.

This fact allows us to compute recursively the probability distribution of $R(G)$ as follows. Let $G = (V, E)$. If $E = \emptyset$, we have $R(G) = 0$. We can easily prove the following:

Proposition 2.1. *Let $G = (V, E)$ with $n = |V| \geq 2$ and $m = |E| \geq 1$. We have then*

$$Pr(R(G) = k) = \frac{1}{m} \sum_{e \in E} Pr(R(G_e) = k - 1), \quad \forall k \geq 1.$$

It should be noted that, $m \geq 1$, this algorithm allows at least 1 handshake with probability 1 in a round; this is *not* the case in [10].

- In the case of complete graphs of size $|V| = n$, we have with probability 1, $R(G) = \lfloor n/2 \rfloor$.
- In the case of star graphs of size $n \geq 2$, we have with probability 1, $R(G) = 1$.
- Let G be the graph of Fig. 1. A simple computation yields $Pr(R(G) = 1) = 1/5$ and $Pr(R(G) = 2) = 4/5$.

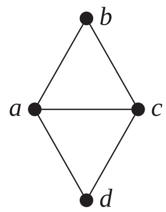

Figure 1: A simple graph

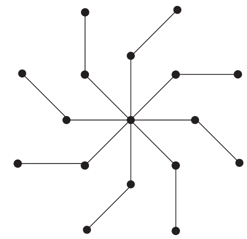

Figure 2: A large-star graph

- In the case of large-star graphs of order $n \geq 3$, Fig. 2, we have with probability 1, $R(G) = (n-1)/2$.

Now let $M_e(G)$ be 1 if there is a handshake on e and 0 otherwise, for the edge $e = \{u, v\}$ in the graph $G = (V, E)$ (in a round). The r.v. $R(G)$ can be written as the sum $\sum_{e \in E} M_e(G)$. Moreover it is easy to see that:

Proposition 2.2. *Let $F(e)$ be the set of edges of G not incident on e. The r.v. $M_e(G)$ can inductively be characterised by*

$$Pr(M_e(G) = 1) = \frac{1}{m} + \frac{1}{m} \sum_{f \in F(e)} Pr(M_e(G_f) = 1).$$

The first term is the probability that the first handshake takes place on e and the second term is that it takes place leter on the residual graph.

In [10], the authors propose and analyse a randomised algorithm based on the uniform selection of a neighbour vertex. Whenever two neighbour vertices choose each other, there will be handshake between the two vertices. The probability of a handshake on the edge $e = \{u, v\}$ in a round of the MSZ-algorithm is $1/(d(u)d(v))$, where $d(u)$ and $d(v)$ are degree of u and degree of v respectively. The following proposition proves that the present algorithm is at least as efficient as the MSZ-algorithm in that the probability of a handshake over a given edge by the new algorithm is greater than or equal to the probability of the same event by the previous algorithm.

Proposition 2.3. *Let $e = \{u, v\}$ be an edge in $G = (V, E)$. Then*

$$Pr(M_e(G) = 1) \geq \frac{1}{d(u)d(v)}.$$

*Proof:*By induction on $m = |E|$. The proposition clearly holds for $m = 1$ and $m = 2$. Let it be true for graphs having less than m edges and prove it for G with m edges. According to the previous proposition, the investigated probability can be written as:

$$Pr(M_e(G) = 1) = \frac{1}{m} + \frac{1}{m} \sum_{f \in F(e)} Pr(M_e(G_f) = 1).$$

In the summation there are $m - d(u) - d(v) + 1$ terms. If this number is zero, then

$$\frac{1}{m} = \frac{1}{d(u) + d(v) - 1} \geq \frac{1}{d(u)d(v)}$$

and the proposition holds. Otherwise, according to the inductive hypothesis, each term of the summation is greater than or equal to $1/(d(u)d(v))$ and, hence:

$$Pr(M_e(G) = 1) \geq \frac{1}{m} + \frac{1}{m}(m - d(u) - d(v) + 1)\frac{1}{d(u)d(v)}.$$

And it is easily shown that the last expression is greater than or equal to $1/(d(u)d(v))$.

■

In fact in many cases, the probability of a handshake over an edge in this new algorithm, is substantially larger than according to the MSZ. Indeed, it is possible to construct simple examples in which $d(u)d(v)$ is much greater than m and, hence, the coarse lower bound $1/m$ for $Pr(M_e(G) = 1)$ exceeds the value $1/(d(u)d(v))$.

2.1 Expected Handshake Number

In a randomised distributed algorithm, an important parameter which measures the efficiency is the expected number of events which can take place in a round, see [10]. We shall study this parameter for our new algorithm. Let $\overline{R}(G)$ be the mathematical expectation of $R(G)$. Let $C(G)$ denote the set of all maximal matchings over G. We set:

$$\overline{R}(G) = \mathbb{E}(R(G)) = \sum_k kPr(R(G) = k) = \sum_{e \in E} \mathbb{E}(M_e(G)).$$

($\mathbb{E}$ denotes the mathematical expectation.)

We derive easily from Proposition 2.3:

Proposition 2.4. *For all graphs, the expected handshake number in the MSZ-algorithm does not exceed that in the new algorithm.*

Let us consider some particular instances.

1. Complete graphs. For the complete graph K_n of size n, we have $\overline{R}(K_n) = \lfloor n/2 \rfloor$. Comparing with the expected handshake number in a round of MSZ-algorithm, which is $n/(2(n-1))$, for $K_n, n \geq 2$. We observe that the new algorithm is substantially more efficient on complete graphs.

2. Star graphs. If G is a star graph of size $n \geq 2$, we have $\overline{R}(G) = 1$.

3. Large-star graphs. For the large-star graph (Fig. 2) of size $n \geq 4$, we have $\overline{R}(G) = n/2$.

4. Chain graphs. As the last example, we consider chain graphs and derive the asymptotic expected handshake number on them. Let $G_n = (V_n, E_n)$ be the chain graph such that $|V_n| = n$, and let $R_n = \overline{R}(G_n)$ denote the expected number of handshakes in G_n. We have the following proposition:

Lemma 2.5. *We have:*

1. *The expected handshake number in chain graphs satisfies the recurrence:*

$$\forall n \geq 2, \; R_n = 1 + \frac{2}{n-1} \sum_{i=0}^{n-2} R_i \quad \text{and} \quad R_0 = R_1 = 0.$$

2. *If we denote by $R(z)$ the ordinary generating function defined by $R(z) = \sum_{n \geq 0} R_n z^n$, then $R(z) = z(1 - e^{-2z})/(2(1-z)^2)$.*

Proof: The first clause is trivial. To prove the second one, we derive from Clause *1* the equality:

$$R_{n+1} = \frac{n-1}{n} R_n + \frac{2}{n} R_{n-1} + \frac{1}{n}.$$

Now, an easy transformation of the above equation into the corresponding ordinary generating functions (OGF) yields the following differential equation:

$$z(1-z)R'(z) - (1 - z + z^2)R(z) = \frac{z^2}{1-z},$$

which admits the solution $R(z) = z(1 - e^{-2z})/(2(1-z)^2)$, ending the proof.

■

Remark 2.6. *This generating function is the generating function corresponding to the "Seating Arrangement Problem" proposed and resolved by Flajolet in [4].*

The OGF presented in the previous proposition can be used to derive an explicit expression for the term R_n. Indeed, a straightforward computation yields:

$$R_n = \frac{1}{2}\sum_{i=1}^{n-1} \frac{(-1)^{n+1}2^i}{i!}(n-i).$$

Moreover, we have:

Proposition 2.7. *The asymptotic value of R_n is given by:*

$$R_n \sim (1 - e^{-2})\frac{n}{2} \approx 0.432332n \qquad (n \to \infty).$$

Proof: To compute the asymptotic value of R_n, we use Bender's method presented in [4]. $R(z)$ can be written as a product of two generating functions $R(z) = a(z)b(z)$ with $a(z) = 1-e^{-2z}$ and $b(z) = z/(2(1-z)^2)$.

The generating function $a(z)$ is convergent everywhere and $b(z)$ has $\beta = 1$ as the convergence radius. Since $b(z) = \sum_{n\geq 0}(n/2)z^n$ and $a(z) = \sum_{n\geq 1} -((-2)^n/n!)z^n$, and $b(z)$ satisfies $(b_{n-1}/b_n) \to \beta = 1$ as $n \to \infty$, the coefficients of the product $R(z) = a(z)b(z)$ are given by:

$$[z^n]R(z) = R_n \sim a(1)b_n = (1 - e^{-2})\frac{n}{2} \approx 0.432332n \qquad (n \to \infty).$$

■

2.2 Impact of a Link Insertion

Consider the expected handshake number for the graphs of Fig. 3. G_2 is obtained by the elimination and G_3 by the insertion of an edge in G_1. A simple computation yields $\overline{R}(G_1) = 9/5$ and $\overline{R}(G_2) = \overline{R}(G_3) = 2$.

Figure 3: Three connected graphs having the same set of vertices

These examples show that the impact of an edge insertion in a connected graph does not have a monotonic effect on the expected handshake number.

2.3 Impact of an Extension

In the previous section we showed that edge insertions in a graph may have a negative effect on the expected handshake number. A natural question arising is what will be the impact of adding a new vertex and new links between the initial vertices and the new one. Although the effect seems to be positive, the proof is not obvious. The following proposition shows that the impact is non-negative and is at most 1 for the insertion of one vertex.

Proposition 2.8. *Let $G = (V, E)$ be a graph and G' a graph obtained from G by adding a new vertex x and zero or more edges between x and vertices of G. We have:*

$$\overline{R}(G) \leq \overline{R}(G') \leq \overline{R}(G) + 1.$$

Proof: By induction over $n = |V|$. For $n = 0$ the proposition obviously holds. Let it hold for graphs of size less than n and prove it for G of size n. Let e be the first edge on which a handshake takes place in G'. We have two cases depending on whether or not x is incident to e. To prove the proposition, it suffices to show the above inequalities on the expected values, conditioned by each of these events which we denote $\mathcal{E}_1$ and $\mathcal{E}_2$, hold.

1. **Event $\mathcal{E}_1$.** Let $e = \{x, v\}$ for some $v \in V$. We have on the one hand $\overline{R}(G'/\mathcal{E}_1) = \overline{R}(G'_e) + 1$ and, by the induction hypothesis, $\overline{R}(G'_e) \leq \overline{R}(G) \leq \overline{R}(G'_e) + 1$ on the other hand. This yields $\overline{R}(G) \leq \overline{R}(G'/\mathcal{E}_1) \leq \overline{R}(G) + 1$.

2. **Event $\mathcal{E}_2$.** Let e be in G. Now, by the induction hypothesis $\overline{R}(G_e) \leq \overline{R}(G'_e) \leq \overline{R}(G_e)+1$, conditioned by the choice of any $e \in E$. But $\overline{R}(G)$ is the average value of $\overline{R}(G_e) + 1$ and $\overline{R}(G'/\mathcal{E}_2)$ is the average of $\overline{R}(G'_e) + 1$, e ranging over E. Hence $\overline{R}(G) \leq \overline{R}(G'/\mathcal{E}_2) \leq \overline{R}(G) + 1$.

The proposition follows.

■

2.4 Asymptotic Expected Handshake Number in Random Graphs

We consider here random graphs of average degree $c > 0$. We first compute the asymptotic value for the probability of a handshake over a given edge of the graph. Let G be a random graph of size n whose average degree is c. In this model, the probability that two distinct vertices are linked by an edge is $c/(n-1)$, so that the average degree is c. So the considered model is the constant probability model (see [1]), in which the average degree remains bounded. Let $\{a, b\}$ be an existing edge in the random graph G. The main subject of the study here is to derive an asymptotic expression of the probability $P(n, c)$ of a handshake over $\{a, b\}$. First we consider a simplified model, called in the sequel the *tree model* in which this probability for the edge $\{a, b\}$ is asymptotically the same as in the initial random graph. In the tree model, the graph consists of the edge $\{a, b\}$ and two disjoint random trees rooted at a and b in each of which a vertex v has out-degree chosen so as to have average value $c + o(1)$ by adding an edge to each of $n-2$ new vertices, each independently with probability $c/(n-1)$, the new vertices being different for each v. This model differs from the initial random graph only in that all the potential new vertices are *distinct*; in the graph model the potential new vertices are all the graph vertices other than v and not already connected to v. The process is *only* carried out for vertices v such that edges added beyond v could influence the outcome of the algorithm on $\{a, b\}$. To simplify the main result on $P(n, c)$, we first show some lemmas.

Lemma 2.9. *With probability tending to 1 the trees constructed above are finite.*

Proof: The probability of an edge $e = \{a, b\}$ taking part in the matching depends on the smaller of $X_e(a)$ and $X_e(b)$ a random variable with a non-uniform distribution. To simplify the discussion we define a random variable z_e as:

$$Pr[\min(X_e(a), X_e(b)) < \min(Y, Z)],$$

where Y and Z are independent random variables with the same distribution as the X_e. z_e is uniformly distributed on $[0, 1]$ and between two edges e and f, e will be chosen if $z_e > z_f$. Suppose that v is at the end of a chain $v_0 = b, v_1, v_2, ..., v_h = v$; then the edge $\{v_{h-1}, v_h\}$ can influence the outcome at $\{a, b\}$ only if it is chosen thereby inhibiting $\{v_{h-2}, v_{h-1}\}$ from being chosen and otherwise $\{v_{h-2}, v_{h-1}\}$ would have been chosen inhibiting $\{v_{h-3}, v_{h-2}\}$ etc. This can only happen if $z_{\{v_{h-1}, v_h\}} > z_{\{v_{h-2}, v_{h-1}\}} \cdots > z_{\{v_0, v_1\}}$ which has probability of $1/h!$. But the expected number of vertices v at distance h from a or b is $2c^h$ and so the probability that any of them influences $\{a, b\}$ is at most $2c^h/h!$ which tends to 0. Thus choosing h sufficiently large but growing slowly with n, say $h = \ln \ln n$, and then truncating the tree at that level gives a probability $1 - o(1)$ of having the same probability of choosing $\{a, b\}$ as in the non-truncated case.

■

The next lemma gives the limit of the probability that a given edge is not inhibited from being chosen.

Lemma 2.10. *The probability P that the edge $\{a, b\}$ is not inhibited in the tree model tends to $1/(1+c)$ as $n \to \infty$.*

Proof: We first consider graphs consisting of the edge $\{a, b\}$ and just one random tree constructed as described above starting from b. In such a tree truncated at height h, the edge $\{a, b\}$ has probability p_h of not being inhibited and for any x ($0 \le x \le 1$) a conditional probability $f_h(x)$ that z is greater than x given that $\{a, b\}$ is not inhibited. Each edge $\{b, c_i\}$ leaving b has probability p_{h-1} of not being inhibited by an edge beyond it in the tree and probability $f_{h-1}(x)$ of having its z greater than x conditioned on this event. Hence if $z_{\{a,b\}} = x$, the probability of $\{a, b\}$ not being inhibited is $\approx \prod_{i=1}^{n-2}(1 - (cp_{h-1}f_{h-1}(x))/(n-1))$ where we write $\approx$ for equality up to $o(1)$. This gives us

$$p_h \approx \int_0^1 e^{-cp_{h-1}f_{h-1}(x)}dx$$

and

$$f_h(y) \approx \frac{\int_y^1 e^{-cp_{h-1}f_{h-1}(x)}dx}{p_h}.$$

Taking the limit as $h \to \infty$ gives

$$p = \int_0^1 e^{-cpf(x)}dx$$

$$f(y) = \frac{\int_y^1 e^{-cpf(x)}dx}{p}$$

f satisfies the obvious boundary conditions $f(0) = 1$ and $f(1) = 0$.

These equations have the solution $f(x) = \ln(1 + c - cx)/\ln(1 + c)$ and $p = \ln(1 + c)/c$ so that $e^{-cpf(x)} = 1/(1 + c - cx)$.

Now, at last, we can consider the probability P that $\{a, b\}$ is not inhibited. Taking into account the possibility of it being inhibited by an edge incident on a or b, we have $P = \int_0^1 e^{-2cpf(x)}dx = \int_0^1 (1 + c - cx)^{-2} = 1/(1 + c)$.

■

We are now ready to state the main result on the random graph:

Theorem 2.11. *The probability $P(n, c)$ that a handshake takes place on an existing edge $\{a, b\}$ in a random graph G, with average degree c and of size n, tends to $1/(1+c)$, as $n \to \infty$.*

Proof: By virtue of the previous lemmas, it suffices to show that the limiting probabilities are the same for the random graph and the tree. In the ball of radius h around $\{a, b\}$ these probabilities are the same in both models unless two distinct nodes of the trees rooted at a and b turn out to be the same graph vertex. Since the expected number of pairs of tree nodes is $O(1)^h$, which is $(\ln n)^{O(1)}$, the probability of this happening is $(\ln n)^{O(1)}/n = o(1)$, giving probability in the graph model of $P + o(1)$. This establishes the theorem.

■

Let $\mathcal{R}(n, c)$ be the expected handshake number in the random graph considered here. We have:

Proposition 2.12. *$\mathcal{R}(n, c)$ has the asymptotic value $(n/2)(1 - 1/(1 + c))$, as n tends to ∞.*

Proof: Straightforward, by the above theorem and the fact that the average number of edges incident to a vertex is c.

■

We now consider the other random graph model with a determined number $m = cn/2$ of edges, see the model $\mathcal{G}(n, m)$ in [1]. The following proposition shows that in the matter of handshakes on edges, the two models of random graphs are asymptotically equivalent.

Proposition 2.13. *In the random graph of model $\mathcal{G}(n, m)$ with $m = cn/2$, the probability $P(n, c)$ that a given vertex is not inhibited tends to $1/(1+c)$.*

Proof: We show that the new model of random graph (using the same source of random numbers in two cases) will realize the same ball of radius h around $\{a, b\}$ with probability $1 - o(1)$, so that the limiting probabilities in two models are identical. The difference between the balls of radius h in the two cases is that whereas the constant probability model always adds a candidate edge with probability $c/(n-1)$, the fixed number of edges model adds a candidate with probability m'/p' where m' is the remaining number of edges required and p' the remaining number of vertex pairs to be considered. First we note that with probability $1 - o(1)$, the constant probability model adds a number of vertices at most twice the expected number, and so $(\ln n)^{O(1)}$. Now we can ignore the cases with more than $(\ln n)^{O(1)}$ vertices in the trees and deduce that each potential edge is added with probabilities $c/(n-1)$ or between $cn/(n^2 - n(\ln n)^{O(1)})$ and $c(n - n(\ln n)^{O(1)}/n^2)$ according to the model. So the difference between the probabilities in the two models for each potential edge considered is $(\ln n)^{O(1)}/n^2$ and the sum of these differences over all these potential edges is $(\ln n)^{O(1)}/n = o(1)$. Hence the probability that the two random processes give different radius h balls is $o(1)$ so that the same limiting probabilities of $1/(c+1)$ holds for the fixed edge number model as well.

■

3 Efficiency of the Algorithm

In the previous section we compared the new algorithm to the previous one in terms of the expected handshake number. It should be noted, however, that this measurement of the performance is relative. Consider the simple case of star graphs; the expected handshake number is no more than 1, whatever the size of the star. This does not mean that the algorithm is not good, the star graph does not allow a greater matching. We recall the efficiency of randomised handshakes algorithm introduced in [10].

Let $\mathcal{A}$ be any arbitrary randomised handshake algorithm working on graphs. Its *efficiency* over a given graph $G = (V, E)$, with $E \neq \emptyset$, is the ratio $\Delta_{\mathcal{A}}(G) = \overline{R}_{\mathcal{A}}(G)/K(G)$, where $\overline{R}_{\mathcal{A}}(G)$ is the expected handshake number whenever $\mathcal{A}$ is applied to G and $K(G)$ is the matching number of G.

According to this definition the efficiency of our algorithm is 1 for complete graphs, star graphs and large-star graphs. It is asymptotically $1 - e^{-2}$ for chain graphs.

In the sequel of the study we first find a lower bound for the efficiency of the algorithm over the class of all graphs; this lower bound turns to be tight. In the case of trees (sparsely connected networks), the following study shows that the algorithm realises a better performance, having an efficiency greater than 0.75.

All graphs considered in this section have a non-empty set of edges and $\mathcal{A}$ denotes our handshake algorithm. We denote by $K(G)$ the matching number of G.

3.1 Lower Bound on the Efficiency

Let us recall first that the efficiency of the MSZ-algorithm for the complete graph K_n is asymptotically $1/n$ and, therefore the algorithm over the class of all graphs does not admit a positive lower bound. We prove that our algorithm over this class realises an efficiency of at least 0.5.

Proposition 3.1. *For any non-empty graph G, $\overline{R}_{\mathcal{A}}(G) \geq 0.5$.*

Proof: It suffices to see that the size of any maximal matching in G is at least half that of any other.

■

Remark 3.2. *The straightforward proof of the proposition goes beyond its main goal. Indeed, it shows that with probability 1, the handshake number is at least half of the matching number (ideal performance).*

We prove now that the lower bound 1/2 for the efficiency of the algorithm over all graphs is a tight one.

Proposition 3.3. *There is sequence of graphs G_n of size $4n$, such that the efficiency of G_n tends to $1/2$, as $n \to \infty$.*

Proof: To prove this consider a graph $G(n_1, n_2)$ with $2n_1 + 2n_2$ vertices in $V \cup U \cup W \cup X$, with $V = \{v_1, v_2, \cdots, v_{n_1}\}$, $U = \{u_1, u_2, \cdots, u_{n_2}\}$, $W = \{w_1, w_2, \cdots, w_{n_1}\}$ and $X = \{x_1, x_2, \cdots, x_{n_2}\}$. For each $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$ there are edges $\{v_i, w_i\}$, $\{w_i, u_j\}$ $\{u_j, x_j\}$, see Fig. 4. It is easy to see that $K(G(n_1, n2)) = n_1 + n_2$, and if we denote by $\mathbb{E}(n_1, n_2)$

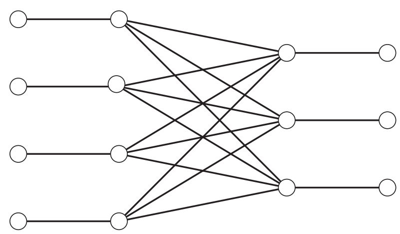

Figure 4: Graph $G(4,3)$

the expected number of handshake in $G(n_1, n_2)$, a straightforward computation leads to the following inductive formula:

$$\begin{aligned} \mathbb{E}(n,0) = \quad \mathbb{E}(0,n) = \quad & n \quad \text{and} \\ \mathbb{E}(n_1,n_2) \quad = \quad & 1 + \tfrac{n_1 n_2 \mathbb{E}(n_1-1,n_2-1) + n_1 \mathbb{E}(n_1-1,n_2) + n_2 \mathbb{E}(n_1,n_2-1)}{n_1 n_2 + n_1 + n_2}, \quad \text{for } n_1, n_2 \geq 1. \end{aligned}$$

Hence, for $n_1 = n_2$, a simple reasoning yields:

$$\mathbb{E}(n,n) \leq 1 + \mathbb{E}(n-1, n-1) + \frac{2n}{n^2 + 2n}.$$

Hence $\mathbb{E}(n,n) = n + o(n)$. Now letting $G_n = G(n,n)$, we have $K(G_n) = 2n$ and the theorem follows.

J

Dyer and Frieze [3] finds the tight lower bound 0.7690397... for forests.

4 Conclusion and Further Investigations

The proposed algorithm is sequential. The authors are studying the distributed version of this algorithm. The study seems interesting in both efficiency mesure and complexity. We prove, in particular, that the complexity of a version of this algorithm is logarithmic in size of the graph.

Acknowledgements

The authors are grateful to the anonymous referee for helpful comments and suggestions.

References

[1] B. Bollobás, Random graphs, *Cambridge University Press*, 2001.

[2] E. Demaine, Protocols for non-deterministic communication over synchronous channels, *Proceedings of the 12th International parallel processing symposium / 9th Symposium on parallel and distributed processing (IPPS/SPDP'98), IEEE Computer Society*, 24–30, 1998.

[3] M. E. Dyer, A. M. Frieze, Randomized greedy matching, *Random Struct. Algorithms*, 2(1):29–46, 1991.

[4] P. Flajolet, A Seating Arrangement Problem, *ALCOM-FT*, http://algo.inria.fr/libraries/autocomb/fatmen.ps, 1997.

[5] N. Francez, M. Rodeh, A distributed abstract data type implemented by a probabilistic communication scheme, pages, *Proceedings of the 21st ACM Symposium on FOCS, ACM Press*, 133–138, 1980.

[6] M. Hanckowiak, M. Karonski, A. Panconesi, On the Distributed Complexity of Computing Maximal Matchings, *SIAM J. Discrete Math.*, 15(1):41–57, http://epubs.siam.org/sam-bin/dbq/article/37312, 2001.

[7] C. Lavault, Évaluation des algorithmes distribués, *HERMES*, 1995.

[8] R. Motwani and P. Raghavan, Randomized algorithms, *Cambridge University Press*, 1995.

[9] Y. Métivier, N. Saheb, A. Zemmari, Randomized rendezvous, *Mathematics and computer science : Algorithms, trees, combinatorics and probabilities*,Trends in mathematics, 183–194, 2000.

[10] Y. Métivier, N. Saheb, A. Zemmari, Analysis of a randomized rendezvous algorithm, *Information and Computation*, 184: 109–128, 2003.

[11] J. Reif and P. Spirakis, Real-time synchronization of interprocess communications, *ACM Journal of transactions on programming languages and systems*, 6(2):215–238, 1984.

[12] J. Reif and P. Spirakis, Real time resource allocation in distributed systems, *Proceedings of the 1th ACM Symposium on principles of distributed computing, ACM Press*, 84–94, 1982.

[13] G. Tel, Introduction to distributed algorithms, *Cambridge University Press*, 2000.