HAL
open science

Bipartite Random Graphs and Cuckoo Hashing

Reinhard Kutzelnigg

To cite this version:

Reinhard Kutzelnigg. Bipartite Random Graphs and Cuckoo Hashing. Fourth Colloquium on Mathematics and Computer Science Algorithms, Trees, Combinatorics and Probabilities, 2006, Nancy, France. pp.403-406, 10.46298/dmtcs.3486 . hal-01184689

HAL Id: hal-01184689

https://inria.hal.science/hal-01184689v1

Submitted on 17 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bipartite Random Graphs and Cuckoo Hashing

Reinhard Kutzelnigg†

Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Vienna, Austria

The aim of this paper is to extend the analysis of Cuckoo Hashing of Devroye and Morin in 2003. In particular we make several asymptotic results much more precise. We show, that the probability that the construction of a hash table succeeds, is asymptotically $1 - c(\varepsilon)/m + O(1/m^2)$ for some explicit $c(\varepsilon)$, where m denotes the size of each of the two tables, $n = m(1-\varepsilon)$ is the number of keys and $\varepsilon \in (0,1)$. The analysis rests on a generating function approach to the so called Cuckoo Graph, a random bipartite graph. We apply a double saddle point method to obtain asymptotic results covering tree sizes, the number of cycles and the probability that no complex component occurs.

Keywords: hashing, random bipartite graphs, generating functions, double saddle point method

1 Introduction

Cuckoo Hashing is a hashing algorithm with the advantage of constant worst case search time, contrary to standard hashing algorithms as Open Addressing or Hashing with Chaining. (See Knuth (1973) for surveys on hashing.) The algorithm was introduced by Pagh and Rodler (2001) and a further analysis was done by Devroye and Morin (2003).

The main idea of Cuckoo Hashing is to split up the available amount of memory. Therefore, it uses two tables of size m, and two independent hash functions h_1 and h_2, to store the n data points. Every key x might be stored at exactly two positons, namely $h_1(x)$ and $h_2(x)$. So, one has to inspect at most two positions during the search for x. Further, we allow at most one element to be stored at any position. It may of course still happen, that both possible storage places of a given key x are already occupied. We solve this problem imitating the nesting habits of the cuckoo, and allow x to throw out the key (say y) occupying the storage position. Next, we insert the y at its alternative position, which is also possibly occupied. We repeat this steps, until we find an empty position or conjecture that we have entered an endless loop. In the the latter case, we have to choose new hash functions and rebuild the data structure.

Let $x_1, x_2, \ldots, x_n$ denote the the keys. We will assume from now on, that the sequences of hash values $(\langle h_1(x_i), h_2(x_i)\rangle)_{i=1\ldots n}$ are independent uniform random integers drawn from $\{1, 2, \ldots, m\}$. Further, if a rehash is necessary, we assume that the new hash values are independent from previous attempts.
Consider the bipartite graph on $\{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with n unlabeled and undirected edges, where each edge connects the two possible storage locations of a fixed key. Note that repeated vertices are possible. This graph is called the Cuckoo Graph. The shape of it determines, whether the construction of the data structure is successful or not, since we enter an endless loop iff there is a complex component (a connected component with more than one cycle). Recall that a complex component having k nodes and $k + l$ edges (with $l \geq 1$) corresponds to the attempt to put $k + l$ keys in only k locations.

Because of their close relation, we can obtain results on the expected behaviour of Cuckoo Hashing by an average case analysis of properties of random bipartite graphs. A fixed graph without multiple edges is generated by $n!$ series of hash values $(\langle h_1(i), h_2(i)\rangle)_{i=1\ldots n}$. There are less possibilities, if the graph possesses at least one multiple edge. That's why, we assign a compensation factor

$$\kappa(G) = \left(\prod_{i=1}^{m_1}\prod_{j=1}^{m_2} a_{ij}!\right)^{-1}, \tag{1}$$

†This work was motivated by Luc Devroye and Michael Drmota and is also partly a joint work with them.
The author was supported by the Austrian Science Foundation FWF, project S9604, that is part of the Austrian National Research Network "Analytic Combinatorics and Probabilistic Number Theory".

1365–8050 © 2006 Discrete Mathematics and Theoretical Computer Science (DMTCS), Nancy, France

to a graph G, where a_{ij} is the number of undirected edges connecting the vertices i and j. This is similar to the multigraph process in Janson et al. (1993). So, $\kappa(G)n!$ is the number of different sequences of pairs of hash values, which produce the same Cuckoo Graph G. Henceforth we use the symbol $\#_w$ when we count using the weighting factor. So we achieve

$$\#_w G_{m,m,n} = [v^n]\left(1 + v + \frac{v^2}{2!} + \frac{v^3}{3!} + \cdots\right)^{m^2} = \frac{m^{2n}}{n!} \tag{2}$$

for the weighted number of all Cuckoo Graphs posessing two times m vertices and n edges.

Section 3 establishes a generating function model of random bipartite graphs without complex components and Section 4 presents the methods to obtain an asymptotic expansion of the coefficients. The results are presented in Section 2. We only consider the case with $m(1-\varepsilon)$ keys, where $\varepsilon \in (0,1)$.

2 Results

Theorem 1 *The probability that a Cuckoo Hash of $n = (1-\varepsilon)m$ data points into two tables of size m succeeds (i.e. the corresponding Cuckoo Graph contains no complex component), provided that $\varepsilon \in (0,1)$, is equal to*

$$1 - \frac{(2\varepsilon^2 - 5\varepsilon + 5)(1-\varepsilon)^3}{12(2-\varepsilon)^2\varepsilon^3}\frac{1}{m} + O\left(\frac{1}{m^2}\right). \tag{3}$$

Theorem 1 of Kalugin (1991) already states that the amount of graphs containing complex components tends to zero, but it does not provide an asymptotic approximation. Lemma 2 of Devroye and Morin (2003) claims the bound $1 + O(1/m)$, but the proof is incomplete.

Theorem 2 *Let m and ε be as in Theorem 1. The number of unicyclic components with cycle length $2k$ has in limit a Poisson distribution $Po(\lambda_k)$ with parameter*

$$\lambda_k = \frac{1}{2k}(1-\varepsilon)^{2k}. \tag{4}$$

Further, the number of tree components with l vertices converges in distribution to a Gaussian random variable with asymptotic mean

$$\mu = 2m\frac{l^{l-2}(1-\varepsilon)^{l-1}e^{l(\varepsilon-1)}}{l!}, \tag{5}$$

and variance

$$\sigma^2 = \mu - 2m\frac{e^{2l(\varepsilon-1)}l^{2l-4}(1-\varepsilon)^{2l-3}(l^2\varepsilon^2 + l^2\varepsilon - 4l\varepsilon + 2)}{(l!)^2}. \tag{6}$$

The first statement is closely related to Theorem 2 of Kalugin (1991).

Finally, we give an upper bound for the running time.

Theorem 3 *Under the assumptions of Theorem 1, the expected construction time of a successful Cuckoo Hash attempt is bounded above by*

$$2n\frac{1-e^{\varepsilon-1}}{1+\varepsilon-e^{\varepsilon-1}}. \tag{7}$$

3 Generating Functions

We call a tree bipartite if the vertices are partitioned into two classes V_1 and V_2 such that no node has a neighbour of the same class. Note that we are considering labeled trees, that is nodes in V_1 are labeled by $1, 2, \ldots, |V_1|$ and the nodes in V_2 are labeled by $1, 2, \ldots, |V_2|$.

Let T_1 denote the set of bipartite rooted trees, where the root is contained in V_1, T_2 the set of bipartite rooted trees, where the root is contained in V_2, and $\tilde{T}$ the class of unrooted bipartite trees. Furthermore, let t_{1,m_1,m_2} resp. t_{1,m_1,m_2} denote the number of trees in T_1 resp. T_2 with m_1 nodes of type of type 1 and m_2 of type 2. Similarly we define $\tilde{t}_{m_1,m_2}$. The corresponding generating functions are defined by

$$t_1(x,y) = \sum_{m_1,m_2\geq 0} t_{1,m_1,m_2}\frac{x^{m_1}}{m_1!}\frac{y^{m_2}}{m_2!}, \quad t_2(x,y) = \sum_{m_1,m_2\geq 0} t_{2,m_1,m_2}\frac{x^{m_1}}{m_1!}\frac{y^{m_2}}{m_2!}, \tag{8}$$

and

$$\tilde{t}(x,y) = \sum_{m_1,m_2\geq 0} \tilde{t}_{m_1,m_2}\frac{x^{m_1}}{m_1!}\frac{y^{m_2}}{m_2!}. \tag{9}$$

We obtain equations $t_1(x,y) = xe^{t_2(x,y)}$, $t_2(x,y) = ye^{t_1(x,y)}$ and $\tilde{t}(x,y) = t_1(x,y) + t_2(x,y) - t_1(x,y)t_2(x,y)$ for the generating functions. Furthermore we have (due to Scoins (1962))

$$\tilde{t}_{m_1,m_2} = m_1^{m_2-1} m_2^{m_1-1}. \tag{10}$$

Note that $t_1(x,y)$ equals $t_2(y,x)$ and that $t_1(x,x)$ equals the usual tree function $t(x)$ that is given by $t(x) = xe^{t(x)} = \sum_{n\geq 1} n^{n-1}x^n/n!$. Thus, $t_1(x,y)$ and $t_2(x,y)$ are surely analytic functions for $|x| < e^{-1}$ and $|y| < e^{-1}$. This is due to the fact that the radius of convergence of $t(x)$ equals $1/e$.

Our main goal is to count those graphs for which all components are either trees or have exactly one cycle. We will denote this class of graphs $G^{\circ}_{m,m,n}$, an the corresponding generating function $g^{\circ}(x,y,v)$.

Lemma 1 *The generating function $g^{\circ}(x,y,v)$ is given by*

$$g^{\circ}(x,y,v) = \sum_{m,n\geq 0} \#_w G^{\circ}_{m,m,n} \frac{x^m}{m!}\frac{y^m}{m!} v^n = \frac{e^{\frac{1}{v}\tilde{t}(xv,yv)}}{\sqrt{1 - t_1(xv,yv)t_2(xv,yv)}}. \tag{11}$$

4 Asymptotic Methods

We use Lemma 1 and Cauchy's formula to obtain:

$$\#_w G^{\circ}_{m,m,n} = \frac{-(m!)^2}{4\pi^2(2m-n)!} \int_{|x|=x_0}\int_{|y|=y_0} \frac{\tilde{t}(x,y)^{2m-n}}{\sqrt{1-t_1(x,y)t_2(x,y)}} \frac{dx}{x^{m+1}} \frac{dy}{y^{m+1}}. \tag{12}$$

This is in fact an integral that can be asymptotically evaluated with help of a (double) saddle point method. It turns out that (if $\varepsilon \in (0,1)$ is fixed) the saddle point of our function is given by

$$x_0 = y_0 = \frac{n}{m} e^{-\frac{n}{m}} = (1-\varepsilon)e^{\varepsilon - 1} < \frac{1}{e}. \tag{13}$$

Good (1957) provides a result to calculate the asymptotic expansion of $[x^m y^m] f(x,y)^k$, for a suitable function f (see also Drmota (1994)). We use a generalization of this result which provides us an asymptotic expansion of $[x^m y^m] f(x,y)^k g(x,y)$ for suitable function f and g. Comparing this result with the weighted number of all graphs (see Equation 2), provides us Theorem 1.

Introducing a new variable w to extend our generating function $g^{\circ}(x,y,v)$ allows us to mark a parameter like the number of cycles. This grants us excess to the limiting distributions stated in Theorem 2 with help of characteristic functions. See Drmota and Soria (1997) for details of this method.

References

L. Devroye and P. Morin. Cuckoo hashing: Further analysis. *Inf. Process. Lett.*, 86(4):215–219, 2003.

M. Drmota. A bivariate asymptotic expansion of coefficients of powers of generating functions. *Eur. J. Comb.*, 15(2):139–152, 1994.

M. Drmota and M. Soria. Images and preimages in random mappings. *SIAM J. Discrete Math.*, 10(2): 246–269, 1997.

I. J. Good. Saddle-point methods for the multinomial distribution. *Ann. Math. Stat.*, 28(4):861–881, 1957.

S. Janson, D. E. Knuth, T. Luczak, and B. Pittel. The birth of the giant component. *Random Struct. Algorithms*, 4(3):233–359, 1993.

I. B. Kalugin. The number of components of a random bipartite graph. *Discrete Math. Appl.*, 1(3):289–299, 1991.

D. E. Knuth. *The Art of Computer Programming, Volume III: Sorting and Searching*. Addison-Wesley, 1973. ISBN 0-201-03803-X.

R. Pagh and F. F. Rodler. Cuckoo hashing. Research Series RS-01-32, BRICS, Department of Computer Science, University of Aarhus, 2001.

H. I. Scoins. The number of trees with nodes of alternate parity. *Proc. Cambridge Philos. Soc.*, 58:12–16, 1962.