Greedy algorithms for minimisation problems in random regular graphs

Michele Zito

Department of Computer Science, University of Liverpool, Liverpool, L69 7ZF, UK.

Abstract. In this paper we introduce a general strategy for approximating the solution to minimisation problems in random regular graphs. We describe how the approach can be applied to the minimum vertex cover (MVC), minimum independent dominating set (MIDS) and minimum edge dominating set (MEDS) problems. In almost all cases we are able to improve the best known results for these problems. Results for the MVC problem translate immediately to results for the maximum independent set problem. We also derive lower bounds on the size of an optimal MIDS.

1 Introduction

This paper is concerned with graphs generated uniformly at random according to the $\mathcal{G}(n, r\text{-reg})$ model [19, Chap. 9]. Let n urns be given, each containing r balls (with rn even): a set of $rn/2$ pairs of balls is chosen at random. To get a random graph $G = (V, E)$, identify the n urns with the graph's n vertices and let $\{i, j\} \in E$ if and only if there is a pair with one ball belonging to urn i and the other one belonging to urn j. The maximum degree of a vertex in G is at most r. Moreover, for every integer $r > 0$, there is a positive fixed probability that the random pairing contains neither pairs with two balls from the same urn nor couples of pairs with balls coming from just two urns. In this case the graph is r-regular (all vertices have the same degree r). Notation $G \in \mathcal{G}(n, r\text{-reg})$ will signify that G is selected according to the model $\mathcal{G}(n, r\text{-reg})$. An event $\mathcal{E}_n$, describing a property of a random graph depending on a parameter n, holds *asymptotically almost surely* (a.a.s.), if the probability that $\mathcal{E}_n$ holds tends to one as n tends to infinity.

Heuristics for approximating the solution to minimisation problems in random graphs are often based on algorithms for maximisation problems. A small vertex cover or a good vertex colouring can be found by running an independent set algorithm (see for example [17]). The minimum k-center problem [18] can be solved optimally a.a.s. by using a greedy heuristic for a generalisation of the maximum independent set problem [23]. In this paper we introduce a general strategy for approximating the solution to minimisation problems in random regular graphs. The approach has a number of advantages over previously known heuristics. First of all it is more natural to these problems: it does not use as a subroutine any algorithm for an associated maximisation problem. Secondly, it is simple to implement. It is based on a greedy algorithm that repeatedly picks vertices of given degree in the current graph, updates the partial solution obtained so far and then removes few vertices from the graph. The proposed algorithm is also

quite simple to analyse. At the price of performing a larger case analysis, the only mathematical tool needed to estimate the size of the final solution in each case is a theorem of Wormald [26]. The analyses given in other papers [3, 23, 25] use other results in the theory of random processes (e.g. random walks or birth-death processes). No such a tool is needed here. A further advantage of the approach described in this paper is that it uniformly extends to random r-regular graphs for each fixed $r \geq 3$. Finally, using this approach, we are able to give best known results in almost all cases considered.
The rest of the paper is organised as follows: in Section 2 we define the problems of interest and present a summary of the results proved in this paper, in Section 3 we describe our algorithmic approach, present the analysis method along with the statement of the main result of this paper; in Section 4 we complete the proof of the main theorem for each of the problems considered; some conclusions are drawn in Section 5.

2 Problems and Results

We first define the problems of interest and give a short survey of the relevant literature. The reader may find elsewhere [15, 4, 9, 12] more detailed bibliographic notes. The tables at the end of the section give, for each problem and for the first few values of r, a lower bound $\lambda = \lambda(r)$ on the optimal size of the structure of interest divided by n, the best known upper bound $\mu = \mu(r)$ on the same quantity, and the bound $\sigma = \sigma(r)$ obtained in this paper. In all cases the bounds hold a.a.s. for $G \in \mathcal{G}(n, r\text{-reg})$.

Minimum Vertex Cover. (MVC) A *vertex cover* in a graph G is a set of vertices that intersects every edge of G. Let $\tau(G)$ be the size of the smallest vertex covers in G. The MVC problem asks for a vertex cover of size $\tau(G)$.
The problem is NP-hard to approximate within some fixed constant larger than one for planar cubic graphs [16, 2]. It is approximable within $\frac{7}{6}$ for graphs of maximum degree three [5]. The author knows of no result explicitly stated in terms of vertex covers on any random graph model, but results on small vertex covers can be derived from results on large independent sets. The values of λ and μ in the table MVC below are implicitly obtained in [22, 25].

Maximum Independent Set. (MIS) An *independent set* in a graph G is a set of vertices containing no edge of G. Let $\alpha(G)$ be the size of the largest independent sets in G. The MIS problem asks for an independent set of size $\alpha(G)$.
The problem is NP-hard to solve optimally for planar cubic graphs [16], and NP-hard to approximate within some fixed constant for bounded degree graphs [6]. On the same class of graphs the problem is approximable within a constant factor [5]. Many results are known on the most likely value of $\alpha(G)$ in random graphs [7, 13]. If $G \in \mathcal{G}(n, r\text{-reg})$, for any $\epsilon > 0$ there is an r_ϵ such that

$$\left|\alpha(G) - \tfrac{2n}{r}(\log r - \log\log r + 1 - \log 2)\right| \leq \tfrac{\epsilon n}{r}$$

a.a.s. for any $r \geq r_\epsilon$ [14]. For smaller values of r the best bounds are proved in [22] and [25].

Minimum Independent Dominating Set. (MIDS) An *independent dominating set* in a graph G is an independent set U such that all vertices in $V \setminus U$ are adjacent to at least one element of U. Let $\gamma(G)$ be the size of the smallest independent dominating sets in G. The MIDS problem asks for an independent dominating set of size $\gamma(G)$.
The problem is NP-hard [15] to solve optimally for graphs of maximum degree three. Moreover, for each r, there are constants $1 < b_1 < b_2$ such that the problem is approximable within b_2 [1] and NP-hard to approximate within b_1 [20] in graphs of max degree r. The values of $\lambda(3)$ and $\mu(3)$ in the table for MIDS were obtained in [11] whereas all other values of μ follow from the fact that almost all regular graphs are hamiltonian [24]. All values of $\lambda(r)$ for $r \geq 4$ are proved for the first time in Theorem 2.

Minimum Edge Dominating Set. (MEDS) An *edge dominating set* in a graph G is a set of edges F such that every $e \in E \setminus F$ is incident to at least an element of F. Let $\beta(G)$ be the size of the smallest edge dominating sets in G. The MEDS problem asks for an edge dominating set of size $\beta(G)$.
The problem was first shown to be NP-hard to solve optimally by Yannakakis and Gavril [27]. It is NP-hard to solve within some constant factor of the optimal for cubic graphs [28]. Any maximal matching has at most $2\beta(G)$ edges [21]. Recently Duckworth and Wormald [10] gave the first non-trivial approximation result for (worst-case) cubic graphs (this gives $\mu(3)$). The values of λ in the table below were proved in [29].

MVC			
r	λ	μ	σ
3	0.5446	0.5672	0.5708
4	0.5837	0.6099	0.6122
5	0.6156	0.6434	0.6459
6	0.642	0.6704	0.6721
7	0.6643	0.6929	0.6966

MIDS			
r	λ	μ	σ
3	0.2641	0.2794	0.281
4	0.2236	$0.\overline{3}$	0.2453
5	0.1959	$0.\overline{3}$	0.2197
6	0.1755	$0.\overline{3}$	0.2001
7	0.1596	$0.\overline{3}$	0.1848

MEDS			
r	λ	μ	σ
3	0.3158	$0.45 + o(1)$	0.3528
4	0.315		0.3683
5	0.3176		0.3797
6	0.3212		0.3881
7	0.3321		0.3978

3 Algorithm and Analysis Method

Next we describe the algorithm (template) that will be used to solve the problems above. The procedure is parameterised to the problem Π, the degree of the input graph r, a constant $\epsilon \in (0, 1)$, and a sequence of numbers $c_1, \ldots, c_r$ where c_r can be safely set to be one, and details on how to choose all other c_j's will be given towards the end of this section. Let $V_i = \{v : \deg_G v = i\}$. We call a *step* one iteration of the while loop and a *phase* one complete execution of such loop. The algorithm runs for r phases: Phase 0, Phase 1, etc. In the following discussion G always denotes the subgraph of the input graph still to be dealt with after a certain number of steps. In each step a further portion of G is dealt with.

Algorithm DescendingDegree$_{\Pi,r,\epsilon,c_1,\ldots,c_r}(G)$:
Input: an r-regular graph $G=(V,E)$ on n vertices.

$\mathcal{S} \leftarrow \emptyset$;
for $j=0$ **to** $r-1$
  while $|\mathcal{S}| \leq \lceil c_{j+1} n \rceil$
    pick at random a vertex u of degree $r-j$;
    update$_{\Pi,r}(\mathcal{S})$;
    shrink$_{\Pi,r}(G)$;
    if $(j<r-1) \wedge (\exists i : V_i = \emptyset)$ **return** FAILURE;
    else if $(j=r-1) \wedge \sum_{i=1}^{r} i|V_i| \leq \lceil n^{\epsilon} \rceil$
      clean-up$_{\Pi,r}(\mathcal{S},G)$;
      output$(\mathcal{S})$;
    else if $(\exists i : V_i = \emptyset)$ **return** FAILURE;
if $|E|>0$ **return** FAILURE.

In all cases the output structure $\mathcal{S}$ is updated as follows. Let $N(u)=\{v \in G : \{u,v\} \in E\}$. For each Π, function *update* probes the degree of each $v \in N(u)$ and decides what to add to $\mathcal{S}$ based on the sequence $\boldsymbol{d} \equiv (d_1, d_2, \ldots, d_{\deg u})$ where $r-1 \geq d_1 \geq d_2 \geq \ldots \geq d_{k_u} > d_{k_u+1} = \ldots = d_{\deg u} = 0$ are the degrees of u's neighbours in $G \setminus u$, the graph obtained from G by removing u and all its incident edges. In all cases $|\mathcal{S}|$ increases by one if $k_u < \deg u$ (i.e. u has at least one neighbour of degree one) and by at least one if $k_u = \deg u$ (i.e. all neighbours of u have degree at least two). We call *Type 1* (resp. *Type 2*) *configurations* the sequences $\boldsymbol{d}$ of the first (second) kind. Function *shrink* will then remove u and all its incident edges from G. The choice of further vertices to be removed depends on Π and details are given in Section 4. Each step may end in one of two legal conditions. If $V_i \neq \emptyset$ for all i, the algorithm proceeds to the next step (possibly starting a new phase). If $j=r-1$ and the $\sum_{i=1}^{r} i|V_i|$ is less than $\lceil n^{\epsilon} \rceil$ the algorithm is (almost) ready to output the structure $\mathcal{S}$. The function *clean-up*, in each case, completes $\mathcal{S}$ according to some fixed (but otherwise arbitrary) greedy heuristic (e.g. *clean-up*$_{\mathrm{MEDS},r}(\mathcal{S},G)$ might return any maximal matching in G). Any other condition is illegal and the algorithm terminates with a "FAILURE" message.

The major outcome of this paper is that although in principle, for a given choice of the c_j's, the evolution of the $|V_i|$'s could be quite arbitrary, for the problems considered it is possible to "fine-tune" the c_j's to control the $|V_i|$'s (in particular these numbers remain positive throughout the algorithm execution, at least a.a.s.) and hence the size of the structure of interest. For $j \geq 0$, let $Y_i = Y_i^j(t)$ be the size of V_{r-i} after step t of Phase j, with $Y_0^0(0) = n$ and $Y_i^0(0) = 0$ for all $i > 0$, and $Y_i^{j+1}(0) = Y_i^j(\lceil c_j n \rceil)$ for all $i \geq 0$ and $j \in \{1, \ldots, r-1\}$. Notice in particular that $|Y_i^j(t+1) - Y_i^j(t)| \leq r^{O(1)}$ for all t, i, and j. Finally, let $X^j(t) = \sum_{i=0}^{r-1}(r-i)Y_i^j(t)$. From now on the dependency on t and j will be omitted unless ambiguity arises. The key ingredient in the analysis of the algorithm above is the use (in each Phase) of Theorem 5.1 in [26] which provides tight asymptotics for the most likely values of Y_i (for each $i \in \{0, \ldots, r\}$), as the algorithm progresses through successive Phases.

Note that it would be fairly simple to modify the graph generation process described in Section 1 to incorporate the decisions made by the algorithm DescendingDegree: the input random graph and the output structure $\mathcal{S}$ would be generated at the same time. In this setting one would keep track of the degree sequence of the so called *evolving*

graph H_t (see for instance [25]). H_0 would be empty, and a number of edges would be added to H_t to get H_{t+1} according to the behaviour of algorithm DescendingDegree at step $t+1$. The random variables Y_i also denote the number of vertices of degree i in the evolving graph H_t. We prefer to give our description in terms of the original regular graph.

Let $\mathrm{E}(\Delta Y_i)$ denote the expected change of Y_i during step $t+1$ in some given Phase, conditioned to the history of the algorithm execution from the start of the Phase until step t. This is asymptotically

$$\sum_{\boldsymbol{d}} \mathrm{E}(\Delta Y_i \mid \boldsymbol{d}) \Pr[\boldsymbol{d}] \tag{1}$$

where $\mathrm{E}(\Delta Y_i \mid \boldsymbol{d})$ is the expected change of Y_i conditional to the degrees of u's neighbours in $G \setminus u$ being described by $\boldsymbol{d}$, $\Pr[\boldsymbol{d}]$ denotes the probability that $\boldsymbol{d}$ occurs in u's neighbourhood conditioned to the algorithm history so far and the sum is over all possible configurations in the given Phase. Also, in each case, the expected change in the size of the structure output by the algorithm, $\mathrm{E}(\Delta|\mathcal{S}|)$, is asymptotically

$$\sum_{\boldsymbol{d}\in \mathrm{Type1}} \Pr[\boldsymbol{d}] + c_{\Pi} \sum_{\boldsymbol{d}\in \mathrm{Type2}} \Pr[\boldsymbol{d}] \tag{2}$$

where c_{Π} is a problem specific constant that will be defined in Section 4. Setting $x = t/n$, $y_i^j(x) = Y_i^j/n$ (again dependency on j will be usually omitted), and $\sigma(x) = |\mathcal{S}|/n$, the following system of differential equations is associated with each Phase,

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = \tilde{\mathrm{E}}(\Delta Y_i)\big|_{t=xn, Y_i = y_i n} \quad \frac{\mathrm{d}\sigma}{\mathrm{d}x} = \tilde{\mathrm{E}}(\Delta|\mathcal{S}|)\big|_{t=xn, Y_i = y_i n} \tag{3}$$

where $\tilde{\mathrm{E}}(\Delta Y_i)$ and $\tilde{\mathrm{E}}(\Delta|\mathcal{S}|)$ denote the asymptotic expressions for the corresponding expectations obtained from (1) and (2) using the estimates on $\Pr[\boldsymbol{d}]$ and $\mathrm{E}(\Delta Y_i \mid \boldsymbol{d})$ given later on. Since x does not occur in $\tilde{\mathrm{E}}(\Delta Y_i)$ and $\tilde{\mathrm{E}}(\Delta|\mathcal{S}|)$, we actually solve, one after the other, the systems

$$\frac{\mathrm{d}y_i}{\mathrm{d}\sigma} = \frac{\tilde{\mathrm{E}}(\Delta Y_i)}{\tilde{\mathrm{E}}(\Delta|\mathcal{S}|)}\big|_{Y_i = y_i n} \tag{4}$$

where differentiation is w.r.t. σ, setting $y_0^0(0) = 1$ and $y_i^0(0) = 0$ for all $i \geq 1$ and using $y_i^j(c_j)$, the final conditions of Phase j, as initial conditions of Phase $j+1$ for each $j > 0$. Theorem 5.1 in [26] can be applied (as long as X is not too small) to conclude that a.a.s. the solutions to the systems (3) multiplied by n approximate the values of the variables Y_i and $|\mathcal{S}|$. Furthermore, in each case, for any given $\epsilon \in (0,1)$, there is a value $\sigma = \sigma_\epsilon$ such that X becomes less than $\lceil n^\epsilon \rceil$ during Phase $r-1$ for $|\mathcal{S}| > \sigma n$. After this point G has less than n^ϵ vertices of positive degree and the contribution to $|\mathcal{S}|$ coming from running the algorithm DescendingDegrees to completion is only $o(n)$. The following statement summarises our results:

Theorem 1. *Let* $\Pi \in \{\mathrm{MVC}, \mathrm{MIDS}, \mathrm{MEDS}\}$. *For each integer* $r \geq 2$ *and* $\epsilon \in (0,1)$ *there is a sequence of values* $c_1, \ldots, c_r$, *and a constant* σ *such that the algorithm* $\mathrm{DescendingDegree}_{\Pi, r, \epsilon, c_1, \ldots, c_r}$ *returns a structure* $\mathcal{S}$ *with* $|\mathcal{S}| \leq \sigma n + o(n)$ *a.a.s.*

In the remainder of this section a formula is proved for the number of configurations in each Phase (its proof gives a method to list them which is needed to compute $\tilde{\mathrm{E}}(\Delta Y_i)$ and $\tilde{\mathrm{E}}(\Delta|\mathcal{S}|)$), details are given on how to compute asymptotic expressions for $\Pr[\boldsymbol{d}]$

and $\mathrm{E}(\Delta Y_i \mid \boldsymbol{d})$ for each $\boldsymbol{d}$, and some comments are made on how to solve the systems in (4) and how to choose the constants c_j. Section 4 provides a definition for all problem specific quantities needed to complete the proof of Theorem 1.

Listing configurations. Let $A_=(r,k)$ (resp. $A_\le(r,k)$) be the number of configurations having exactly (resp. at most) k vertices and whose largest positive degree is r. Then $r + \sum_{j=2}^{r}\sum_{i=1}^{r-1} A_\le(i,j)$ is the total number of configurations of Type 1 and $\sum_{j=1}^{r}\sum_{i=1}^{r-1} A_=(i,j)$ is the total number of configurations of Type 2. More precisely, in Phase $r-j$ there are $1+\sum_{i=1}^{r-1} A_\le(i,j-1)$ configurations of Type 1 (this reduces to 1 for $j=1$, assuming $A_\le(i,0)=0$) and $\sum_{i=1}^{r-1} A_=(i,j)$ configurations of Type 2.

Lemma 1. *$A_\le(r,k) = A_=(r+1,k) = \binom{k+r-1}{r}$, for every positive integer r,k and $A_=(1,k)=1$ for every positive integer k.*

Proof. The equality for $A_\le(r,k)$ follows from the one for $A_=(r,k)$ by induction on k since a sequence of at most k numbers starting with a r can either be a sequence of length at most $k-1$ starting with a r (there are $A_\le(r,k-1)$ of them) or it can be a sequence of exactly k numbers starting with a r.
$A_=(r,k) = \binom{k+r-2}{r-1}$ can be proved by induction on $r+k$. A sequence of length exactly k beginning with an r can continue with an $i \in \{1,\dots,r\}$ (in $A_=(i,k-1)$ possible ways). Therefore $A_=(r,k) = A_=(r,k-1) + A_=(r-1,k)$ with initial condition $A_=(r,1)=1$ for every $r \ge 1$. Assuming $A_=(i,j) = \binom{j+i-2}{i-1}$ for every i and j with $i+j < r+k$ the equality for $A_=(r,k)$ follows from $\binom{a}{b} + \binom{a}{b-1} = \binom{a+1}{b}$. □
The proof of Lemma 1 implicitly gives a method for listing all configurations of given type.

Probability of a configuration. The formula for $\Pr[\boldsymbol{d}]$ is better understood if we think of the algorithm DescendingDegree as embedded in the graph generation process. A configuration $\boldsymbol{d} \equiv (d_1, d_2, \dots, d_{\deg u})$ occurs at the neighbourhood of a given u if the $\deg u$ balls still available at the given step in the urn U associated with u are paired up with balls from random urns containing respectively $d_1+1, d_2+1, \dots, d_{\deg u}+1$ free balls. The probability of pairing up one ball from U with a ball from an urn with $r-i$ free balls is $P_i = \frac{(r-i)Y_i}{X}$ for $i \in \{0,\dots,r\}$ at the beginning of a step, and this only changes by a $o(1)$ factor during each step due to the multiple edge selections which are part of a step. Hence $\Pr[\boldsymbol{d}]$ is asymptotically equal to $\binom{\deg u}{m_1,\dots,m_{\deg u}} P_{r-(d_1+1)} \cdot P_{r-(d_2+1)} \cdot \ldots \cdot P_{r-(d_{\deg u}+1)}$ where m_l are the multiplicities of the possibly $\deg u$ distinct values occurring in $\boldsymbol{d}$.

Conditional expectations. Note that for each problem Π, the sequence $\boldsymbol{d}$ contains all the information needed to compute asymptotic expressions for all the conditional expected changes of the variables Y_i. There will be a function $\mathrm{rm}_{\Pi,r}(i,\boldsymbol{d})$, giving, for each $\boldsymbol{d}$ in a given Phase, the number of vertices of degree $r-i$ removed from the subgraph of G induced by $u \cup N(u)$. Then

$$\mathrm{E}(\Delta Y_i \mid \boldsymbol{d}) = -\mathrm{rm}_{\Pi,r}(i,\boldsymbol{d}) - \mathrm{Spec}_{\Pi,r}(i,\boldsymbol{d})$$

where Spec is a problem specific function depending on a wider neighbourhood around u. In each case Spec may involve $R_i = P_i - P_{i-1}$ for $i \in \{0, \ldots, r\}$ (with $P_{-1} = 0$) where $-R_i$ describes asymptotically the contribution to $\mathrm{E}(\Delta Y_i \mid \boldsymbol{d})$ given by the removal from G of one edge incident to some $v \in N(u)$ (different from $\{u, v\}$), or

$$Q_i = \sum_{k=0}^{r-1} P_k[\delta_{ik} + (r-k-1)R_i], \text{ for } i \in \{0, 1, \ldots, r\},$$

where $-Q_i$ is (asymptotically) the contribution to $\mathrm{E}(\Delta Y_i \mid \boldsymbol{d})$ given by the removal from G of a vertex adjacent to some $v \in N(u)$ (different from u).

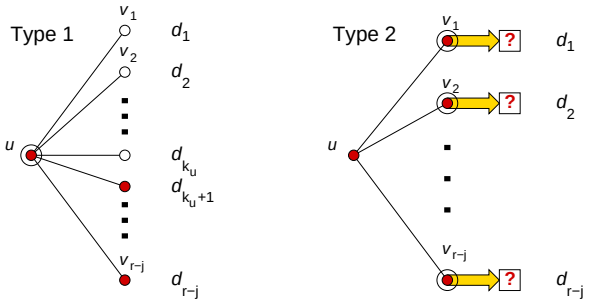
Computational aspects. The performances of algorithm DescendingDegree depend heavily on a good choice of the points c_j. Unfortunately, although in any given Phase j the systems in (4) are sufficiently well-behaved to guarantee the existence of a unique solution in a domain $\mathcal{D}_j \subseteq \{(\sigma, z^{(1)}, \ldots, z^{(a)}) \in [0,1]^{a+1}, \sum_{i=1}^{r-1}(r-i)z^{(i)} \geq \delta\}$ (for some $\delta > 0$), where $a = r$ (resp. $r-1$) for MIDS (for MVC and MEDS), it is not easy to find a closed expression (depending on the c_j's) for the functions y_i^j and consequently optimise the choice of the c_j's. However an alternative approach based on solving the relevant systems numerically using a Runge-Kutta method [8], gives remarkably good results.
Given a positive integer M and an upper bound c_r on σ, let $t_0 = 0$ and $t_h = t_{h-1} + \frac{c_r}{M}$ for all $h \in \{1, \ldots, M\}$. Any $(r-1)$-set $\{s_1, \ldots, s_{r-1}\}$ over $\{1, \ldots, M-1\}$ defines a subsequence of t_h. The system for Phase j is then solved numerically using a variable step Runge-Kutta method in the interval $[0, t_{s_{j+1}} - t_{s_j}]$ (with $s_0 = 0$). The initial values for Phase 0 are $y_0^0(0) = 1$ and $y_i^0(0) = 0$ and the final values for the functions in Phase j are the initial values for the functions in Phase $j+1$, for $j \in \{1, \ldots, r-1\}$. The *cost* of a given $(r-1)$-set $\{s_1, \ldots, s_{r-1}\}$ is defined to be the first index x such that $y_i(t_{x-1}) \neq 0$ and $y_i(t_x) = 0$ for all $i \in \{0, \ldots, r-2\}$ (with the cost being ∞ if the transition from non-zero to zero does not occur for all i at the same time or it does not occur during Phase $r-1$). The parameters c_j are defined as $c_j = t_{\hat{s}_j}$ for $j \in \{1, \ldots, r-1\}$ where $\{\hat{s}_1, \ldots, \hat{s}_{r-1}\}$ is a minimum cost $(r-1)$-set.

4 Specific Problems

In this Section we complete the description of algorithm DescendingDegree by giving full definition of all problem specific quantities. In each case a picture is drawn to represent the neighbourhood of the chosen vertex u when a given configuration $(d_1, d_2, \ldots)$ occurs. The circled vertices (resp. the thick black edges) are added to the output structure in each case. Dark vertices are removed from G at the end of the given step. White vertices have their degree decreased by one. Question-marked square vertices represent vertices whose degree is not probed at the given step (but which nonetheless are affected by the incurring graph transformation). Note that, since graphs in $\mathcal{G}(n, r\text{-reg})$ do not contain small cycles a.a.s. [19, Theorem 9.5], the question marked vertices are a.a.s. all distinct. All edges in the pictures are removed at the end of the given step (the arrows being just a shorthand for a number of edges). A table is included in each of the following sections giving, for the first few values of r, the sequence of values $c_1, \ldots, c_{r-1}$ that allow the algorithm to attain the bounds given in Section 2.

4.1 Vertex Cover

Each of the r systems is formed by r equations, since there is no need to keep track of Y_r. The Figure below shows the two types of configurations around vertex u along with the choices of the vertices added to the cover C.



In a given Phase j, for configurations of Type 1, $\mathrm{rm}_{\mathrm{MVC},r}(i,\boldsymbol{d})$ will have a contribution of one if $i=j$, for the degree of u and a contribution of one for each of the $\deg u - k_u$ vertices that are only connected to u, if $i = r-1$. For configurations of Type 2, all neighbours of u are removed from G. Therefore $\mathrm{rm}_{\mathrm{MVC},r}(i,\boldsymbol{d})$ will have the term $\delta_{i,j}$ and a possible contribution from each neighbour of u in V_{r-i}. In symbols this is

$$\mathrm{rm}_{\mathrm{MVC},r}(i,\boldsymbol{d}) = \begin{cases} \delta_{i,j} + (r-j-k_u)\delta_{i,r-1} & \boldsymbol{d} \in \text{ Type 1} \\ \delta_{i,j} + \sum_{h=1}^{r-j} \delta_{i,r-(d_h+1)} & \boldsymbol{d} \in \text{ Type 2} \end{cases}$$

where $\delta_{xy} = 1(0)$ if $x = y$ (resp. $x \neq y$). Furthermore, for each $i \in \{0, r-1\}$ and $\boldsymbol{d}$ in the given Phase,

$$\mathrm{Spec}_{\mathrm{MVC},r}(i,\boldsymbol{d}) = \begin{cases} \sum_{h=1}^{k_u} (\delta_{i,r-(d_h+1)} - \delta_{i,r-d_h}) & \boldsymbol{d} \in \text{ Type 1} \\ R_i \sum_{h=1}^{r-j} d_h & \boldsymbol{d} \in \text{ Type 2} \end{cases}$$

For configurations of Type 1 a vertex of degree $r-i$ is created (resp. removed) if the h-th neighbour (of degree at least two) of u has indeed degree $r-i+1$ (resp. $r-i$) before the current step takes place. This event is accounted for by the term $(\delta_{i,r-(d_h+1)} - \delta_{i,r-d_h})$. For configurations of Type 2, $R_i \sum_{h=1}^{r-j} d_h$ accounts for all the vertices of degree $r-i$ that are either created or removed when one of the edges incident to some $v \in N(u)$ is removed. Finally, the expected change in the size of the cover is obtained from (2) setting $c_{\mathrm{MVC}} = r-j$, since one vertex (resp. $r-j$ vertices) is (are) added to C if a configuration of Type 1 (resp. Type 2) is met.

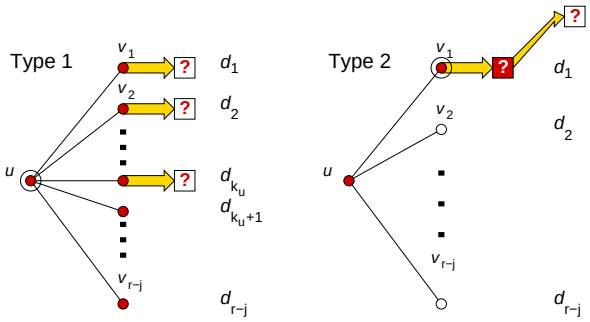
Note that the algorithm DescendingDegree$_{\mathrm{MVC},r}$ can be easily modified to output the independent set $V \setminus C$. Therefore, as announced previously, all results for the MVC problem translate immediately to results for the MIS problem.

r	c_1	c_2	c_3	c_4	c_5	c_6
3	0.001624	0.336168				
4	0.002294	0.045756	0.479508			
5	0.002145	0.01443	0.175305	0.5694		
6	0.00816	0.01972	0.08636	0.29036	0.60452	
7	0.00686	0.02107	0.06517	0.17647	0.37751	0.68754

4.2 Independent Dominating Set

The Figure below shows the two types of configurations around vertex u along with the choices of the vertices to add to the dominating set D. The function $\mathrm{rm}_{\mathrm{MIDS},r}(i, \boldsymbol{d})$ is defined as follows

$$\mathrm{rm}_{\mathrm{MIDS},r}(i, \boldsymbol{d}) = \begin{cases} \delta_{i,j} + \sum_{h=1}^{r-j} \delta_{i,r-(d_h+1)} & \boldsymbol{d} \in \text{ Type 1} \\ \delta_{i,j} + \delta_{i,r-(d_1+1)} & \boldsymbol{d} \in \text{ Type 2} \end{cases}$$



When applied to the MIDS problem, the algorithm DescendingDegree may generate a number of isolated vertices that are not covered by a vertex in D. This may happen at any of the white question marked vertices in the Figure above. The number of these vertices must be accounted for in assessing the quality of a solution. Therefore one needs to keep track of the conditional expected changes in the variables Y_i, for $i \in \{0, \ldots, r\}$, where Y_r counts the number of uncovered isolated vertices. Let $\mathrm{rm}_{\mathrm{MIDS},r}(r, \boldsymbol{d}) = 0$. For each $i \in \{0, \ldots, r\}$ and each $\boldsymbol{d}$ in the given Phase, function $\mathrm{Spec}_{\mathrm{MIDS},r}$ takes the following expressions

$$\mathrm{Spec}_{\mathrm{MIDS},r} = \begin{cases} R_i \sum_{h=1}^{r-j} d_h & \boldsymbol{d} \in \text{ Type 1} \\ \sum_{h=2}^{r-j} (\delta_{i,r-(d_h+1)} - \delta_{i,r-d_h}) + d_1 Q_i & \boldsymbol{d} \in \text{ Type 2} \end{cases}$$

Finally the expected change in the size of the dominating set is given by (2) with $c_{\mathrm{MIDS}} = 1$.

r	c_1	c_2	c_3	c_4	c_5	c_6
3	0.00015	0.0558817				
4	0.0005	0.0062501	0.079565			
5	0.00184	0.00552	0.0220828	0.0858274		
6	0.0014	0.0056	0.0098	0.0378094	0.0862792	
7	0.002856	0.0057122	0.0085692	0.0229683	0.0474604	0.0838873

Theorem 2. *For each $r \geq 3$ there exists a constant $\lambda > 0$ such that $\gamma(G) > \lambda n$ for $G \in \mathcal{G}(n, r\text{-reg})$ a.a.s.*

Proof. The expected number of independent dominating sets of size s in a random pairing on n urns each containing r balls is

$$\binom{n}{s}(sr)![x^{sr}]((1+x)^r - 1)^{n-s} \frac{N([r(n-2s)]/2)}{N(nr/2)}$$

where $N(x) = \frac{(2x)!}{x!\,2^x}$ and $(sr)![x^{sr}]((1+x)^r - 1)^{n-s}$ counts the ways of pairing the sr balls in s chosen urns in such a way that each of the remaining $n-s$ urns is connected to at least one of the urns in the chosen set (if $p(x)$ is a polynomial then $[x^k]p(x)$ is the standard notation of the coefficient of the monomial of degree k). This number is at most $(sr)!\frac{((1+x)^r-1)}{x^{sr}}^{n-s}$. Using Stirling's approximation to the factorial and setting $s = \lambda n$ implies that the expectation above is at most

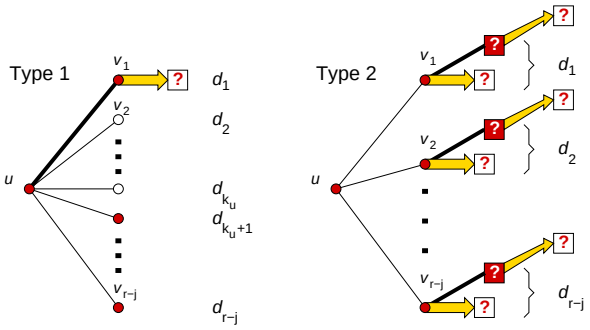
$$n^{O(1)} \left\{ \frac{(1-2\lambda)^{\frac{r(1-2\lambda)}{2}} (\lambda r)^{\lambda r}}{\lambda^{\lambda}(1-\lambda)^{1-\lambda} r^{r\lambda}} \frac{((1+x)^r-1)}{x^{\lambda r}}^{1-\lambda} \right\}^n$$

where x takes the value that minimises $\frac{((1+x)^r-1)}{x^{\lambda r}}^{1-\lambda}$. For every r there exists a positive $\lambda = \lambda(r)$ such that the expected number of independent dominating sets of size $s < \lambda n$ in a random pairing tends to zero and therefore, by the Markov inequality, with high probability a random r-regular graph will not contain an independent dominating set of this size. The values of λ are reported in the second column of the table for MIDS in Section 2. □

4.3 Edge Dominating Set

The Figure below shows the two types of configurations around vertex u along with the choices of the edges to add to the edge dominating set M. The function $\mathrm{rm}_{\mathrm{MEDS},r}(i, \boldsymbol{d})$ is defined as follows

$$\mathrm{rm}_{\mathrm{MEDS},r}(i, \boldsymbol{d}) = \begin{cases} \delta_{i,j} + \delta_{i,r-(d_1+1)} + (r-j-k)\delta_{i,r-1} & \boldsymbol{d} \in \text{ Type1} \\ \delta_{i,j} + \sum_{h=1}^{r-j} \delta_{i,r-(d_h+1)} & \boldsymbol{d} \in \text{ Type2} \end{cases}$$



Also, for each $i \in \{0, \ldots, r-1\}$ and $\boldsymbol{d}$ in the given Phase, let

$$\mathrm{Spec}_{\mathrm{MEDS},r}(i, \boldsymbol{d}) = \begin{cases} \sum_{h=2}^{k_u} (\delta_{i,r-(d_h+1)} - \delta_{i,r-d_h}) & \boldsymbol{d} \in \text{ Type 1} \\ R_i \sum_{h=1}^{r-j} (d_h - 1) + Q_i(r-j) & \boldsymbol{d} \in \text{ Type 2} \end{cases}$$

(with the convention that the sum in the first line is empty if $k_u = 1$).
Finally, the expected change in the size of the edge dominating set is obtained from formula (2) setting $c_{\mathrm{MEDS}} = r - j$, since one vertex (resp. $r-j$ vertices) is (are) added to M if a configuration of Type 1 (resp. Type 2) is met. Again we solve r systems of r equations in the variables $y_0, \ldots, y_{r-1}$.

r	c_1	c_2	c_3	c_4	c_5	c_6
3	0.00054	0.123696				
4	0.00038	0.0152	0.22078			
5	0.008848	0.015408	0.06864	0.26692		
6	0.003485	0.010045	0.033005	0.12505	0.28946	
7	0.0066	0.0154	0.0286	0.0814	0.1694	0.33022

5 Conclusions

In this paper we presented a new approach for the approximate solution to minimisation problems in random regular graphs. We defined a greedy algorithm that, in a number of successive Phases, picks vertices of given degree in the current graph, updates the partial solution produced so far and removes few vertices from the graph. Although in principle it is not clear that all Phases (as defined in Section 3) can be successfully completed, one after the other, we prove that, for the problems at hand under a suitable choice of the algorithm's parameters this is indeed the case. This in turns leads to improved approximation performances in almost all cases considered.

Many questions are left open by this work. First of all, the results in the paper rely on a numerical integration procedure based on a second order Runge-Kutta method. A tedious analysis of the first few partial derivatives of the functions defining the systems in each case would give a quantitative measure of their stability and a-priori bounds on the quality of the numerical approximation. Esperimentally, successive runs of the integration procedure with ever decreasing step sizes confirmed that our results are accurate to the first four decimal digits.

Secondly, no serious attempt was made to optimise the values of the constants c_j. The value of M was set to 10000 for $r = 3$ but to much smaller values for all other r. Slight improvements on the values of σ for all problems may be possible by using a more careful search through the various multi-dimensional parameter spaces.

The algorithm performances could also be improved by expanding the neighbourhood of the vertex considered at each step or analysing more carefully the set of configurations in each Phase. The price for this will be an increased complexity in finding the problem specific parameters $c_1, c_2, \ldots, c_{r-1}$.

Finally, in principle, the same approach can be exploited in other random graph models, such as $\mathcal{G}(n, p)$ or $\mathcal{G}(n, m)$ or even in the design of greedy algorithms for solving the same minimisation problems making no assumption on the input distribution.

References

1. P. Alimonti, T. Calamoneri. Improved approximations of independent dominating set in bounded degree graphs. In *Proc. 22nd WG*, pp 2-16, LNCS 1197, Springer-Verlag, 1997.
2. P. Alimonti, V. Kann. Hardness of approximating problems on cubic graphs. In *Proc. 3rd CIAC*, pp 288-298. LNCS 1203, Springer-Verlag, 1997.
3. J. Aronson, A. Frieze, B. G. Pittel. Maximum matchings in sparse random graphs: Karp-Sipser revisited. *RSA*, 12:111-178, 1998.

4. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. *Complexity and Approximation*. Springer-Verlag, 1999.
5. P. Berman, T. Fujito. Approximating independent sets in degree 3 graphs. In *Proc. WADS'95*, pp 449-460. LNCS 955, Springer-Verlag, 1995.
6. P. Berman, M. Karpinski. On some tighter inapproximability results. Technical Report TR98-29, ECCC, 1998.
7. B. Bollobás, P. Erdős. Cliques in random graphs. *Math. Proc. Camb. Phil. Soc.*, 80:419-427, 1976.
8. R. L. Burden, J. D. Faires, A. C. Reynolds. *Numerical Analysis*. Wadsworth Int., 1981.
9. P. Crescenzi, V. Kann. A compendium of NP optimization problems. Available at `http://www.nada.kth.se/~viggo/wwwcompendium/`, 2000.
10. W. Duckworth, N. C. Wormald. Linear programming and the worst case analysis of greedy algorithms for cubic graphs. To be submitted. Pre-print available from the authors.
11. W. Duckworth, N. C. Wormald. Minimum independent dominating sets of random cubic graphs. Submitted to RSA, 2000.
12. A. Frieze, C. McDiarmid. Algorithmic theory of random graphs. *RSA*, 10:5-42, 1997.
13. A. M. Frieze. On the independence number of random graphs. *Disc. Math.*, 81:171-175, 1990.
14. A. M. Frieze, T. Łuczak. On the independence and chromatic number of random regular graphs. *J. Comb. Theory*, B 54:123-132, 1992.
15. M. R. Garey, D. S. Johnson. *Computer and Intractability, a Guide to the Theory of NP-Completeness*. Freeman & Company, 1979.
16. M. R. Garey, D. S. Johnson, L. Stockmeyer. Some simplifi ed *NP*-complete graph problems. *TCS*, 1:237-267, 1976.
17. G. R. Grimmett, C. J. H. McDiarmid. On colouring random graphs. *Math. Proc. Camb. Phil. Soc.*, 77:313-324, 1975.
18. D. S. Hochbaum. Easy solutions for the k-center problem or the dominating set problem on random graph. In *Annals of Disc. Math.*, 25:189-210, 1985.
19. S. Janson, T. Łuczak, A. Rucínski. *Random Graphs*. John Wiley & Sons, 2000.
20. V. Kann. *On the Approximability of NP-complete Optimization Problems*. PhD thesis, Royal Institute of Technology, Stockholm, 1992.
21. B. Korte, D. Hausmann. An analysis of the greedy heuristic for independence systems. *Annals of Disc. Math.*, 2:65-74, 1978.
22. B. D. McKay. Independent sets in regular graphs of high girth. *Ars Combinatoria*, 23A:179-185, 1987.
23. T. Nierhoff. *The k-Center Problem and r-Independent Sets*. PhD thesis, Humboldt-Universitt zu Berlin, Berlin, 1999.
24. R. W. Robinson, N. C. Wormald. Almost all regular graphs are hamiltonian. *RSA*, 5:363-374, 1994.
25. N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Prob.*, 5:1217-1235, 1995.
26. N. C. Wormald. The differential equation method for random graph processes and greedy algorithms. In M. Karoński, H. J. Prömel, editors, *Lectures on Approximation and Randomized Algorithms*, pages 73–155. PWN, Warsaw, 1999.
27. M. Yannakakis, F. Gavril. Edge dominating sets in graphs. *SIAM J. Applied Math.*, 38(3):364-372, June 1980.
28. M. Zito. *Randomised Techniques in Combinatorial Algorithmics*. PhD thesis,University of Warwick, 1999.
29. M. Zito. Small maximal matchings in random graphs. In *Proc. LATIN 2000*, pp 18-27. LNCS 1776, Springer-Verlag, 2000.