Occupancy urn models in the analysis of algorithms

Danièle GARDY

PRISM, UMR 8636 CNRS and Université de Versailles Saint-Quentin

78035 Versailles Cedex, France.

Abstract

We survey some problems that appear in the analysis of different problems in Computer Science, and show that they can be cast in a common framework (occupancy urn models) and admit a uniform treatment.

1 Introduction

Although data structures such as trees and graphs are ubiquitous in Computer Science, and may well be the most frequent models in the analysis of data structures and algorithms, a small but interesting number of problems relative to random allocations can be cast in a common discrete probabilistic framework, known as urn models. Roughly speaking, we have a certain number of urns, into which we throw balls (we may be allowed to remove them), and we are interested in some parameter of the model, such as the total number of balls, or the fraction of urns satisfying some property. When we have a single urn and balls of different colors, that we may draw from or add to the urn, we have variations on the so-called Polya urn model. Such models have proved useful for analyzing balanced trees by fringe analysis; see for example a paper by Aldous et al. applied to several tree models [1], or a recent work by Mahmoud on random binary search trees [24]. We concentrate here on so-called "occupancy" urn models, where we have a sequence of urns and throw balls at random into them; often the parameter under study is the number of urns satisfying some property.

We do not claim to make a complete survey of all appearances of occupancy urn models in analysis of algorithms, and we shall mainly concentrate on our former work; our goal in this paper is to show that all the problems we present (and probably many others) share a common probabilistic description and can be dealt with in a unified manner, by using the tools of the analysis of algorithms. We refer the reader to [9, 10] for a general introduction to these tools, from a generating function description of our random allocations to asymptotic analysis and limiting probability distributions.

We present in Section 2 a tentative classification of occupancy urn models and recall results on the basic versions of these models, then show how a general approach by generating functions might prove useful. Sections 3 to 5 then give some examples drawn from Computer Science, and show how our general framework applies to them.

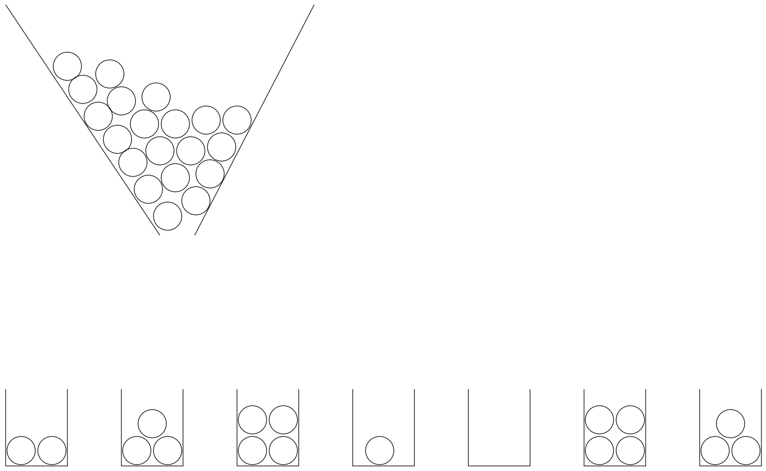

Figure 1: Allocation of balls into urns

2 Occupancy urn models

2.1 The different problems

Many of the problems related to the occupancy of a sequence of urns can be roughly classified into one of the following three types :

- *Static* problems : We throw a given number of balls into the sequence of urns, and look at the final configuration, characterized by the probability distribution of some random variable (often the number of urns satisfying some property $\mathcal{P}$);
- *Waiting time* models : We throw the balls one by one, and wait for the first appearance of a specified configuration (a specified number of urns satisfying a property $\mathcal{P}$);
- *Dynamic* models : We throw the balls one by one, and consider the sequence of configurations.[1]

The occupancy models share a common framework : We have a sequence (sometimes a set) of urns, into which balls are thrown according to some rules (Fig 1). The models differ according to the number of balls thrown at each trial, and to the possible evolution of each urn. The urns and the balls may be distinguishable, or not; the number of balls that an urn may receive is usually unbounded, but may be finite; when the urns are distinguishable (we are dealing with a sequence of urns) they may have uniform or non uniform probabilities to receive a ball, etc.

The basic reference is the book by Johnson and Kotz [20], augmented by the recent survey of Kotz and Balakrishnan [22] (which, however, deals mostly with variations on the Polya model). The book by Kolchin et al. [21] presents detailed studies about the number of empty urns, and also about the number of urns with a given number of balls, in the classical model.

General references also include Chapter 8, "Words and maps", of the book [9] by Flajolet and Sedgewick, which deals with string and urn models. The report [7] is an introduction to urn models, oriented towards the use of symbolic tools (such as Maple); it presents many examples and gives a good classification of major models.

[1] A variation on this appears when we accept the withdrawal of a ball already present.

Some notations : Throughout the paper, we shall use m for the number of urns (which we assume is finite), and n for the number of balls, when appropriate. When speaking about asymptotic properties, we shall mostly consider the *central domain,* i.e. the domain where the (final) number of balls and the number of urns are large and proportional.

2.2 The classical occupancy model

A model that has been the basis of many studies appears when the urns are distinguishable and the balls are undistinguishable. Results have been obtained for the number of urns having a specified number of balls (most notably for the number of empty urns), both in the static and dynamic cases. The waiting time version, when we consider the number of urns with at least two (k) balls, is closely related to the classical birthday problem : Urns represent dates; what is the number of balls that must be thrown to get for the first time an urn with at least two (k) balls, i.e. a shared anniversary (of order k)?

Hashing tables can be seen as occupancy urn models in a simple way : each address is associated with an urn, and each key with a ball; the hashing function that maps keys to addresses is equivalent to throwing a ball at random into one of the urns. The classical parameters of hashing translate into urn problems; for example the first time that an urn receives two (k) balls is the time of the first collision (of order k), the number of empty urns is the number of addresses without keys, the number of keys in an urn is the number of keys that hash to this address, and the maximum number of keys that hash to a given address is the maximum occupancy of an urn.

2.3 The number of empty urns and the Coupon Collector's problem

Before turning to more complex urn models, we shall return to the number of empty urns, and show how the use of generating functions may unify the treatment of the static and waiting-time problems. References are [21] for a detailed study of the number of empty urns; for the Coupon Collector's problem, we shall refer to [8], which gives a presentation using generating functions, and allowing for easy generalization to non equiprobable coupons.

In the Coupon Collector's problem, coupons are drawn with replacement from a finite set, and we are interested in the number of coupons one must draw to get for the first time at least one (k) occurrence of each coupon, or a set of j distinct (unspecified) coupons. The related occupancy model is simple : A candidate coupon is an urn, and drawing a coupon is simply the allocation of a ball to an urn. The waiting time for the first occurence of a set of j distinct coupons is the number of balls one must throw to get for the first time j non-empty urns.

All the information that allows us to treat the static and waiting time problems is encoded into the generating function describing the allocations, where the variables x and y mark respectively the non-empty urns[2] and the total number of balls : Define p_i as the probability of the i-th urn, or coupon; then the probability generating function is

$$F(x,y) = \prod_{i=1}^{m} \left(1 - x + x e^{p_i y}\right).$$

[2]When we know the total number of urns, studying the number of empty urns, or the number of non-empty urns, are equivalent problems.

The static model amounts to studying the distribution of the random variable X equal to the number of occupied urns, for a fixed number of balls n. The probability generating function of X is $[y^n]F(x,y)/[y^n]F(1,y)$; in the uniform case ($p_i = 1/m$), it may be simpler to use the enumerating function $(1-x+xe^y)^m$. From whatever function F we have decided to use, we can get probabilities, exact and asymptotic formulae for all the moments, and study the limiting distribution when n and m grow large. For example, in the uniform case the r.v. X, suitably normalized, converges to a Gaussian distribution in the central domain; in the *left-hand domain* ($me^{-n/m}$ has a finite or null limit) it converges to a Poisson distribution, and in the *right-hand domain* (n^2/m has a finite limit) the r.v. $X-(m-n)$ has for limit a Poisson distribution [20, p. 318-320]. It is also possible to get asymptotic results for non-uniform distributions on the urns; see for example [20, p. 321].

For the Coupon Collector's problem, the average waiting time to obtain for the first time a collection of j distinct coupons can be expressed in terms of the probability generating function $F(x,y)$:

$$\sum_{q=0}^{j-1} \int_0^{+\infty} [x^q]F(x,y)e^{-y}dy,$$

and the average waiting time for a full collection is $\int_0^{+\infty}(1-F(1,y))e^{-y}dy$. In the uniform case, we find back the well-known expression $\int_0^{+\infty}(1-(1-e^{y/m})^m)dy$.

Generating functions can also be the basis for the study of the dynamic case; see [21, Ch. 4] for a systematic treatment.

2.4 A general approach for the static and dynamic cases

We give here the general ideas that underlie asymptotic results, mostly in the central domain. We shall formulate them for a random variable X_m equal to the number of urns satisfying some property $\mathcal{P}$, but we believe that they may be adapted to other models, for example to the join models presented in Section 3.2.

The most important assumption is the independence of the urns : The state of an urn is independent of the state of the other urns, at least as long as the number of balls is not fixed (when the number of balls is fixed, we are dealing with weakly correlated random variables). Hence the bivariate function associated to the static case, marking the balls and the urns satisfying $\mathcal{P}$, is the m-th power of a simpler function, and is often entire. From it, we can extract an asymptotic expression for the probability generating function of the r.v. X, most often by a saddle-point approximation, and general theorems ensure that the limiting distribution in (a suitable equivalent of) the central domain exists and is Gaussian (see for example results by Drmota [5], Bender and Richmond [2], or the "quasi-powers" framework of Hwang [19]) .

Following the ideas presented by Kolchin et al. [21] and used for example by Louchard in [23], we can sketch an analytic method for studying the asymptotic process $X_m(n)$, when the balls are thrown one at a time :

1. Check that the limiting distribution of $X_m(n)$ at a given time n, i.e. for a given (large!) number n of balls, is asymptotically Gaussian in some range (usually the central domain);

2. Suitably normalize the sequence of random variables and the time intervall;

3. Check that finite-dimensional distributions are Gaussian;

4. Obtain candidate covariance for a limiting process $X(t)$;

5. Introduce the random variable equal to the number of urns whose state has changed in some interval; get a bound on some moment, then use a probabilistic result (see for example [17, p. 514]) to conclude to tightness of the sequence of the (normalized) random variables X_m and to convergence towards a Gaussian process.

As was the case for point 1, points 3 to 5 can be attacked by a generating function approach, due to the fact that these functions are large powers of simpler functions. However, actual computations for a given model often turn out to be quite involved. The paper [6] gives some general conditions under which the above approach holds, as well as several examples.

3 Database problems

3.1 Yao's formula

More than twenty years ago, Yao [26] gave a simple formula for the expected number of blocks to be retrieved, assuming that one wants to access a given number of items. We present here a sketch of the modelization by urns and balls, and refer an interested reader to [14] for detailed results and applications.

Consider a set $\mathcal{E}$ of p items, whose representation in memory requires a specified number m of pages (each page contains $b = p/m$ objects). Assume that we want to access n items, which are distributed at random among the m pages; how many pages (blocks) must be read in order to get all the desired items?

The number of blocks to be read is a random variable X with integer values. Some time before Yao, Cardenas [4] (without mentioning it) gave a formula for the expectation of X that comes straight from the classical empty urns model, where urns have unbounded capacity. However, this assumption is unrealistic from a database point of view. Yao obtained the expectation of X by elementary computations :

$$E[X] = m\left(1 - \frac{\binom{p-b}{n}}{\binom{p}{n}}\right).$$

By introducing an urn model which is a variation of the empty urns model, where the size of the urns is finite and equal to b, we were able to do a detailed probabilistic analysis, and to obtain the limiting distribution on X. The translation from databases to urns and balls is as follows : A block of size b is an urn of capacity b; choosing an item to be retrieved is equivalent to throwing a ball at random into one of the urns; and the number of blocks that must be read is the number of non-empty urns.

We recall that X is the random variable *number of selected blocks, or of non empty urns.* The generating function enumerating the set of possible allocations of balls into urns is

$$F(x, y) = \left(1 + x((1+y)^b - 1)\right)^m.$$

From this function, we can derive exact expressions for all moments, involving binomial coefficients; for example the average value $E[X]$ is equal to $[y^n]\partial F/\partial x(1, y)/[y^n]F(1, y)$; thus it is simple to obtain asymptotic values in terms of n and m. In the central domain, the limiting distribution of the random variable X exists and is Gaussian.

The modelization by an urn model makes it comparatively easy to generalize Yao's formula to at least some classes of non uniform distributions on urns, which is relevant in some problems appearing for example in object databases [14].

3.2 Evaluation of sizes of derived relation

When relational databases came into use, it was quickly apparent that they needed efficient query optimizers, which lead to questions about choosing an execution plan; to be able to make such a choice, one of the many tools used was evaluation of statistical parameters of the databases, the so-called "statistical profile" [25]. The sizes of intermediate results, i.e. of the sets of objects that are obtained by applying the operators of whatever query language is used, are among the parameters that appear in a statistical profile. When using the relational algebra, it has thus become relevant to estimate the sizes of relations obtained by relational operators, most notably by a projection or a join.

We recall that relations are basically tables with several columns and without duplicates; the projection of a relation is obtained by suppressing some columns and removing the duplicates; the join of two relations on a common column (attribute) is obtained from the cartesian product of the two relations, by keeping only those lines that agree on the shared attribute.

We proposed some years ago a parametric model to this effect, that uses a variation of the empty urns model for the projection, and that introduces new urn models for joins. We refer to [15, 16] for an introduction to the models, dealing with the database aspects, and to [11, 12] for a detailed presentation and for asymptotic results, and sum up the main points below.

For the *projection,* we basically have to deal with an empty urns model when the urns are either infinite, as in the classical case, or bounded, i.e. can receive at most δ balls; the choice depends on the underlying database dependencies. The parameter under study, which is the size of the projection, is equal to the number of non-empty urns. Of course the case of bounded urns is equivalent to Yao's problem. For the *(equi)join,* we have a model with different types of balls (see Fig 2) : We throw balls of two different colors (red and blue) into urns, which are either infinite or bounded, according to the underlying data dependencies, then put green balls into each urn that contains both red and blue balls; the number of green balls in each urn is the product of the number of red and blue balls in this urn. The *semijoin,* which is the composition of a projection and an equijoin, can be similarly modelized (Fig. 3) : After throwing red and blue balls, we consider again urns that contain balls of the two colors, and we put as many green balls as the urn contains red balls. In both cases, the size of the (equi or semi) join is equal to the cumulated number of green balls in all the urns.

For these models, a generating function description has allowed us to prove the existence of a Gaussian limiting distribution in a suitably defined "central domain". The generating function for the projection has the general form

$$F(x, y) = (1 - x + x\lambda(y))^m,$$

where x marks the projection size and y the initial relation size, and where $\lambda(y)$ is the generating function describing the allocation of balls into a single urn : For infinite urns, $\lambda(y) = e^y$, but urns of size δ (and distinguishable places in an urn) give $\lambda(y) = (1 + y)^\delta$; other functions are possible. For the equijoin, if the functions describing the allocations of red and blue balls into an urn are $\lambda_1(y) = \sum_k a_k y^k$ and $\lambda_2(y) = \sum_k b_k y^k$, the function describing the allocations, with x marking the green balls, y and z the red and blue balls, is

$$\left(\sum_{k,l} a_k b_l x^{kl} y^k z^l \right)^m .$$

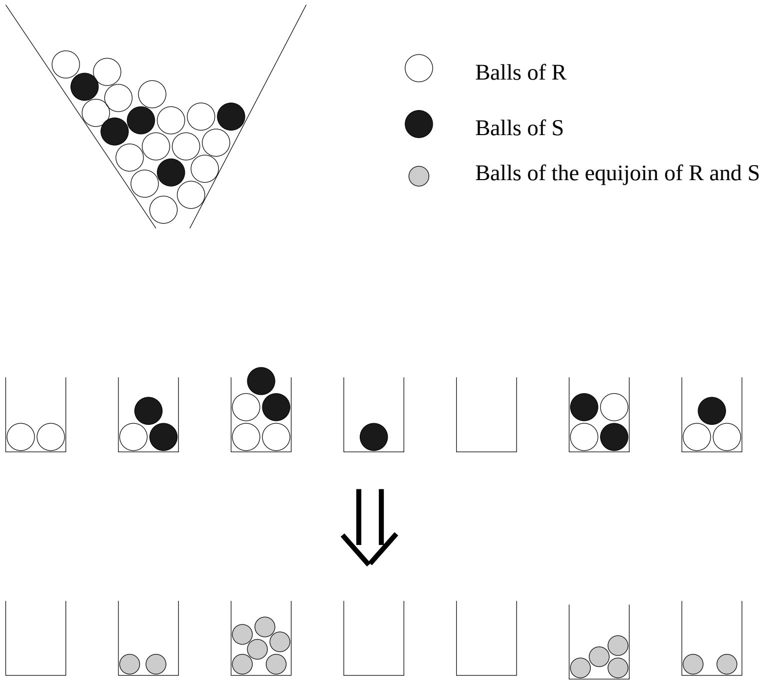

Figure 2: Allocation of balls for the equijoin

The function associated with the semijoin is simpler :

$$(\lambda_1(y) + \lambda_1(xy)(\lambda_2(z) - 1))^m .$$

It is also possible to obtain dynamic results; in [13] we considered a case that is a generalization of the one presented above, in which we allowed for withdrawal of balls (corresponding to deletion of some items on the database). The resulting process is asymptotically Gaussian, with a covariance that can be explicitely computed.

4 Balanced urns

When studying a problem from Learning Theory, we came to a model that involves balls of two colors, where the state of an urn is related to its *balance,* i.e. to the difference between the numbers of balls of each color. We refer the reader to [3] for a detailed presentation and give below a brief summary of the problem and of its modelization.

Let us consider the following (very) simple problem : Assume that we want to learn some boolean function on p boolean variables; the domain of this function has for cardinality $m = 2^p$. Associate an urn to each initial configuration of the variables, and a ball (labelled *true* or *false*) for the result of each trial. A sequence of n trials is thus represented by the allocation of n balls into the urns. Now assume that, for each trial, there is a (small) probability that we won't get the correct answer, but a wrong one. We shall thus have two types of balls, "good" and "bad" ones. If there are not too many wrong answers for a given configuration, we might still be able to decide, by a majority argument, what the correct value of the function for this configuration should be. But if, for some urn, there are too many wrong answers, then the value of the function for the corresponding configuration of the p boolean variables will be wrong. Thus we see that an important factor in evaluating the performance of this learning method is the number of urns for which we shall learn a wrong value, i.e. of urns that have a majority of wrong answers.

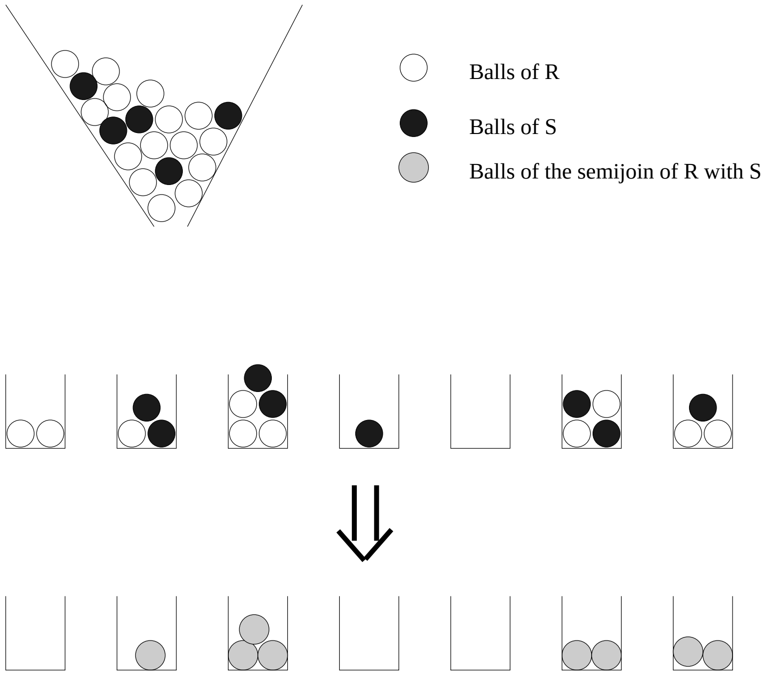

Figure 3: Allocation of balls for the semijoin

More generally, we shall be interested in the number of urns having an equal number of good and bad balls, a specified relative excedent (balance) of good balls, or a positive balance (see Fig. 4). For example, when good and bad balls have the same probability, the generating function marking the urns of balance q by x and the total number of balls by y can be expressed using the Bessel coefficients $I_q(t) = \sum_r (t/2)^{q+2r}/r!(q+r)!$ (if desired, we might as well mark separately the balls of each type; we also assume that the urns can receive an unbounded number of balls; extensions to bounded urns are possible) :

$$F(x, y) = (e^y + (x-1)I_q(y))^m.$$

If good and bad balls have different probabilities μ and $1-\mu$, the probability generating function is

$$F(x, y) = \left(e^y + (x-1)\sqrt{\frac{1-\mu}{\mu}} I_q(2\sqrt{\mu(1-\mu)}y)\right)^m.$$

Again, we have a limiting Gaussian distribution in the central domain, of known expectation and variance; when throwing the balls one at a time, the asymptotic process is Gaussian.

5 Conclusion

We have tried to present in a unified manner some problems involving occupancy urn models. We do not claim that our approach covers all interesting appearances of these models. For example, in static models we always assume that the number of balls is known; another approach might consider that this number is a random variable, following a given distribution (often Poisson) [18].

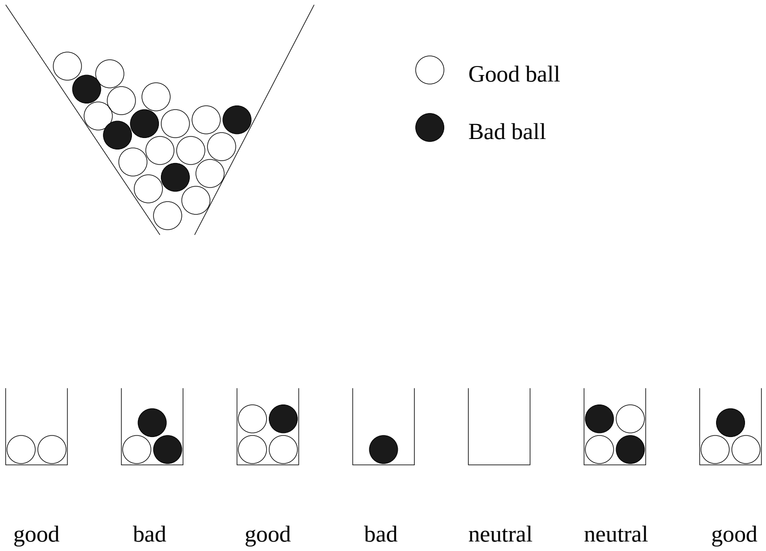

Figure 4: Allocation of balls of two types and corresponding urn types

References

[1] D.J. ALDOUS, B. FLANNERY, and J. PALACIOS. Two applications of urn processes. *Probab. Engineering Inform. Sci.*, 2:293–307, 1988.

[2] E.A. BENDER and L.B. RICHMOND. Central and local limit theorems applied to asymptotic enumeration II : multivariate generating functions. *Journal of Combinatorial Theory, series A*, 34:255–265, 1983.

[3] S. BOUCHERON and D. GARDY. An urn model from learning theory. *Random Structures and Algorithms*, 10(1-2):43–67, January 1997. Special issue on the Analysis of Algorithms.

[4] A.F. CARDENAS. Analysis and performance of inverted data base structures. *Comm. ACM*, 18(5):253–263, 1975.

[5] M. DRMOTA. A bivariate asymptotic expansion of coefficients of powers of generating functions. *European Journal of Combinatorics*, 15:139–152, 1994.

[6] M. DRMOTA, D. GARDY, and B. GITTENBERGER. A unified presentation of some urn models. Technical Report 1999-2, Laboratoire PRISM, University of Versailles-St Quentin, January 1999.

[7] P. FLAJOLET. Balls and urns, etc. Technical report, INRIA Rocquencourt (France), 1996. Studies in automatic combinatorics, Vol. I; Link to *http://pauillac.inria.fr/algo/libraries/autocomb/*.

[8] P. FLAJOLET, D. GARDY, and L. THIMONIER. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 39:207–229, 1992.

[9] P. FLAJOLET and R. SEDGEWICK. *An introduction to the analysis of algorithms*. Addison-Wesley, 1996.

[10] P. FLAJOLET and R. SEDGEWICK. The average-case analysis of algorithms : multivariate asymptotics and limit distributions. Technical Report 3162, INRIA, May 1997.

[11] D. GARDY. Normal limiting distributions for projection and semijoin sizes. *SIAM Journal on Discrete Mathematics*, 5(2):219–248, 1992.

[12] D. GARDY. Join sizes, urn models and normal limiting distributions. *Theoretical Computer Science (A)*, 131:375–414, August 1994.

[13] D. GARDY and G. LOUCHARD. Dynamic analysis of some relational data bases parameters. *Theoretical Computer Science (A)*, 144(1-2):125–159, June 1995. Special volume on Mathematical Analysis of Algorithms.

[14] D. GARDY and L. NEMIROVSKI. Urn models and Yao's formula. In *International Conference on Database Theory,* C. Beeri and P. Buneman (editors), pages 100–112, Jerusalem (Israel), January 1999. Lecture Notes in Computer Science, Springer Verlag, no 1540.

[15] D. GARDY and C. PUECH. On the sizes of projections : a generating function approach. *Information Systems*, 9(3/4):231–235, 1984.

[16] D. GARDY and C. PUECH. On the effect of join operations on relation sizes. *ACM Transactions On Database Systems*, 14(4):574–603, December 1989.

[17] I. GUIKHMAN and A. SKOROKHOD. *Introduction à la théorie des processus aléatoires.* Editions MIR, Moscou, 1980.

[18] L. HOLST. A unified approach to limit theorems for urn models. *Journal of Applied Probability*, 16:154–162, 1979.

[19] H.K. HWANG. On convergence rates in the central limit theorems for combinatorial structures. *European Journal of Combinatorics*, 19:329–343, 1998.

[20] N.L. JOHNSON and S. KOTZ. *Urn models and their application.* Wiley & Sons, 1977.

[21] V. KOLCHIN, B. SEVAST'YANOV, and V. CHISTYAKOV. *Random Allocations.* Wiley & Sons, 1978.

[22] S. KOTZ and N. BALAKRISHNAN. Advances in urn models during the past two decades. In *Advances in combinatorial methods and applications to probability and statistics*, pages 203–257, 1997.

[23] G. LOUCHARD. Trie size in a dynamic list structure. *Random Structures and Algorithms*, 5(5):665–702, 1994.

[24] H. M. MAHMOUD. On rotations in fringe-balanced binary trees. *Information Processing Letters*, 65:41–46, 1998.

[25] M. V. MANNINO, P. CHU, and T. SAGER. Statistical profile estimation in database systems. *ACM Computing Surveys*, 20(3):191–221, September 1988.

[26] S.B. YAO. Approximating block accesses in data base organizations. *Comm. ACM*, 20(4), 1977.