Approximating Almost All Instances of MAX-CUT Within a Ratio Above the Håstad Threshold*

A.C. Kaporis[1], L.M. Kirousis[1,2], and E.C. Stavropoulos[1]

[1] University of Patras, Department of Computer Engineering and Informatics
GR-265 04 Patras, Greece
{kaporis, kirousis, estavrop}@ceid.upatras.gr
[2] Research Academic Computer Technology Institute, P.O. Box 1122, GR-261 10 Patras, Greece

Abstract. We give a deterministic polynomial-time algorithm which for any given average degree d and *asymptotically almost all* random graphs G in $\mathcal{G}(n, m = \lfloor \frac{d}{2} n \rfloor)$ outputs a cut of G whose ratio (in cardinality) with the maximum cut is at least 0.952. We remind the reader that it is known that unless P=NP, for no constant $\epsilon > 0$ is there a MAX-CUT approximation algorithm that for *all inputs* achieves an approximation ratio of $(16/17) + \epsilon$ $(16/17 < 0.94118)$.

1 Introduction

There is a vast and growing literature on approximation algorithms for NP-hard problems. Both in the direction of designing algorithms that give good approximations, as well as in the direction of showing, under a putative hypothesis like P≠NP, that no approximation better than a given bound exists. In this work, we concentrate on the problem of MAX-CUT, that of partitioning the vertex set V of a graph $G = (V, E)$ in two parts so that the number of edges joining vertices in different parts is as large as possible. In more colorful language, MAX-CUT is the problem of coloring the vertices of a graph with two colors (red or blue) so that the bichromatic edges are as many as possible. It is probably needless to elaborate on the interest, from the point of view of either theory or practice, of the NP-hard optimization problem MAX-CUT. Just as an example, let us mention the early considerations of MAX-CUT in relation to circuit layout design and Statistical Physics mentioned in [1] (as pointed out in [4]). In the language of Statistical Physics, MAX-CUT is equivalent to computing the ground energy of the antiferromagnetic Ising model defined on graphs [15].

* The authors are partially supported by European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly *Pythagoras*. The second author is partially supported by Future and Emerging Technologies programme of the EU under contract 001907 *"Dynamically Evolving, Large-Scale Information Systems (DELIS)."*

Y. Azar and T. Erlebach (Eds.): ESA 2006, LNCS 4168, pp. 432–443, 2006.

For maximization problems, like MAX-CUT, we say that an algorithm $\mathcal{A}$ achieves an approximation ratio $0 < \alpha < 1$, if for any input I, the output of the algorithm $\mathcal{A}(I)$ on I relates to an optimum solution $\mathrm{OPT}(I)$ for I as in:

$$\frac{|\mathcal{A}(I)|}{|\mathrm{OPT}(I)|} \geq \alpha.$$

Similarly, we define the approximation ratio of minimization problems. For general graphs, the best MAX-CUT approximation algorithm is, for more than a decade now, the one by Goemans and Williamson [12], which can achieve a ratio arbitrarily close (from below) to

$$\alpha_{\mathrm{GW}} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta} > 0.87856.$$

Under the "Unique Games" and the "Majority is Stablest" conjectures, the above approximation ratio was shown to be optimal by Khot et al. [16] (however, very recently a hypothesis only slightly stronger than the Unique Games conjecture was falsified [5]).

Also by a now classical inapproximability result by Håstad [13] and Trevisan et al. [20], unless P$\neq$NP, MAX-CUT cannot be approximated for general graphs by a deterministic algorithm that attains a ratio strictly exceeding $16/17$ ($16/17 < 0.94118$).

Let now $\mathcal{G}(n;d)$ be the probability space of random graphs with n vertices and $m = \lfloor \frac{d}{2} n \rfloor$ edges selected uniformly at random. It is convenient for the probabilistic calculations to allow repetitions and even self-loops in the selection of edges. This does not affect the results as such selections happen with vanishingly small probability as n grows large. We say that a property $\mathcal{E}$ holds for asymptotically almost all (a.a.a.) random graphs from $\mathcal{G}(n;d)$ if $\lim_n \mathbf{Pr}[G \in \mathcal{G}(n;d)\ \&\ \mathcal{E}$ holds for $G] = 1$. Notice that the negative result for approximation ratios $> 16/17$ does not exclude the possibility of a deterministic algorithm that achieves a ratio of $(16/17) + c$ (c a positive constant) for a.a.a. input instances from $\mathcal{G}(n;d)$, for any given fixed d (see e.g. the pioneering work of Frieze and McDiarmid on graph algorithms on random instances [11]).

With respect to a different problem, namely MAX-SAT, Fernandez de la Vega and Karpinski [8] analyzed an algorithm that achieves an approximation ratio of $8/9$ for a.a.a. instances, with any given ratio of clauses to variables (Håstad [13] has proved that there is no approximation algorithm for MAX-SAT whose ratio strictly exceeds $7/8$). The ratio of $8/9$ was further improved to $19/20$ by Interian [14]. These algorithms for MAX-SAT are Davis-Putnam-style heuristics that do not take into account the number of occurrences of the variable selected to be assigned the value "true" at each step.

Similar heuristics, that ignore degree considerations of the vertices to be put into each part of the cut under construction, have been analyzed for the case of MAX-CUT, or more general versions of it like MAX-k-CUT, in various papers (see [15, 6, 7]), giving a series of interesting lower bound results for the optimal cut. The fact that degree considerations are not taken into account in these algorithms, greatly simplifies their probabilistic analysis. However, as far as the question of the ratio of the size of the output of these algorithms to the size of the optimal cut is concerned, they all yield values that are far below the Håstad threshold, even below the Goemans-Williamson ratio, for values of the average degree in a sizable interval. Also heuristics that take into account degree considerations, but for different graph problems, are analyzed in the work of Beis et al. and others (see [2, 3] and references therein).

To break the Håstad barrier for Max-Cut (for a.a.a. input instances with any given d), it became necessary to follow a double front approach. On one hand, since the size of optimum cut is not known, we had to find improved upper bounds for the optimum cut. On the other, we had to considerably improve the known algorithmic lower bounds. So that using both bounds we could come up with a ratio that exceeds 16/17. Both upper and lower bounds are computed not for the graph itself, but for its 2-core, the maximum induced subgraph whose vertices have degree at least 2. The reason being that, as it is easy to prove, the edges of a graph not belonging to its 2-core, belong to any max cut. Therefore, once we have a max cut of the 2-core, then a max cut of the original graph can be found by adding to the cut the edges that are outside the 2-core. This pruning preprocessing phase considerably improves the bounds, but necessitates carrying our analysis not in $G \in \mathcal{G}(n; d)$, but in the uniform probability space of graphs with a given degree sequence. Our algorithm for the lower bound takes into account the degree of each vertex. The numerical analysis makes use of computer aided computations.

The approximation ratio we get, besides crossing the Håstad threshold, substantially improves the Goemans-Williamson value (0.87856) and thus, to the best of our knowledge, constitutes the first after more than a decade improvement of the approximation ratio of Max-Cut, valid for general graphs (but only for a.a.a. input instances with any given average degree). In the next section we give some necessary formal definitions, state the main result and give some preliminary facts. The main tools of the proof are given in the sections that come after the next one.

2 Preliminaries

Definition 1. *Given a cut $\mathcal{C}$ of a graph G, the cut size of $\mathcal{C}$, denoted by $|\mathcal{C}|$, is the number of edges of G that connect vertices in different parts of $\mathcal{C}$ (bichromatic edges).*

We now give definitions of a.a.a. upper and lower bounds that are given as percentages of (scaled with respect to) m, the number of edges.

Definition 2. *A function* $\mathrm{ub} : \mathbb{R}^+ \to \mathbb{R}^+$ *a.a.a. defines a scaled (with respect to the number of edges m) upper bound* $\mathrm{ub}(d)$ *for the maximum cut size* $\mathrm{mc}(G)$ *of a random graph $G \in \mathcal{G}(n; d)$ if*

$$\lim_n \mathbf{Pr}\left[G \in \mathcal{G}(n; d) \quad \& \quad (\mathrm{ub}(d) + o(1))m \geq \mathrm{mc}(G)\right] = 1, \forall d > 0.$$

Definition 3. *Given (i) a function* lb : $\mathbb{R}^+ \to \mathbb{R}^+$ *and (ii) a deterministic algorithm* $\mathcal{A}$ *that on input a graph* G *outputs a cut* $\mathcal{A}(G)$ *of* G*, we say that* $\mathcal{A}$ *a.a.a. establishes a scaled (with respect to the number of edges* m*) lower bound* lb(d) *on the maximum cut size of a random graph in* $G \in \mathcal{G}(n; d)$ *if*

$$\lim_n \mathbf{Pr}\left[G \in \mathcal{G}(n;d) \quad \& \quad (\mathrm{lb}(d) - o(1))m \le |\mathcal{A}(G)|\right] = 1, \forall d > 0.$$

Proposition 1. *If there are functions* ub *and* lb *and an algorithm* $\mathcal{A}$ *as in Definitions (2) and (3), then* $\forall d$ *and* $\forall \epsilon > 0$, $\mathcal{A}$ *achieves an approximation ratio* $\frac{\mathrm{lb}(d)}{\mathrm{ub}(d)} - \epsilon$ *for* MAX-CUT *for a.a.a. input instances* $G \in \mathcal{G}(n; d)$.

Proof. Immediate from the definitions. □

Theorem 1. *There are functions* ub *and* lb *and an algorithm* $\mathcal{A}$ *as in Definitions (2) and (3) such that* $\forall d$, $\frac{\mathrm{lb}(d)}{\mathrm{ub}(d)} > 0.952$.

Proof. For the case when $d < 1$, then by the proof in [7, Theorem 19] we know that the cut obtained by considering (i) all edges of the tree-components of G, (ii) all edges of its even cyclic components and (iii) all edges but one of its odd cyclic components has cardinality equal to the total number of edges of G within an $o(1)$ additive term. This procedure defines the algorithm $\mathcal{A}$. Also, we set $\mathrm{ub}(d) = \mathrm{lb}(d) = 1$.

For the case of large d, first observe that by coloring red an arbitrary half of the vertices of G, we get a trivial lower bound $\mathrm{lb}(d) = (1/2) - \epsilon$, for any $\epsilon > 0$. By combining this trivial lower bound with the upper bound $\mathrm{ub}(d) = 1/2 + \sqrt{(\ln 2/d)}$ [4], we easily get by solving for d the equation

$$\frac{1/2}{1/2 + \sqrt{(\ln 2/d)}} = \frac{16}{17},$$

that the Theorem holds for $d > 710$,

As for the interval $1 \le d \le 710$, in Section 4 we define the function ub, while in Section 5 we describe and analyze the algorithm $\mathcal{A}$ and define the function lb. The computations involved are computer-aided (but the probabilistic analysis and the derivations of all formulas are analytic). The computer-aided analysis shows that the for $d \ge 20$, the ratio $\frac{\mathrm{lb}(d)}{\mathrm{ub}(d)}$ is bounded below by numbers greater than 0.952 by 0.01 or more. Actually for $d \ge 20$, easier upper and lower bound functions, some of which already given in the literature [15, 6, 7], yield a ratio $\frac{\mathrm{lb}(d)}{\mathrm{ub}(d)}$ that easily exceeds 0.952. So in the following sections we concentrate in the interval $[1, 20]$, where the real difficulty lies, i.e. the interval where the ratio $\frac{\mathrm{lb}(d)}{\mathrm{ub}(d)}$ for the improved upper and lower bound functions that we define closely approaches from above the value 0.952. □

Corollary 1 (Main Result). *There is a deterministic algorithm* $\mathcal{A}$ *such that for any average degree* $d > 0$, $\mathcal{A}$ *achieves an approximation ratio* 0.952 *for* MAX-CUT *for a.a.a. random graphs in* $\mathcal{G}(n; d)$.

Proof. Immediate from Theorem 1 and Proposition 1. □

3 The 2-Core

The 2-core of a graph G is defined to be the largest induced subgraph of G with minimum degree at least 2. For technical reasons, we use an essentially equivalent but formally slightly different definition:

Definition 4. *Given a graph $G = (V, E)$ the 2-core of G, denoted by $K_2(G)$, is the unique subgraph $K_2(G) = (V, E')$, where E' is the maximum (with respect to set-inclusion) subset of E so that with respect to $K_2(G)$ all vertices in V have degree either zero (isolated vertices) or degree at least 2.*

By our definition, the 2-core results by edge-deletions only (and no change in the set of vertices) and the resulting graph has either isolated vertices or vertices of degree at least 2 (retaining throughout our analysis the same set of vertices avoids unnecessary technical complications).

It immediately follows by well known results that $K_2(G)$ can be obtained from G by recursively deleting one at a time and in any order edges that are incident on vertices of degree 1. By assumption, when we delete the edge incident on a vertex v of degree 1, v remains in the graph (but becomes isolated).

Consider now the uniform probability space of graphs such that the number of vertices of degree i is $(e^{-d}(d^i/i!)+o(1))n$, i.e. graphs whose degree sequence is Poisson distributed with mean d. It is known that a.a.a. graphs in $\mathcal{G}(n; d)$ have a Poisson distributed degree sequence with mean d.

In general, let $\mathcal{G}(n; \langle d_i \rangle_{i=0,\ldots,m})$ be the uniform probability space of graphs with n vertices and scaled degree sequence $\langle d_i \rangle_{i=0,\ldots,m}$ (i.e. the number of vertices of degree i is $(d_i+o(1))n$; d_i are assumed to be independent of n). For such graphs we use the configuration model which models random pairings of copies of the vertices, the number of copies of each vertex being equal to its degree. It is well known that results that hold for a.a.a. such pairings in the configuration model, also hold for a.a.a. uniformly distributed simple graphs with the same degree sequence.

It is known that if G is random with a Poisson degree sequence, then $K_2(G)$ is random in $\mathcal{G}(n; \langle d_i \rangle_{i=0,\ldots,m})$ for the same n and a new degree sequence, for which $d_1 = 0$. To compute the new degree sequence, we follow the technique of differential equations of Wormald [21]: we write differential equations that give the dynamics each d_i during the execution of the edge-deletion process. The solution of the differential equations give the final values of d_i within $o(1)$, for $i = 0, \ldots, n-1$. These values hold for a.a.a. input graphs. Our analysis closely follows the methodology given by Mitzenmacher [17] for the case of deletion of pure literals from 3-SAT formulas. We symbolically solve the resulting system of differential equations. Actually, the system of differential equations in our case is easier to obtain and solve, as we do not have the complication of handling the negation of a deleted literal. For reasons of space, we avoid the details (that follow standard techniques) and only give the final result without proof:

Theorem 2. *The number of vertices of degree $i = 0, \dots, m$ of the 2-core $K_2(G)$ of a random graph G in $\mathcal{G}(n, m = \lfloor \frac{d}{2} n \rfloor)$ is a.a.a. $(d_i + o(1))n$, where*

$$d_i = \begin{cases} -\frac{1}{d} W_d (d + 1 + W_d) \text{ when } i = 0, \\ 0 \text{ when } i = 1, \\ -\frac{W_d}{d} \frac{(d+W_d)^i}{i!} \text{ when } i \geq 2, \end{cases} \tag{1}$$

and where W_d is LambertW$(-de^{-d})$, i.e. the value of the principal branch of Lambert's W-function at $-de^{-d}$. Also the number of deleted edges during the edge deletion process that yields the 2-core a.a.a. is $\left(-W_d - \frac{W_d^2}{2d} + o(1)\right) n$. Finally, a property holds a.a.a. for $K_2(G)$ iff it holds a.a.a. for a random graph conditional its degree sequence is as described in Equation (1) above.

It can be easily seen that the number of the edges that are deleted to yield the 2-core are part of any maximum size cut. Therefore, the size of the max cut of G can be obtained from the size of the max cut of $K_2(G)$ by adding to the latter the number $\left(-W_d - \frac{W_d^2}{2d} + o(1)\right) n$. It easily follows that:

Proposition 2. *If the functions* $\mathrm{ub}(d)$ *and* $\mathrm{lb}(d)$ *give scaled (with respect to the number of edges m) upper and lower, respectively, bounds for the size of the max cut of a random graph conditional its degree sequence is that of Theorem 2, then the functions* $\mathrm{ub}(d)+2\left(-W_d - \frac{W_d^2}{2d}\right)/d$ *and* $\mathrm{lb}(d)+2\left(-W_d - \frac{W_d^2}{2d}\right)/d$ *give scaled (with respect to the number of edges m) upper and lower bounds, respectively, for the size of the maximum cut of a random graph in $\mathcal{G}(n, m = \lfloor \frac{d}{2} n \rfloor)$.*

The previous proposition allows us to work with a random graph conditional its degree sequence is as in Theorem 2.

4 The Upper Bound

For a random graph in $\mathcal{G}(n, m = \lfloor \frac{d}{2} n \rfloor)$, a simple application of the first moment method gives that the maximum cut is no more than $\left(\frac{1}{2} + \sqrt{\frac{\ln 2}{d}}\right) \frac{d}{2} n$ for a.a.a. input instances with average degree d, for $d \geq 4 \ln 2$ [4]. This bound is established by estimating the probability of existence of a cut of a given size z by the expectation of the number of cuts of size z. However, first moment estimations are in general, and in this particular case as well, rather gross.

Another well known approach to the question of finding an upper bound for the optimum cut is by semidefinite relaxation of the problem [12]. However, it is in general difficult to estimate the average-case (or typical-case, i.e. a.a.a.-instances-case) output of a semidefinite program. A related result can be found in [6, Theorem 4], which however gives an estimation of the SDP upper bound of MAX-CUT in terms of an unspecified constant only. An earlier bound was obtained by Linear Programming relaxation [1], but with respect to sparse graphs it is shown in [19] that the upper bound obtained by an LP relaxation of MAX-CUT is a.a.a. at most the total number of edges, i.e. no information is obtained.

So we have to resort to other means in order to compute a better bound suited for typical-case considerations. We compute the expected number of *majority* cuts of given size z for a random graph conditional its degree sequence is as in Theorem 2.

Definition 5. *A cut is called a majority cut if (i) at least half of the edges incident on any vertex are bichromatic (i.e., they connect vertices in different parts of the cut) and (ii) any vertex of even degree whose exactly half of its incident edges are bichromatic is necessarily colored red (i.e., it belongs to a prescribed part of the cut).*

Theorem 3. *If a cut of size z exists then also a majority cut of size at least z exists.*

Proof. Given a cut which is not necessarily a majority cut move —one at a time and recursively— vertices that violate any of the two conditions of Definition 5 to the other part of the cut. In any such move, the cut size either remains constant or strictly increases. Also, the process cannot continue indefinitely, as at each move either (i) the cut size increases strictly (when we move a vertex with a minority of bichromatic incident edges), or alternatively (ii) the cut size remains constant but the cardinality of the vertices colored red strictly increases in comparison to its immediately previous value (when we move to the red color a vertex with equal number of bichromatic and monochromatic incident edges). To prove more formally that the process does not continue indefinitely, introduce as the *potential* of a cut the pair of numbers (c, r), where c is the current size of the cut and r is the current cardinality of red vertices, order the set of these pairs lexicographically and observe that each move of a violating vertex to the other part drives the cut to a strictly higher potential, because each move either strictly increases c, or keeps c constant and strictly increases r (in comparison to its previous value). Therefore there must be a stopping time. □

Let $\mathcal{G}(n; d, \text{2-core})$ denote the uniform probability space of the 2-core of a random graph in $\mathcal{G}(n, m = \lfloor \frac{d}{2}n \rfloor)$ (see Theorem 2 and Proposition 2 of Section 3). In the sequel, let G be a random graph in $\mathcal{G}(n; d, \text{2-core})$.

Let $\mathcal{C}_\zeta(G)$ be the class of all majority cuts of G with cut size *at least* ζm, where ζ is a real in $[0, 1]$ and $\zeta m = \zeta \lfloor (d/2)n \rfloor$ is an integer in $\{0, \ldots, m\}$. We will compute an a.a.a. scaled upper bound $\text{ub}(d)$ to the values of ζ for which $\mathcal{C}_\zeta(G) \neq \emptyset$ (which by Theorem 3 is also an a.a.a. scaled upper bound to the maximum cut size of G) by finding a minimum value of ζ such that:

$$\lim_n \mathbf{Pr}[|\mathcal{C}_\zeta(G)| > 0] = 0 \tag{2}$$

Towards this end, first observe that the following Markov-type inequality holds:

$$\mathbf{Pr}[|\mathcal{C}_\zeta(G)| > 0] \leq \mathbf{Ex}(|\mathcal{C}_\zeta(G)|). \tag{3}$$

Therefore, to find a minimum ζ for which Equation 2 holds, it is sufficient to find a minimum ζ for which

$$\lim_n \mathbf{Ex}(|\mathcal{C}_\zeta(G)|) = 0. \tag{4}$$

Let now $\mathcal{E}(b_{00}, b_{11}, b_{01})$ be the expected number of majority cuts whose edges connecting two red (respectively, blue, of different color) vertices have cardinality *exactly* $b_{00}n$ (respectively, $b_{11}n, b_{01}n$), where b_{00}, b_{11}, b_{01} belong to the interval $[0, 1]$ and sum to the scaled number of edges $d/2$. It is easy to see that Equation (4) holds iff the following is true:

$$\lim_{n} \left(\max_{\zeta m \leq b_{01}n, b_{00}, b_{11}} \{\mathcal{E}(b_{00}, b_{11}, b_{01})\} \right) = 0. \tag{5}$$

The analytic computation of $\mathcal{E}(b_{00}, b_{11}, b_{01})$ and the computer-aided numerical calculation of the smallest ζ for which Equation (5) holds follow techniques previously used in [10] (see also [9]). Details are omitted for reasons of space.

In Figure 1 we indicatively plot ub(d) for values of $d \in [1, 20]$, juxtaposing it with the plot of the scaled with respect to m upper bound $\frac{1}{2} + \sqrt{\frac{\ln 2}{d}}$ obtained in [4] by the simple first moment method.

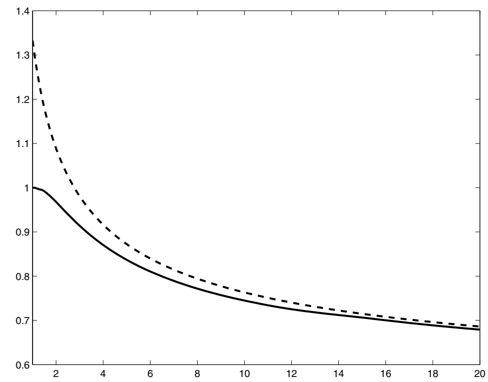

Fig. 1. The upper bound ub(d) given in Section 4 (solid line) versus the upper bound $\frac{1}{2} + \sqrt{\frac{\ln 2}{d}}$ given in [4] (dashed line) for values of average degree $d \in [1, 20]$

5 The Algorithmic Lower Bound

In this section we describe an algorithm $\mathcal{A}$ that on input a the 2-core $K_2(G)$ of a random graph G in $\mathcal{G}(n; d)$ outputs a coloring C of the vertices of $K_2(G)$ with one of the colors in $\{R, B\}$ (i.e. C is a cut). We remind the reader that by Theorem 2, $K_2(G)$ can be assumed to be random conditional its degree sequence is as in Equation (1). Let $|\mathcal{A}(K_2(G))|$ be the size of the cut C, i.e. the number of its bichromatic edges. From $|\mathcal{A}(K_2(G))|$, we can then compute a scaled lower bound of the max cut of the original graph G by Proposition 2.

The algorithm $\mathcal{A}$ colors the vertices of $K_2(G)$ one at a step. Let $d(v)$ be the degree of the vertex v in $K_2(G)$. At any step t of the algorithm, let U^t be the set of yet uncolored vertices of $K_2(G)$. For $v \in U^t$, let $d^t_R(v)$ ($d^t_B(v)$, respectively) be the number of vertices that are neighbors of v and are already colored with R (B, respectively). Also let $d^t_U(v) = d(v) - d^t_R(v) - d^t_B(v)$, i.e. $d^t_U(v)$ is the number of neighbors of v in $K_2(G)$ that are yet uncolored. Finally let the *discrepancy* $\Delta^t(v)$ of a vertex $v \in U^t$ be $|d^t_R(v) - d^t_B(v)|$. The algorithm $\mathcal{A}$ at any step t first locates the vertices $v \in U^{t-1}$ that have the largest discrepancy $\Delta^{t-1}(v)$ and chooses among them one with the lowest $d^{t-1}_U(v)$. It then assigns to v the color R if $d^{t-1}_B(v) \geq d^{t-1}_R(v)$ and B otherwise. Intuitively, $\mathcal{A}$ at any step greedily maximizes the difference of the number of edges to be placed in the cut from the number of edges to remain out of it. At the same time, it minimizes the impact of each color assignment to future assignments.

The algorithm $\mathcal{A}$ is described in pseudo-code in Algorithm 5. Its analysis is based on the method of differential equations. The equations give a lower bound on the size of the maximum cut of $K_2(G)$. They are analytically obtained and numerically solved (details are omitted for reasons of space). In Figure 2, we give a plot of the final value of lb(d) (i.e. the value obtained after applying Proposition (2)) for $d \in [1, 20]$ compared with the values of the algorithms in Coja-Oghlan et al. [6] and Coppersmith et al. [7]. To corroborate the results obtained by numerically solving the analytically derived differential equations, we performed simulation experiments. The simulations gave, as expected, the same values for lb(d) as the numerical solutions of the differential equations. In Table 1 we juxtapose the simulation results with the results obtained from the differential equations, for certain indicative values of d.

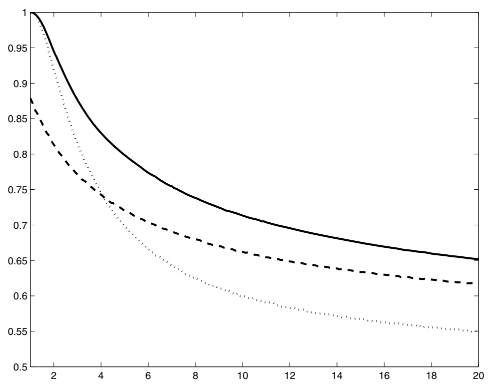

Fig. 2. Our values of the lower bound $lb(d)$ (solid line), obtained by the numerical solution of differential equations (and corroborated by simulation experiments), juxtaposed with the corresponding values obtained by simulating the algorithms in Coja-Oghlan et al. [6] (dashed line) and Coppersmith et al. [7] (dotted line), for values of average degree $d \in [1, 20]$

Algorithm. $\mathcal{A}(K_2(G) = (V_{K_2}, E_{K_2}), C)$
$t = 0$; $U^0 = V_{K_2}$; /* Initialize the set of yet uncolored vertices of $K_2(G)$ */
for all $v \in V_{K_2}$ **do** /* Initialize the number of neighbors of each vertex v of $K_2(G)$ */
  $d_U^0(v) = d(v)$; $d_R^0(v) = 0$; $d_B^0(v) = 0$;
end for
while $U^t \neq \emptyset$ **do** /* while there are uncolored vertices */
  $t = t + 1$;
  Locate all vertices $v \in U^{t-1}$ having the largest discrepancy $\Delta^{t-1}(v)$;
  Among them, arbitrarily choose a vertex v with the lowest $d_U^{t-1}(v)$;
  if $d_B^{t-1}(v) \geq d_R^{t-1}(v)$ **then**
    $C[v] = R$; /* Assign color R to v */
    /* Update the number of colored R and yet uncolored neighbors of each neighbor u of v */
    for each vertex u adjacent to v **do**
      $d_R^t(u) = d_R^{t-1}(u) + 1$;
      $d_U^t(u) = d_U^{t-1}(u) - 1$;
    end for
  else
    $C[v] = B$; /* Assign color B to v */
    /* Update the number of colored B and yet uncolored neighbors of each neighbor u of v */
    for each vertex u adjacent to v **do**
      $d_B^t(u) = d_B^{t-1}(u) + 1$;
      $d_U^t(u) = d_U^{t-1}(u) - 1$;
    end for
  end if
  $U^t = U^{t-1} \setminus \{v\}$; /* Update the set of yet uncolored vertices of $K_2(G)$ */
end while

Algorithm 1. Algorithm $\mathcal{A}$ takes as input the 2-core $K_2(G)$ of a random graph $G = (V, E)$ and returns a coloring C of its vertices

6 Conclusion and Discussion

Putting together the computations of the previous sections, we reach the conclusion that for every average degree $d > 0$, a.a.a. $\frac{\mathrm{lb}(d)}{\mathrm{ub}(d)} > 0.952$. Therefore our main result, Corollary 1, has been proved. In Figure 3 we give a plot of the ratio $\mathrm{ub}(d)/\mathrm{lb}(d)$ for various values of d, especially close to the average densities where the ratio approaches (from above) the Håstad threshold. Theoretically it is conceivable that there might exist a deterministic algorithm that a.a.a. computes exactly the maximum cut size of a random graph or, more realistically, offers a Polynomial Time Approximation Scheme to it (PTAS). We believe that for at least certain values of d there is no such PTAS valid a.a.a. However it is conceivable that for every given $\epsilon > 0$, one might come with an algorithm that yields an a.a.a. approximation scheme of ratio ϵ. Finally, when d is not constant, but approaches infinity with n (dense graphs), then it is known that a.a.a. $(1/2)|E| < |E|((1/2) + o(1))$ [18].

Table 1. Simulation experiment results (SE) versus numerical solution of the differential equations (DE) for indicative values of average degree d

d	2.0	3.5	4.0	4.5	5.0	6.0	8.0	10.0	12.0	14.0
SE	0.945	0.850	0.829	0.813	0.798	0.773	0.738	0.713	0.696	0.681
DE	0.945	0.851	0.830	0.813	0.798	0.774	0.738	0.713	0.695	0.681

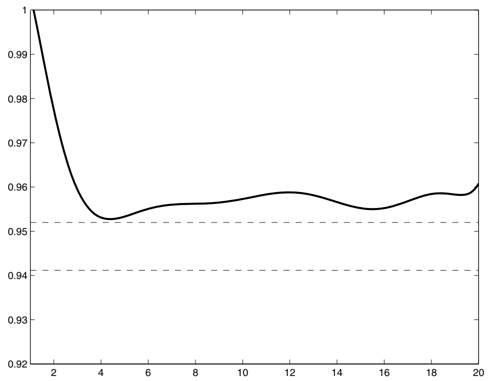

Fig. 3. The approximation ratio $\text{lb}(d)/\text{ub}(d)$, for values of average degree $d \in [1, 20]$. The lower dashed line corresponds to Håstad inapproximability threshold 16/17, while the upper dashed line to our approximation ratio 0.952.

We believe that these results can also be extended to the case of d-regular graphs. We are currently working on this. Also these results extend to k-Max-Cut for $k > 2$.

Acknowledgment

We thank Giorgos Kounenis for providing us with the symbolic solution of the system of differential equations that give the degree sequence of the 2-core.

References

1. F. Barahona, M. Grotschel, M. Junger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36:493–513, 1988.
2. M. Beis, W. Duckworth, and M. Zito. Packing edges in random regular graphs. In K. Diks and W. Rytter, editors, *MFCS*, volume 2420 of *Lecture Notes in Computer Science*, pages 118–130. Springer, 2002.
3. M. Beis, W. Duckworth, and M. Zito. Large k-separated matchings of random regular graphs. In V. Estivill-Castro, editor, *ACSC*, volume 38 of *CRPIT*, pages 175–182. Australian Computer Society, 2005.

4. A. Bertoni, P. Campadelli, and R. Posenato. Un upper bound for the maximum cut mean value. In *Proceedings of the 23rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG97)*, volume 1335, pages 78–84. Lecture Notes in Computer Science, Springer, 1997.
5. M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th ACM Symposium on Theory of Computing Seattle, Washington, USA*, 2006.
6. A. Coja-Oghlan, C. Moore, and V. Sanwalani. Max k-cut and approximating the chromatic number of random graphs. In J.C.M. Baeten, J.K. Lenstra, J. Parrow, and G.J. Woeginger, editors, *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2003.
7. D. Coppersmith, D. Gamarnik, M.T. Hajiaghayi, and G.B. Sorkin. Random max sat, random max cut, and their phase transitions. *Random Struct. Algorithms*, 24(4):502–545, 2004.
8. W. Fernandez de la Vega and M. Karpinski. 9/8-approximation algorithm for random max-3sat. *Electronic Colloquium on Computational Complexity (ECCC)*, (070), 2002.
9. J. Díaz, G. Grammatikopoulos, A.C. Kaporis, L.M. Kirousis, X. Pérez, and D.G. Sotiropoulos. 5-regular graphs are 3-colorable with positive probability. In G.S. Brodal and S. Leonardi, editors, *ESA*, volume 3669 of *Lecture Notes in Computer Science*, pages 215–225. Springer, 2005.
10. O. Dubois and J. Mandler. On the non-3-colourability of random graphs. *ArXiv Mathematics e-prints*, September 2002.
11. A.M. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Struct. Algorithms*, 10(1-2):5–42, 1997.
12. M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.
13. J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48:798–869, 2001.
14. Y. Interian. Approximation algorithm for random MAX-k-SAT. In *International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, 2004.
15. V. Kalapala and C. Moore. Max-cut on sparse random graphs. *TR-CS-2002-24, University of New Mexico Department of Computer Science*, 2002.
16. S. Khot. Hardness of approximating the shortest vector problem in lattices. In *FOCS*, pages 126–135. IEEE Computer Society, 2004.
17. M. Mitzenmacher. Tight thresholds for the pure literal rule. Technical report, Digital Equipment Corporation, 1997. Available at: www.research.compaq.com/SRC/.
18. Ng. Van Ngoc and Zs. Tuza. Linear-time algorithms for the max cut problem. *Combinatorics, Probability & Computing*, 2:201–210, 1993.
19. S. Poljak and Z. Tuza. The expected relative error of the polyhedral approximation of the max-cut problem. *Operations Research Letters*, 16:191–198, 1994.
20. L. Trevisan, G. Sorkin, M. Sudan, and D. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.
21. N.C. Wormald. Differential equations for random processes and random graphs. *The Annals of Applied Probability*, 5(4):1217–1235, 1995.