Weighted Distributed Match-Making
(Preliminary Version)

Evangelos Kranakis

Centrum voor Wiskunde en Informatica Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Paul M. B. Vitányi

Centrum voor Wiskunde en Informatica and Universiteit van Amsterdam, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

ABSTRACT

In many distributed computing environments, processes are concurrently executed by nodes in a store-and-forward network. Distributed control issues as diverse as name-server, mutual exclusion and replicated data management, involve making matches between processes. The generic paradigm is a formal problem called "distributed match-making". The applications require solutions to weighted versions of the problem. We define new multi-dimensional and weighted versions, and the relations between the two, and develop a very general method to prove lower bounds on the complexity as a trade-off between number of messages and "distributedness". The resulting lower bounds are tight in all cases we have examined.

1. Introduction

A distributed system consists of computers (*nodes*) connected by a communication network. Each node can communicate with each other node through the network. There is no other communication between nodes. Distributed computation entails the concurrent execution of more than one process, each process being identified with the execution of a program on a computing node. Communication networks come in two types: broadcast networks and store-and-forward networks. In a broadcast network a message by the sender is broadcasted and received by all nodes, including the addressee. In such networks the communication medium is usually suited for this, like ether for radio. An example is Ethernet. Here we are interested in the latter type, store-and-forward networks, where a message is routed from node to node to its destination. Such networks occur in the form of wide area networks like Arpa net, but also as the communication network of a single multicomputer. The necessary coordination of the separate processes in various ways constitutes distributed control. The situation gets more complicated by assuming that processes can migrate from host to host, e.g., to balance the load in the system.

We focus on a common aspect of seemingly unrelated issues in this area, such as name server, mutual exclusion and replicated data management. Namely, processes residing in different nodes need to find each other, without knowing the host addresses of each other in advance. E.g., in a name-server a client process wants to know the host address of a server process providing a particular service; in distributed mutual exclusion a process that wants to enter the critical section needs to know whether some other process wants to do so as well (see [7] for a general overview). This aspect is formalized in [4] as the paradigm "Distributed Match-Making." Roughly speaking, the problem consists in associating with each node v in the network two sets of network nodes, $P(v)$ and $Q(v)$, such that the intersection $P(v) \cap Q(v')$ for each ordered node pair (v,v') is nonempty. We want to minimize the average of $|P(v)| + |Q(v')|$, the average taken over all pairs (v,v'). This average is related to the amount of *communication* (number of messages) involved in implementations of the distributed control issues mentioned. In the application to name-server: v is a server that posts its whereabouts in all nodes $P(v)$, and v' is a client that looks for a particular service (as provided by v) in all nodes in the query set $Q(v')$. Nodes in $P(v) \cap Q(v')$ can establish contact between v and v' by e.g. sending a message to v with the address of v'. In distributed mutual exclusion the interpretation is about the same, except that there is no difference between client and server, i.e. $P(v)=Q(v)$, see e.g. [3], [4]. For application to replicated data management see [4], final version. We make the simplifying assumption that the involved processes do not migrate during execution of a match-making instance.

Previously, for instance in name servers in distributed operating systems, only ad hoc solutions were proposed, e.g, [5] and references in [4]. Lack of any theoretical foundation necessarily entailed that comparisons of the relative merit of different solutions could only be conducted on a haphazard basis. The question about how to distribute the name-server in a distributed operating system that is currently being implemented [6], prompted our initial investigation in distributed match-making [4]. Our analysis leads to a natural quantification of the distributedness of a match-making algorithm, and trade-offs between number of messages and distributedness. Thus, the complexity results hold for the full range from centralized via hierarchical to totally distributed algorithms for match-making. As pointed out in [4], in many applications we are actually interested in *weighted* versions, i.e., we want to minimize the average of $|P(v')| + \alpha(v',v)|Q(v)|$. It turns out that to do so we have to look at *multi-dimensional* versions first. We develop a very general argument to obtain lower bounds on both versions that include as special case the ones in [4]. The structure of the paper is as follows. First we formally define the multidimensional version and the weighted version of the problem. In the next section we derive the lower bound trade-off (Theorem 1) on the multidimensional case. We then show that the lower bound is tight for the binary n-cube topology and projective n-space topology, by exhibiting distributed algorithms that match the lower bound. In the final section, we derive the promised lower bound on the weighted version of distributed match-making (Theorem 2). This development enhances applicability of the theory of distributed match-making in practical situations.

1.1. Formal Framework

To simplify notation from now on let the set N of network nodes be equal to $\{1,...,n\}$. Let $\mathbf{P} = (P_a,...,P_s)$ be a communication strategy in a given network as follows. (For convenience, with some abuse of notation, we use letters a through s to denote both node variables and the numbers 1 through s.) For each $j = a,...,s$, $P_j: N \to 2^N$ is a total function, and for each s-tuple $(a',b',...,s')$ of nodes $P_a(a') \cap P_b(b') \cap ... \cap P_s(s') \neq \varnothing$. For any s-tuple $(a',b',...,s')$ of nodes let $m[\mathbf{P}](a',...,s') = |P_a(a')| + ... + |P_s(s')|$ be the number of messages required for the match-making instance $(a,...,s)$ following strategy $\mathbf{P}$. The average number $M[\mathbf{P}]$ of point-to-point messages necessary for match-making is (deleting here and elsewhere $[\mathbf{P}]$ because $\mathbf{P}$ is understood):

$$M = n^{-s} \sum m(a',...,s'), \tag{1}$$

with the sum taken over $(a',...,s') \in N^s$. Let us interpret the case $s=2$ in terms of the name-server, in order to give the intuitive background for considering weighted versions. Since a server i posts its whereabouts at all the nodes in $P(i)$, by sending messages to all these nodes, and a client j queries each node in $Q(j)$, we have $\mathbf{P} = (P,Q)$. The number $m(i,j)$ of point-to-point messages in the match-making instance (i,j) must be at least $|P(i)| + |Q(j)|$. Another more general situation arises when the average call for a service i by a client j occurs $\alpha(i,j)$-times more often than the average posting of a service available at i. Here one wants to minimize (1), with $m(i,j) = |P(i)| + \alpha(i,j)|Q(j)|$. A similar case arises when in the match-making instance (i,j) the server i is allowed to post $p(i,j)$-many times to the nodes in $P(i)$ and the customer j is allowed to query $q(i,j)$-many times the nodes in $Q(j)$ in order to increase reliability of the network. In this case the number $m(i,j)$ of point-to-point messages is equal to $p(i,j)|P(i)| + q(i,j)|Q(j)|$.

In contrast to the *post-query* case ($s=2$), which is best visualized in two dimensions, the more general case ($s>2$) is best visualized in s dimensions. (Each axis is marked with a node from $1,...,n$ and at the vertex $(a,...,s)$ a point of the intersection $\cap P_r$ is located.) To obtain lower bounds on the complexity of the weighted versions and the versions with retransmission, it turns out that it is advantageous to analyse the general s-dimensional case first.

2. The s-Dimensional Lower Bounds

In this section the main lower bound results are derived. In order to be able to prove the most general results possible it will be necessary to formulate the required concepts with a higher level of abstraction than in the introduction. The motivation however is derived from the previous section, and the results are necessary to resolve weighted match-making in the next section.

Let N, $N_a,\dots,N_s$ be nonempty sets, and $n = |N|$, $n_a = |N_a|,\dots,$ $n_s = |N_s|$. For convenience we set $N = \{1,\dots,n\}$. It is important to note that, in this general setting, N_a , ..., N_s are *arbitrary* finite sets (of integers), in particular, they can have more elements than N. Consider a *strategy* $\mathbf{P} = \{P_a(a'),\dots,P_s(s') : a' \in N_a,\dots,s' \in N_s\}$, with total mappings $P_r : N_r \to 2^N$, and $p_r(x) = |P_r(x)|$, for $r \in \{a,\dots,s\}$. Let K_i be the set of s-tuples $(a',\dots,s')$ such that $i \in P_a(a') \cap \cdots \cap P_s(s')$ and let $k_i = |K_i|$. (It is clear that if each of these intersections is nonempty then $k_1 + \cdots + k_n \geqslant n_a \cdots n_s$, and equality holds if all intersections are singleton sets.) For the given strategy $\mathbf{P}$ define the *product* Π and the *sum* M associated with $\mathbf{P}$ by the following formulas:

$$\Pi = (n_a \cdots n_s)^{-1} \sum p_a(a') \cdots p_s(s') ,$$
$$M = (n_a \cdots n_s)^{-1} \sum [p_a(a') + \cdots + p_s(s')] ,$$

with the sums taken over $(a',\dots,s') \in N_a \times \dots \times N_s$. Further, for $r \in \{a,\dots,s\}$ define

$$M_r = n_r^{-1} \sum p_r(r')$$

(with summation over $r' \in N_r$), so that

$$\Pi = M_a \cdots M_s \text{ and } M = M_a + \cdots + M_s . \tag{2}$$

The main result of the section is the following

Theorem 1. *For any strategy* $\mathbf{P}$ *the following inequalities hold:*

$$\Pi \geqslant (n_a \cdots n_s)^{-1} \left[\sum_{i \in N} k_i^{1/s} \right]^s \quad \text{and} \quad M \geqslant s(n_a \cdots n_s)^{-1/s} \left[\sum_{i \in N} k_i^{1/s} \right].$$

Remark If $n_a = \cdots = n_s = n$ then

$$\Pi \geqslant \left[n^{-1} \sum_{i=1}^{n} k_i^{1/s} \right]^s \quad \text{and} \quad M \geqslant sn^{-1} \left[\sum_{i=1}^{n} k_i^{1/s} \right].$$

Additionally considering the symmetric case where all k_i's are equal, viz., $k_i = n^{s-1}$, $i = 1,\dots,n$. Then Theorem 1 specializes to the important "truly distributed" case: $\Pi \geqslant n^{s-1}$ and $M \geqslant sn^{(s-1)/s}$. We will find matching upper bounds below.

Remark. M equals the right-hand side of the inequality in which it occurs, exactly when $M_a = \cdots = M_s$, i.e. the strategy $\mathbf{P}$ is optimal exactly when the average number of messages is equally balanced in all directions.

Proof: The following inequality, also known as *inequality of the arithmetic and geometric means,* holds for s-many nonnegative real numbers $\alpha, \dots, \sigma$,

$$\alpha + \cdots + \sigma \geqslant s(\alpha \cdots \sigma)^{1/s}. \tag{3}$$

In fact, equality holds exactly when all the summands are equal [2]. Thus, the inequality in the Theorem concerning the sum M follows immediately from the inequality concerning product Π, identities (2), and inequality (3). It is only left to prove the inequality concerning Π. For each $r \in \{a,...,s\}$ and each $i \in N$, define the set $H_{r,i} \subseteq N_r$ such that $r' \in H_{r,i}$ iff for some s-tuple $(a',..,r',.,s')$ holds

$$i \in P_a(a') \cap ... \cap P_r(r') \cap ... \cap P_s(s').$$

Set $h_{r,i} = |H_{r,i}|$. Clearly, for all $i = 1,...,n$,

$$\begin{aligned} h_{a,i} \cdots h_{s,i} &= |H_{a,i} \times \cdots \times H_{s,i}| \\ &\geqslant |\{(a', \ldots, s') : i \in P_a(a') \cap \cdots \cap P_s(s')\}| = k_i . \end{aligned} \tag{4}$$

Now, for all $r \in \{a,...,s\}$,

$$\begin{aligned} \sum_{i \in N} h_{r,i} &\leqslant \sum_{i \in N} |\{r' : i \in P_r(r')\}| \\ &= \sum_{i \in N} \sum_{r' \in N_r} |\{(i,r') : i \in P_r(r')\}| \\ &= \sum_{r' \in N_r} |\{i : i \in P_r(r')\}| \\ &= \sum_{r' \in N_r} p_r(r') = n_r\, M_r . \end{aligned} \tag{5}$$

To obtain the lower bound on Π, we now proceed as follows.

$$\begin{aligned} \Pi &= M_1 \cdots M_s \quad \text{(by (2))} \\ &\geqslant (n_a \cdots n_s)^{-1} \left[\sum_{\alpha \in N} h_{a,\alpha}\right] \cdots \left[\sum_{\sigma \in N} h_{s,\sigma}\right] \qquad \text{(by (5))} \\ &= (n_a \cdots n_s)^{-1} \sum_{\alpha,...,\sigma \in N} h_{a,\alpha} \cdots h_{s,\sigma} \end{aligned}$$

Set $S(\alpha,...,\rho,\sigma) = h_{a,\alpha} \cdots h_{r,\rho} h_{s,\sigma}$. By cyclically rotating the indices $\alpha, \ldots, \rho, \sigma$ of $S(\alpha,...,\rho,\sigma)$ one obtains the following s-many summands:

$$\begin{aligned} a_1 &= S(\alpha,...,\rho,\sigma) = h_{a,\alpha} \cdots h_{s,\sigma} \\ a_2 &= S(\beta,...,\sigma,\alpha) = h_{a,\beta} \cdots h_{s,\alpha} \\ &\cdots \qquad \cdots \\ a_s &= S(\sigma,...,\pi,\rho) = h_{a,\sigma} \cdots h_{s,\rho} . \end{aligned} \tag{6}$$

Using inequalities (3) and (4) and *regrouping* terms in the resulting product $a_1 \cdots a_s$ it is easy to see that $a_1 + \cdots + a_s \geqslant s(a_1 \cdots a_s)^{1/s} \geqslant s(k_\alpha \cdots k_\sigma)^{1/s}$. After adding the s-many summands of (6), each one summed with respect to $\alpha,...,\sigma$, dividing again by s to eliminate s-multiple copies, and taking into account the last inequality, we obtain:

$$\sum_{\alpha,...,\sigma \in N} h_{a,\alpha} \cdots h_{s,\sigma} \geqslant \sum_{\alpha,...,\sigma \in N} (k_\alpha \cdots k_\sigma)^{1/s} = \left[\sum_{i \in N} k_i^{1/s}\right]^s .$$

This completes the proof of the lower bound of II, and hence the proof of the theorem is complete. □

Corollary. Both propositions 1 and 2 of [4] are immediate consequences of Theorem 1.

3. Optimality

We show that Theorem 1 is optimal in some special cases (which are of sufficient generality), by exhibiting matching strategies.

(Multidimensional Cube Network) Let the number of nodes be $n = 2^d$ and suppose that s is a divisor of d. Addresses of nodes consist of d bits, like $u_1 u_2 \cdots u_d$. Nodes are connected by an edge exactly when they differ by a single bit. Let $\mathbf{P} = (P_1,...,P_s)$ be a strategy, and, for each $r \in \{1,...,s\}$, let $P_r(u_1 \cdots u_d)$ be the set

$$\{x_1 \cdots x_{(r-1)d/s} u_{(r-1)d/s+1} \cdots u_{rd/s} x_{rd/s+1} \cdots x_d : x_i \in \{0,1\}\}.$$

Clearly, each of the above sets has size $2^{(s-1)d/s}$ and $k_i = 2^{(s-1)d} = n^{s-1}$. Thus, one easily obtains that $M \leqslant sn^{(s-1)/s}$, i.e. the average number of point-to-point message transmissions is at most $sn^{(s-1)/s}$. In view of Theorem 1 this strategy is also optimal.

(Multidimensional Projective Plane). Consider generalized mutual exclusion in a distributed setting, where $s-1$ processors are allowed to be in the critical section simultaneously, but not s or more processors. For background and nondistributed solutions we refer to [1]. In [3], Maekawa considers the distributed version of mutual exclusion for $s=2$, the commonly studied variant. In our terminology, for mutual exclusion with $s=2$ we can set $P_1(i)=P_2(i)$, which is some sort of symmetry condition. Each instance of mutual exclusion contains a match-making instance [4]. For the truly distributed case, with $k_1 = ... = k_n = n$ and $s=2$ we find that on the average each match-making instance takes at least $2\sqrt{n}$ messages [4]. Maekawa obtains a similar lower bound, and exhibits an algorithm that achieves $5\sqrt{n}$ [3]. Theorem 1 gives a lower bound of $sn^{(s-1)/s}$ for the generalized version. We exhibit an algorithm that achieves this. The s-dimensional projective plane $PG(s,k)$ has $k^s + k^{s-1} + \cdots + 1 = n$ nodes, each node is incident to $k^{s-1} + k^{s-2} + \cdots + 1$ hyperplanes, and each hyperplane contains $k^{s-1} + k^{s-2} + \cdots + 1$ nodes. Each s-element set of hyperplanes intersects in precisely one node. Let $\mathbf{P}=(P_1,...,P_s)$ be a symmetric strategy with each query set $S(i) = P_1(i) = \cdots = P_s(i)$ of a node i consists of the set of $k^{s-1} + k^{s-2} + \cdots + 1$ nodes incident to a hyperplane containing node i. It does not matter which hyperplane we pick, because any s hyperplanes intersect in a single node. The average cost M of point-to-point messages associated with a particular mutual exclusion instance is therefore (generalizing Maekawa's method for $s=2$ [3]) $O(s(k^{s-1} + k^{s-2} + \cdots + 1)) \approx O(sn^{(s-1)/s})$. In view of Theorem 1 this strategy is also optimal.

4. Weighted Distributed Match-Making

We can now examine weighted distributed match-making. This can be formulated as communication strategies with multiple transmissions allowed. We use Theorem 1 to derive significant lower bounds on the average number of message transmissions in distributed networks when multiple transmissions are allowed. Consider a strategy $\mathbf{P}=(P_a,...,P_s)$, with all parameters as above, and define a weighted version of m. I.e., define the number of messages for the match-making instance $S=(a',...,s')$ as $m[\mathbf{P}](S) = l_a(S)p_a(a') + \cdots + l_s(S)p_s(s')$, where each $l_a(S),...,l_s(S)$ is a positive integer. Then, with S as above, define $N_{r,r'}$ for all $r \in \{a,..,s\}$ and $r' \in N_r$ so that it satisfies:

$$(n_a \cdots n_s) M[\mathbf{P}] = \sum_{S \in N_a \times ... \times N_s} l_a(S)p_a(a') + \cdots + l_s(S)p_s(s')$$

$$= \sum_{a' \in N_a} \left[\sum_{S \in \mathbf{S}_a} l_a(S) \right] p_a(a') + \cdots + \sum_{s' \in N_s} \left[\sum_{S \in \mathbf{S}_s} l_s(S) \right] p_s(s')$$

(with $\mathbf{S}_a=\{a'\}\times N_b \times ... \times N_s, \ldots, \mathbf{S}_s = N_a \times ... \times N_r \times \{s'\}$)

$$= \sum_{a' \in N_a} N_{a,a'} p_a(a') + \cdots + \sum_{s' \in N_s} N_{s,s'} p_s(s') \,, \tag{7}$$

where $N_{a,a'} = \Sigma_{S \in \mathbf{S}_a} l_a(S)$, etc. Define $N'_r = \sum_{r' \in N_r} N_{r,r'}$. Consider the following related strategy $\mathbf{Q}$ for the set of nodes N. $\mathbf{Q} = \{Q_a(a'),...,Q_s(s'): a' \in N'_a,...,s' \in N'_s\}$, such that, for each $r \in \{a,...,s\}$ and each $y \in N_r$ there are $N_{r,y}$ distinct x's, with $Q_r(x)=P_r(y)$. I.e., $\mathbf{Q}$ is formed from the strategy $\mathbf{P}$ by repeating each set $P_r(y)$, $N_{r,y}$-times. Let $q_r(x) = |Q_r(x)|$. Note that we have chosen the definitions such that

$$\sum_{r' \in N'_r} q_r(r') = \sum_{r' \in N_r} N_{r,r'} p_r(r') \,,$$

for all r from a through s. Then we can relate $M[\mathbf{P}]$ with $\Pi[\mathbf{Q}]$:

$$M[\mathbf{P}] = (n_a \cdots n_s)^{-1} \left[\sum_{a' \in N'_a} q_a(a') + \cdots + \sum_{s' \in N'_s} q_s(s') \right] \quad \text{(by (7))}$$

$$\geqslant s(n_a \cdots n_s)^{-1} \left[\left[\sum_{a' \in N'_a} q_a(a') \right] \cdots \left[\sum_{s' \in N'_s} q_s(s') \right] \right]^{1/s} \quad \text{(by (3))}$$

$$= s(n_a \cdots n_s)^{-1} (N'_a \cdots N'_s)^{1/s} \Pi[\mathbf{Q}]^{1/s} \quad \text{(by definition)}$$

$$\geqslant s(n_a \cdots n_s)^{-1} \sum_{i \in N} k_i[\mathbf{Q}]^{1/s}. \quad \text{(by Theorem 1)}$$

It remains to compare the quantities $k_i[\mathbf{P}]$, $k_i[\mathbf{Q}]$. This can be done by comparing the sizes of the sets $K_i[\mathbf{P}]$, $K_i[\mathbf{Q}]$. Now, for each s-tuple $(a',...,s')$ such that

$i \in P_a(a') \cap ... \cap P_s(s')$ there are at least $N_{a,a'} \cdots N_{s,s'}$ s-tuples $(a'',...,s'')$ such that $i \in Q_a(a'') \cap ... \cap Q_s(s'')$. Namely, there are $N_{r,r'}$ copies of $P_r(r')$, for r,r' from a,a' through s,s', in $\mathbf{Q}$. Therefore, each $(a',...,s') \in K_i[\mathbf{P}]$ corresponds to a disjoint subset of at least $N_{a,a'} \cdots N_{s,s'}$-many s-tuples in the set $K_i[\mathbf{Q}]$'s. Hence, it has been proved that

$$M[\mathbf{P}] \geqslant s(n_1 \cdots n_s)^{-1} \sum_{i \in N} \left[\sum \{ N_{a,a'} ... N_{s,s'} : (a',...,s') \in K_i[\mathbf{P}] \} \right]^{1/s}. \tag{8}$$

In particular, with some computation we can specialize the general result (8) to:

Theorem 2. *For any strategy* $\mathbf{P}$, *if there are positive integers* $\lambda_a,...,\lambda_s$ *such that for all* $(a',...,s')$ *holds* $m(a',...,s') = \lambda_a p_a(a') + \cdots + \lambda_s p_s(s')$, *then*

$$M \geqslant \frac{s(\lambda_1 \cdots \lambda_s)^{1/s}}{n} \sum_{i=1}^{n} k_i^{1/s}. \tag{9}$$

Moreover, the quantity M *equals the right-hand side of the inequality above, exactly when* $\lambda_a M_a = \cdots = \lambda_s M_s$. □

Corollary. Routine calculation shows that Theorem 2 also holds for *rational* λ's. (Hint: for $\lambda_r = p_r / q_r$ apply Theorem 2 for $\mu_r = c\lambda_r$ with $c = q_a ... q_s$ $(r \in \{a,...,s\})$. This gives an inequality for cM. Substituting the λ's for the μ's, we can cancel c on both sides of the inequality.)

References

[1] Fischer, M.J., Lynch, N.A., Burns, J.E., and Borodin, A., *Distributed FIFO allocation of identical resources using small shared space*, Massachusetts Institute of Technology, Cambridge, Mass., Report MIT/LCS/TM-290, October 1985.

[2] Hardy, G. H., Littlewood, J. E. and Polyâ, G., *Inequalities,* Cambridge University Press, 1934.

[3] Maekawa, M., *A* $\sqrt{N}$ *Algorithm for Mutual Exclusion in Decentralized Systems*, ACM Transactions on Computer Systems **3** (1985), pp. 145-159.

[4a] Mullender, S. J. and Vitânyi, P. M. B., *Distributed Match-Making for Processes in Computer Networks, Preliminary Version*, Proceedings of the 4th annual ACM Symposium on Principles of Distributed Computing, 1985, pp. 261-271.

[4b] Mullender, S. J. and Vitânyi, P. M. B., *Distributed Match-Making*, Algorithmica, (1988).

[5] Powell, M.L. and Miller, B. P., *Process Migration in DEMOS/MP*, Proceedings of the 9th ACM Symposium on Operating Systems Principles, 1983, pp. 110-119.

[6] Tanenbaum, A.S. and S.J. Mullender, *The Design of a Capability Based Distributed Operating System*, The Computer Journal, 29(1986).

[7] Tanenbaum, A.S., *Computer Networks*, Prentice-Hall, 1981.