# Optimal, Distributed Decision-Making: The Case of No Communication


Stavros Georgiades[1], Marios Mavronicolas[2], and Paul Spirakis[3]

[1] Department of Computer Science, University of Cyprus, Nicosia CY-1678, Cyprus. Email: stavrosg@turing.cs.ucy.ac.cy

[2] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269–3155. Part of the work of this author was performed while at Department of Computer Science, University of Cyprus, Nicosia CY-1678, Cyprus, and while at AT&T Labs – Research, as a visitor to the Special Year on Networks, DIMACS Center for Discrete Mathematics and Theoretical Computer Science. Email: mavronic@engr.uconn.edu

[3] Department of Computer Engineering and Informatics, University of Patras, Patras, Greece & Computer Technology Institute, Patras, Greece. Partially supported by the EU ESPRIT Long Term Research Project ALCOM-IT (Proj. # 20244), and by the Greek Ministry of Education. Email: spirakis@cti.gr



**Abstract.** We present a combinatorial framework for the study of a natural class of *distributed optimization problems* that involve *decision-making* by a collection of $n$ *distributed agents* in the presence of incomplete information; such problems were originally considered in a *load balancing* setting by Papadimitriou and Yannakakis (*Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing,* pp. 61–64, August 1991). For any given *decision protocol* and assuming *no communication* among the agents, our framework allows to obtain a combinatorial inclusion-exclusion expression for the probability that no "overflow" occurs, called the *winning probability,* in terms of the *volume* of some simple combinatorial polytope.

Within our general framework, we offer a complete resolution to the special cases of *oblivious algorithms,* for which agents do not "look at" their inputs, and *non-oblivious algorithms,* for which they do, of the general optimization problem. In either case, we derive optimality conditions in the form of combinatorial polynomial equations. For oblivious algorithms, we explicitly solve these equations to show that the optimal algorithm is simple and *uniform,* in the sense that agents need not "know" $n$. Most interestingly, we show that optimal non-oblivious algorithms must be *non-uniform:* we demonstrate that the optimality conditions admit different solutions for particular, different "small" values of $n$; however, these solutions improve in terms of the winning probability over the optimal, oblivious algorithm. Our results demonstrate an interesting trade-off between the amount of knowledge used by agents and uniformity for optimal, distributed decision-making with no communication.



G. Ciobanu and G. Păun (Eds.): FCT'99, LNCS 1684, pp. 293–303, 1999.

## 1 Introduction

In a *distributed optimization problem,* each of $n$ *distributed agents* receives a private *input,* communicates possibly with other agents to learn about their own inputs, and decides, based on this possibly partial knowledge, on an *output*; the task is to maximize a common objective function. Such problems were originally introduced by Papadimitriou and Yannakakis [9], in an effort to understand the crucial economic value of *information* [1] as a computational resource in a distributed system (see, also, [2,4,8,10]). Intuitively, the more information available to agents, the better decisions they make, but naturally the more expensive the solution becomes due to the need for increased communication. Such natural trade-offs between communication cost and the quality of decision-making have been studied in the contexts of *communication complexity* [7] and *concurrency control* [6] as well.

Papadimitriou and Yannakakis [9] examined the special case of such distributed optimization problems where there are just *three* agents. More specifically, Papadimitriou and Yannakakis focused on a natural *load balancing* problem (see, e.g., [3,5,11], where each agent is presented with an input, and must decide on a binary output, representing one of two available "bins," each of capacity one; the input is assumed to be distributed uniformly in the unit interval $[0, 1]$. The load balancing property is modeled by requiring that no "overflow" occurs, namely that inputs dropped into each "bin" not exceed together its capacity. Papadimitriou and Yannakakis [9] pursued a comprehensive study of how the best possible probability, over the distribution of inputs, of "no overflow" depends on the amount of communication available to the agents. For each possible communication pattern, Papadimitriou and Yannakakis [9] discovered the corresponding optimal decision protocol to be unexpectedly sophisticated. The proof techniques of Papadimitriou and Yannakakis [9] were surprisingly complex, even for this seemingly simplest case, combining tools from nonlinear optimization with geometric and combinatorial arguments; these techniques have not been hoped to be conveniently extendible to instances of even this particular load balancing problem whose size exceeds three.

In this work, we introduce a novel combinatorial framework in order to enhance the study of general instances of distributed optimization problems of the kind considered by Papadimitriou and Yannakakis [9]. More specifically, we proceed to the general case of $n$ agents, with each still receiving an input uniformly distributed over $[0, 1]$ and having to choose one out of two "bins"; however, in order to render the problem interesting, we make the technical assumption that the capacity of each "bin" is equal to $\delta$, for some real number $\delta$ possibly greater than one, so as to compensate for the increase in the number of players. Papadimitriou and Yannakakis [9] focused on a specific kind of decision protocols by which each agent chooses a "bin" by comparing a "weighted average" of the inputs it "sees" against some "threshold" value; in contrast, our framework allows for the consideration of *general* decision protocols by which each agent decides by using *any* (computable) function of the inputs it "sees".

Our starting point is a combinatorial result that provides an explicit *inclusion-exclusion* formula [12, Section 2.1] for calculating the *volume* of any particular geometric polytope, in any given dimension, of some speficic form (Proposition 1). Roughly speaking, such polytopes are the intersection of a simplex in the positive quadrant with an orthogonal parallelepiped. An immediate implication of this result are inclusion-exclusion formulas for calculating the (conditional) probability of “no overflow” for a *single* “bin,” as a function of the capacity $\delta$ and the number of inputs that are dropped into the “bin” (Lemmas 1 and 2).

In this work, we focus on the case where there is no communication among the agents, which we completely settle for the case of general $n$. Since communication comes at a cost, which it would be desirable to avoid, it is both natural and interesting to choose the case of no communication as an initial “testbed”. We consider both *oblivious algorithms,* where players do not “look” at their inputs, and *non-oblivious algorithms,* where they do. For each case, we are interested in optimal algorithms.

We first consider oblivious algorithms. Our first major result is a combinatorial expression in the form of an inclusion-exclusion formula for the probability that “no overflow” occurs for either of the “bins” (Theorem 1). This formula incorporates a suitable inclusion-exclusion summation, over all possible input vectors, of the probabilities, induced by any particular decision algorithm, on the space of all possible decision vectors, as a function of the corresponding input vector. The coefficients of these probabilities in the summation are independent of any specific parameters of the algorithm, while they do depend on the input vector. A first implication of this expression is the reduction of the general problem of computing the probability that “no overflow” occurs to the problem of computing, given a particular decision algorithm, the probability distribution of the binary output vectors it yields. Most significantly, this expression contributes a methodology for the design of *optimal* decision algorithms “compatible” with any specific pattern of communication, and not just for the case of no communication that we particularly examine: one simply renders only those parameters of the decision algorithm that correspond to the possible communications, and computes values for these parameters that maximize the combinatorial expression as a function of these parameters. This is done by solving a certain system of *optimality conditions* (Corollary 2).

We demonstrate that our methodology for designing optimal algorithms for distributed decision-making is both effective and useful by applying it to the special case of no communication that we consider. We manage to settle down completely this case for oblivious algorithms. We exploit the underlying “symmetry” with respect to different agents in order to simplify the optimality conditions (by observing that all parameters satisfying them must be equal). This simplification reveals a beautiful combinatorial structure; more specifically, we discover that each optimality condition eventually amounts to zeroing a particular “symmetric” polynomial of a single variable. In turn, we explicitly solve these conditions to show that the best possible oblivious algorithm for the case of no communication is the very simple one by which each agent uses 1/2 as its "threshold" value; given that the optimal (non-oblivious) algorithms presented by Papadimitriou and Yannakakis for the special case where $n = 3$ are somehow unexpectedly sophisticated, it is perhaps surprising that such simple oblivious algorithm is indeed optimal for *all* values of $n$.

We next turn to non-oblivious algorithms, still for the case of no communication. In that case, we demonstrate that the optimality conditions do not admit a "constant" solution. Through a more sophisticated analysis, we are able to compute more complex expressions for the optimality conditions, which still allow exploitation of "symmetry". We consider the particular instances of the optimality conditions where $n = 3$ and $\delta = 1$ (considered by Papadimitriou and Yannakakis [9]), and $n = 4$ and $\delta = 4/3$. We discover that the optimal algorithms are different in each of these cases. However, they achieve larger winning probabilities than their oblivious counterparts. This shows that the improved performance of non-oblivious algorithms comes at the cost of sacrificing *uniformity.*

We believe that our work opens up the way for the design and analysis of algorithms for general instances of the problem of distributed decision-making in the presence of incomplete information. We envision that algorithms that are more complex, general communication patterns, and more realistic assumptions on the distribution of inputs, can all be treated in our combinatorial framework to yield optimal algorithms for distributed decision-making for these cases as well.

## 2 Framework and Preliminaries

Throughout, for any bit $b \in \{0, 1\}$ and real number $\alpha \in [0, 1]$, denote $\overline{b}$ the complement of $b$, and $\alpha^{(b)}$ to be $\alpha$ if $b = 1$, and $1 - \alpha$ if $b = 0$. For any binary vector $\mathbf{b}$, denote $|\mathbf{b}|$ the number of entries of $\mathbf{b}$ that are equal to one.

### 2.1 Model and Problem Definition

Consider a collection of $n$ distributed agents $P_1, P_2, \ldots, P_n$, called *players,* where $n \geq 2$. Each player $P_i$ receives an *input* $x_i$, which is the value of a random variable distributed uniformly over $[0, 1]$; denote $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle^{\mathrm{T}}$ the *input vector.* Associated with each player $P_i$ is a (local) *decision algorithm* $A_i$, that may be either deterministic or randomized, and "maps" the input $x_i$ to $P_i$'s *output* $y_i$. A *distributed decision algorithm* is a collection $\mathcal{A} = \langle A_1, A_2, \ldots, A_n \rangle$ of (local) decision algorithms, one for each player.

A *deterministic decision algorithm* is a function $A_i : [0, 1] \rightarrow \{0, 1\}$, that maps the input $x_1$ to $P_i$'s *output* $y_i = A_i(x_i)$; denote $\mathbf{y}_{\mathcal{A}}(\mathbf{x}) = \langle A_1(x_1), A_2(x_2), \ldots, A_n(x_n) \rangle^{\mathrm{T}}$ the *output vector of* $\mathcal{A}$ *on input vector* $\mathbf{x}$. A *deterministic, single-threshold decision algorithm* is a deterministic decision algorithm $A_i$ that is a *single-threshold* function; that is,

$$A_i(x_i) = \begin{cases} 0, \ x_i \leq a_i \\ 1, \ x_i > a_i \end{cases} ,$$

where $0 \leq a_i \leq \infty$. Distributed, deterministic decision algorithms and distributed, deterministic single-threshold decision algorithms can be defined in the natural way.

Say that $\mathcal{A}$ is *randomized oblivious* if for each $i$, $1 \leq i \leq n$, $A_i$ is a *probability distribution* on $\{0,1\}$; that is, $A_i(0)$ (resp., $A_i(1)$) is the probability that player $P_i$ decides 0 (resp., 1). Denote $a_i = A_i(0)$. Thus, a *distributed, randomized oblivious decision algorithm* is a collection $\mathcal{A} = \langle a_i, a_2, \ldots, a_n \rangle$ is a collection of (local) randomized, oblivious decision algorithms, one for each player.

For each $b \in \{0,1\}$, define $S_b = \sum_{i:A_i(\mathbf{x})=b} x_i$; thus, $S_b$ is the sum of the inputs of the players that "decide" $b$. For each parameter $\delta > 0$, we are interested in the event that neither $S_0$ nor $S_1$ exceeds $\delta$; denote $\mathbf{Pr}_{\mathcal{A}}(S_0 \leq \delta \text{ and } S_1 \leq \delta)$ the probability, taken over all input vectors $\mathbf{x}$, that this event occurs. We wish to maximize $\mathbf{Pr}_{\mathcal{A}}(S_0 \leq \delta \text{ and } S_1 \leq \delta)$ over all protocols $\mathcal{A}$; any protocol $\mathcal{A}$ that maximizes $\mathbf{Pr}_{\mathcal{A}}(S_0 \leq \delta \text{ and } S_1 \leq \delta)$ is a corresponding *optimal* protocol.

### 2.2 Combinatorial Preliminaries

For any *polytope* $\mathbf{\Pi}$, denote $\mathsf{Vol}(\mathbf{\Pi})$ the *volume* of $\mathbf{\Pi}$. A cornerstone of our analysis is the following combinatorial result that calculates the volume of any particular polytope that has some specific form. Fix any integer $m \geq 2$. Consider any pair of vectors $\mathbf{a} = \langle \alpha_1, \alpha_2, \ldots, \alpha_m \rangle^{\mathrm{T}}$, and $\mathbf{b} = \langle \beta_1, \beta_2, \ldots, \beta_m \rangle^{\mathrm{T}}$, where for any $l$, $1 \leq l \leq m$, $0 \leq \alpha_l, \beta_l < \infty$. Define the $m$-dimensional polytope

$$\mathbf{\Pi}^{(m)}(\mathbf{a}, \mathbf{b}) = \{\langle x_1, x_2, \ldots, x_m \rangle^T \in [0, \beta_1] \times [0, \beta_2] \times \ldots \times [0, \beta_m] \mid \sum_{l=1}^{m} \frac{x_l}{\alpha_l} \leq 1\}\,.$$

Thus, $\mathbf{\Pi}^{(m)}(\mathbf{a}, \mathbf{b})$ is the intersection of the $m$-dimensional simplex

$$\mathbf{\Pi}^{(m)}(\mathbf{a}) = \{\langle x_1, x_2, \ldots, x_m \rangle^T \mid \sum_{l=1}^{m} \frac{x_l}{\alpha_l} \leq 1\}\,,$$

with the $m$-dimensional orthogonal parallelepiped $[0, \beta_1] \times [0, \beta_2] \times \ldots \times [0, \beta_m]$; The vectors $\mathbf{a}$ and $\mathbf{b}$ determine the simplex and the orthogonal parallelepiped, respectively. We provide an explicit inclusion-exclusion formula for calculating the volume of $\mathbf{\Pi}^{(m)}(\mathbf{a}, \mathbf{b})$.

**Proposition 1.**

$$\begin{aligned}&\mathsf{Vol}(\mathbf{\Pi}^{(m)}(\mathbf{a}, \mathbf{b})) = \\ &V_{\emptyset} - \sum_{1 \leq i \leq m} V_{\{i\}} + \sum_{1 \leq i < j \leq m} V_{\{i,j\}} - \sum_{1 \leq i < j < k \leq m} V_{\{i,j,k\}} + \cdots + (-1)^m V_{\{1,2,\ldots,m\}}\,,\end{aligned}$$

*where*

$$V_{\emptyset} = \frac{1}{m!} \prod_{l=1}^{m} \alpha_l\,,$$

*and for each set of indices* $\mathcal{I} \subseteq \{1, 2, \ldots, m\}$,

$$V_{\mathcal{I}} = \begin{cases} V_{\emptyset}(1 - \sum_{l \in \mathcal{I}} \beta_l \alpha_l^{-1})^m, & 1 > \sum_{l \in \mathcal{I}} \beta_l \alpha_l^{-1} \\ 0, & 1 \leq \sum_{l \in \mathcal{I}} \beta_l \alpha_l^{-1} \end{cases}.$$

### 2.3 Probabilistic Lemmas

In this section, we present two straightforward implications of Proposition 1 that will be used later.

**Lemma 1.** *Assume that for each* $i$, $1 \leq i \leq m$, $x_i$ *is uniformly distributed over* $[0, \beta_i]$. *Then, for any parameter* $\delta > 0$,

$$\mathbf{Pr}(\sum_{i=1}^{m} x_i \leq \delta) = \frac{1}{m!} \frac{\sum_{I \subseteq \{1,2,\ldots,m\}, \sum_{l \in I} \beta_l < \delta} (-1)^{|I|} (\delta - \sum_{l \in I} \beta_l)^m}{\prod_{l=1}^{m} \beta_l}.$$

An immediate implication of Lemma 1 concerns the special case where for each $i$, $1 \leq i \leq n$, $\beta_i = 1$.

**Corollary 1.** *Assume that for each* $i$, $1 \leq i \leq m$, $x_i$ *is uniformly distributed over* $[0, 1]$. *Then, for any parameter* $\delta > 0$,

$$\mathbf{Pr}(\sum_{i=1}^{m} x_i \leq \delta) = \frac{1}{m!} \sum_{0 \leq l \leq m, l < \delta} (-1)^l \binom{m}{l} (\delta - l)^m.$$

We also show:

**Lemma 2.** *Assume that for each* $i$, $1 \leq i \leq m$, $x_i$ *is uniformly distributed over* $[\beta_i, 1]$. *Then, for any parameter* $\delta > 0$,

$$\mathbf{Pr}(\sum_{i=1}^{m} x_i \leq \delta) =$$
$$= 1 - \frac{1}{m!} \frac{\sum_{I \subseteq \{1,2,\ldots,m\}, |I| - \sum_{l \in I} \beta_l < m - \delta} (-1)^{|I|} (m - \delta - |I| + \sum_{l \in I} \beta_l)^m}{\prod_{l=1}^{m} (1 - \beta_l)}.$$

## 3 Oblivious Algorithms

### 3.1 The Winning Probability

We show:

**Theorem 1.** *Assume that* $\mathcal{A}$ *is any randomized oblivious algorithm. Then,*

$$\mathbf{Pr}_{\mathcal{A}}(S_0 \leq \delta \text{ and } S_1 \leq \delta) =$$
$$= \delta^n \sum_{\mathbf{b} \in \{0,1\}^n} \left( \frac{1}{|\mathbf{b}|!} \sum_{0 \leq l \leq |\mathbf{b}|, l < \delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \right.$$
$$\left. \frac{1}{(n - |\mathbf{b}|!)} \sum_{0 \leq l \leq n - |\mathbf{b}|, l < \delta} (-1)^l \binom{n - |\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{n - |\mathbf{b}|} \cdot \prod_{i=1}^{n} \alpha_i^{(b_i)} \right).$$

The proof of Theorem 1 relies on appropriately using Corollary 1. Theorem 1 immediately implies *necessary* conditions for any optimal protocol. These conditions are determined by simultaneously vanishing the partial derivatives with respect to all parameters of the algorithm.

**Corollary 2 (Optimality conditions for oblivious algorithms).** *Assume that $\mathcal{A}$ is an optimal, randomized oblivious algorithm. Then, for any index $k$,*

$$\sum_{\mathbf{b}\in\{0,1\}^n} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\le l\le|\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \frac{1}{(n-|\mathbf{b}|)!} \sum_{0\le l\le n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \frac{\partial}{\partial \alpha_{kl}} \alpha_k^{(b_k)} \prod_{i=1, i\neq k}^{n} \alpha_i^{(b_i)}\Big) = 0\,.$$

### 3.2 The Optimal Oblivious Algorithm

For each $i$, set $1 \le i \le n$,

$$A_i(\mathbf{x}) = \begin{cases} 0, & x_i \le a_{ii} \\ 1, & x_i > a_{ii} \end{cases};$$

it follows that $\mathbf{Pr}_{\mathcal{A}}(y_i = 0) = a_{ii}$ and $\mathbf{Pr}_{\mathcal{A}}(y_i = 1) = 1 - a_{ii}$. We show that the optimal winning probability is achieved by the very simple protocol for which, for each $i$, $1 \le i \le n$, $a_i = 1/2$.

**Theorem 2.** *Consider the oblivious case. Then,*

$$\max_{\mathcal{A}} \mathbf{Pr}_{\mathcal{A}}(S_0 \le \delta \text{ and } S_1 \le \delta) = \frac{1}{n!}\left(\frac{\delta}{2}\right)^n \cdot \sum_{r=0}^{n} \Big(\binom{n}{r}^2 \sum_{0\le l\le r, l<\delta} (-1)^l \binom{r}{l}\left(1-\frac{l}{\delta}\right)^r \cdot \sum_{\substack{0\le l\le n-r,\\ l<\delta}} (-1)^l \binom{n-r}{l}\left(1-\frac{l}{\delta}\right)^{n-r}\Big)$$

*Proof.* Take any optimal protocol $\mathcal{A}$. By Theorem 1,

$$\begin{aligned} &\mathbf{Pr}_{\mathcal{A}}(S_0 \le \delta \text{ and } S_1 \le \delta) \\ &= \delta^n \sum_{\mathbf{b}\in\{0,1\}^n} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\le l\le|\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l}\left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \\ &\quad \frac{1}{(n-|\mathbf{b}|)!} \sum_{0\le l\le n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l}\left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \prod_{i=1}^{n} a_{ii}^{(b_i)}\Big)\,. \end{aligned}$$

Fix any index $k$, $1 \leq k \leq n$. By Corollary 2,

$$\begin{aligned}
&\sum_{\mathbf{b}\in\{0,1\}^n} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\leq l\leq |\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \\
&\frac{1}{(n-|\mathbf{b}|)!} \sum_{0\leq l\leq n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \\
&\prod_{i=1, i\neq k}^{n} a_{ii}^{(b_i)} \frac{\partial}{\partial a_{kl}} \mathbf{Pr}_{\mathcal{A}}(y_k = b_k)\Big) \\
&= 0\,,
\end{aligned}$$

so that

$$\begin{aligned}
&\sum_{\mathbf{b}\in\{0,1\}^n, b_k=1} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\leq l\leq |\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \\
&\frac{1}{(n-|\mathbf{b}|)!} \sum_{0\leq l\leq n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \prod_{i=1, i\neq k}^{n} a_{ii}^{(b_i)}\Big) - \\
&\sum_{\mathbf{b}\in\{0,1\}^n, b_k=0} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\leq l\leq |\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \\
&\frac{1}{(n-|\mathbf{b}|)!} \sum_{0\leq l\leq n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \prod_{i=1, i\neq k}^{n} a_{ii}^{(b_i)}\Big) \\
&= 0\,.
\end{aligned}$$

By symmetry of optimality conditions, it follows that $a_{11} = a_{22} = \ldots = a_{nn}$; denote $\alpha$ their common value. Clearly,

$$\begin{aligned}
&\sum_{\mathbf{b}\in\{0,1\}^n, b_k=1} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\leq l\leq |\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \frac{1}{(n-|\mathbf{b}|)!} \cdot \\
&\sum_{0\leq l\leq n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \alpha^{|\mathbf{b}|-1}(1-\alpha)^{n-1-(|\mathbf{b}|-1)}\Big) - \\
&\sum_{\mathbf{b}\in\{0,1\}^n, b_k=0} \Big(\frac{1}{|\mathbf{b}|!} \sum_{0\leq l\leq |\mathbf{b}|, l<\delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot \\
&\frac{1}{(n-|\mathbf{b}|)!} \sum_{0\leq l\leq n-|\mathbf{b}|, l<\delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1-\frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \alpha^{|\mathbf{b}|}(1-\alpha)^{n-1-|\mathbf{b}|}\Big) \\
&= 0\,.
\end{aligned}$$

There are $\binom{n-1}{|\mathbf{b}|-1}$ vectors $\mathbf{b} \in \{0,1\}^n$ with $b_k = 1$; for any such vector, $1 \leq |\mathbf{b}| \leq n$. Similarly, there are $\binom{n-1}{|\mathbf{b}|}$ vectors $\mathbf{b} \in \{0,1\}^n$ with $b_k = 0$; for any such vector, $0 \leq |\mathbf{b}| \leq n-1$. It follows that

$$\sum_{|\mathbf{b}|=1}^{n} \binom{n-1}{|\mathbf{b}|-1} \Big( \frac{1}{|\mathbf{b}|!} \sum_{0 \leq l \leq |\mathbf{b}|, l < \delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot$$

$$\frac{1}{(n-|\mathbf{b}|)!} \sum_{0 \leq l \leq n-|\mathbf{b}|, l < \delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \left(\frac{\alpha}{1-\alpha}\right)^{|\mathbf{b}|-1} \Big) -$$

$$\sum_{|\mathbf{b}|=0}^{n-1} \binom{n-1}{|\mathbf{b}|} \Big( \frac{1}{|\mathbf{b}|!} \sum_{0 \leq l \leq |\mathbf{b}|, l < \delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot$$

$$\frac{1}{(n-|\mathbf{b}|)!} \sum_{0 \leq l \leq n-|\mathbf{b}|, l < \delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \left(\frac{\alpha}{1-\alpha}\right)^{|\mathbf{b}|} \Big)$$

$$= 0.$$

The left-hand side is a polynomial in $\alpha/(\alpha-1)$ of degree $n-1$. Consider any integer $r$, where $0 \leq r \leq n-1$; We show that the coefficients of $(\alpha/(\alpha-1))^r$ and $(\alpha/(\alpha-1))^{n-1-r}$ are the negative of each other.

Thus, the left-hand side is a symmetric polynomial of degree $n-1$. Moreover, we can establish along similar lines that for the case where $n$ is odd, the coefficient of $(\alpha/(\alpha-1))^{(n-1)/2}$ is identically zero. This implies that 1 is the only one real root of this polynomial; setting $\alpha/(\alpha-1) = 1$ yields $\alpha = 1/2$, with corresponding optimal winning probability

$$\mathbf{Pr}_{\mathcal{A}}(S_0 \leq \delta \text{ and } S_1 \leq \delta)$$

$$= \delta^n \sum_{\mathbf{b} \in \{0,1\}^n} \Big( \frac{1}{|\mathbf{b}|!} \sum_{0 \leq l \leq |\mathbf{b}|, l < \delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot$$

$$\frac{1}{(n-|\mathbf{b}|)!} \sum_{0 \leq l \leq n-|\mathbf{b}|, l < \delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{n-|\mathbf{b}|} \cdot \left(\frac{1}{2}\right)^n$$

$$= \frac{1}{n!} \left(\frac{\delta}{2}\right)^n \sum_{\mathbf{b} \in \{0,1\}^n} \Big( \binom{n}{|\mathbf{b}|} \sum_{0 \leq l \leq |\mathbf{b}|, l < \delta} (-1)^l \binom{|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{|\mathbf{b}|} \cdot$$

$$\sum_{0 \leq l \leq n-|\mathbf{b}|, l < \delta} (-1)^l \binom{n-|\mathbf{b}|}{l} \left(1 - \frac{l}{\delta}\right)^{n-|\mathbf{b}|} \Big)$$

$$= \frac{1}{n!} \left(\frac{\delta}{2}\right)^n \sum_{r=0}^{n} \Big( \binom{n}{r}^2 \sum_{0 \leq l \leq r, l < \delta} (-1)^l \binom{r}{l} \left(1 - \frac{l}{\delta}\right)^r \cdot$$

$$\sum_{0 \leq l \leq n-r, l < \delta} (-1)^l \binom{n-r}{l} \left(1 - \frac{l}{\delta}\right)^{n-r} \Big),$$

as needed.

Theorem 2 implies that for any integer $n$, the optimal winning probability of an oblivious algorithm is computable in exponential time.

# 4 Non-oblivious Algorithms

## 4.1 The Winning Probability

**Theorem 3.** *Assume that $\mathcal{A}$ is any randomized non-oblivious algorithm. Then,*

$$\mathbf{Pr}_{\mathcal{A}}(S_0 \leq \delta \text{ and } S_1 \leq \delta)$$
$$= \sum_{\mathbf{b}\in\{0,1\}^n} \Bigg( \frac{1}{(n-|\mathbf{b}|)!} \sum_{I\subseteq\{i:b_i=0\}, \sum_{l\in I}\beta_l<\delta} (-1)^{|I|} \left(\delta - \sum_{l\in I}\beta_l\right)^{n-|\mathbf{b}|} \cdot$$
$$\left( \prod_{l=1}^{|\mathbf{b}|}(1-\beta_l) - \frac{1}{|\mathbf{b}|!} \sum_{\substack{I\subseteq\{i:b_i=1\},\\ |I|-\sum_{l\in I}\beta_l<|\mathbf{b}|-\delta}} (-1)^{|I|} \left(|\mathbf{b}| - \delta - |I| + \sum_{l\in I}\beta_l\right)^{|\mathbf{b}|} \right) \Bigg).$$

## 4.2 Optimality Conditions

For non-oblivious algorithms, the analysis is more involved since it must take into account the conditional probabilities "created" by the knowledge of inputs by the agents. We show:

**Theorem 4 (Optimality conditions for non-oblivious algorithms).** *Assume that $\mathcal{A}$ is an optimal, randomized non-oblivious algorithm. Then, for any index $k$,*

$$\sum_{|\mathbf{b}|=0}^{n} \binom{n-1}{|\mathbf{b}|} \Bigg( \frac{1}{(n-1-|\mathbf{b}|)!} \sum_{\substack{0\leq l\leq n-1-|\mathbf{b}|,\\ \delta-\beta l>0}} (-1)^l \binom{n-1-|\mathbf{b}|}{l} (\delta-\beta l)^{n-1-|\mathbf{b}|} \Bigg) \cdot$$
$$\Bigg( -(1-\beta)^{|\mathbf{b}|} - \frac{(|\mathbf{b}|+1)}{(|\mathbf{b}|+1)!} \sum_{\substack{1\leq l\leq |\mathbf{b}|+1,\\ |\mathbf{b}|+1-\delta-l+\beta l>0}} (-1)^l \binom{|\mathbf{b}|}{l-1} l\,(\mathbf{b}+1-\delta-l+\beta l)^{|\mathbf{b}|} \Bigg) +$$
$$\sum_{|\mathbf{b}|=0}^{n-1} \binom{n-1}{|\mathbf{b}|} \Bigg( \Big( (1-\beta)^{|\mathbf{b}|} - \Big( \frac{1}{|\mathbf{b}|!} \sum_{\substack{0\leq l\leq |\mathbf{b}|,\\ |\mathbf{b}|-\delta-l+\beta l>0}} (-1)^l \binom{|\mathbf{b}|}{l} (|\mathbf{b}|-\delta-l+\beta l)^{|\mathbf{b}|} \Big) \Big) \cdot$$
$$\frac{-(n-|\mathbf{b}|)}{(n-|\mathbf{b}|)!} \sum_{1\leq l\leq n-|\mathbf{b}|, \delta-\beta l>0} (-1)^l \binom{n-|\mathbf{b}|-1}{l-1} l\,(\delta-\beta l)^{n-|\mathbf{b}|-1} \Bigg)$$
$$= 0\,.$$

Unfortunately, the conditions in Theorem 4 do not admit a uniform solution (independent of $n$). We discover that the solutions for $n = 3$ and $n = 4$ are different. The solution for $n = 3$ and $\delta = 1$ satisfies the polynomial equation $\beta^2 - 2\beta + 6/7 = 0$; the solution is calculated to be equal to $1 - \sqrt{1/7} = 0.622$, which is the threshold value conjectured by Papadimitriou and Yannakakis in [9] to imply optimality for the same case. On the other hand, the solution for $n = 4$ and $\delta = 4/3$ satisfies the polynomial equation $-(26/3)\beta^3+(98/3)\beta^2-(368/9)\beta+416/27 = 0$; the solution is calculated to be equal to approximately 0.678.

## References


1. K. Arrow, *The Economics of Information,* Harvard University Press, 1984.
2. G. Brightwell, T. J. Ott, and P. Winkler, "Target Shooting with Programmed Random Variables," *Proceedings of the 24th Annual ACM Symposium on Theory of Computing,* pp. 691–698, May 1992.
3. P. Fizzano, D. Karger, C. Stein and J. Wein, "Job Scheduling in Rings," *Proceedings of the 6th Annual ACM Symposium on Parallel Algorithms and Architectures,* pp. 210–219, June 1994.
4. X. Deng and C. H. Papadimitriou, "Competitive, Distributed Decision-Making," *Proceedings of the 12th IFIP Congress,* pp. 350–356, 1992.
5. J. M. Hayman, A. A. Lazar, and G. Pacifici, "Joint Scheduling and Admission Control for ATM-Based Switching Nodes," *Proceedings of the ACM SIGCOMM,* pp. 223–234, 1992.
6. P. C. Kanellakis and C. H. Papadimitriou, "The Complexity of Distributed Concurrency Control," *SIAM Journal on Computing,* Vol. 14, No. 1, pp. 52–75, February 1985.
7. E. Kushilevitz and N. Nisan, *Communication Complexity,* Cambridge University Press, Cambridge, 1996.
8. C. H. Papadimitriou, "Computational Aspects of Organization Theory," *Proceedings of the 4th Annual European Symposium on Algorithms,* September 1996.
9. C. H. Papadimitriou and M. Yannakakis, "On the Value of Information in Distributed Decision-Making," *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing,* pp. 61–64, August 1991.
10. C. H. Papadimitriou and M. Yannakakis, "Linear Programming Without the Matrix," *Proceedings of the 25th Annual ACM Symposium on Theory of Computing,* pp. 121–129, May 1993.
11. C. Polychronopoulos and D. Kuck, "Guided Self-Scheduling: A Practical Scheduling Scheme for Parallel Computers," *IEEE Transactions on Computers,* Vol. 12, pp. 1425–1439, 1987.
12. R. P. Stanley, *Enumerative Combinatorics: Volume 1,* The Wadsworth & Brooks/Cole Mathematics Series, 1986.