Ultimate greedy approximation of independent sets in subcubic graphs

Piotr Krysta* Mathieu Mari† Nan Zhi‡

Abstract

We study the approximability of the maximum size independent set (MIS) problem in bounded degree graphs. This is one of the most classic and widely studied NP-hard optimization problems. It is known for its inherent hardness of approximation.

We focus on the well known minimum degree greedy algorithm for this problem. This algorithm iteratively chooses a minimum degree vertex in the graph, adds it to the solution and removes its neighbors, until the remaining graph is empty. The approximation ratios of this algorithm have been very widely studied, where it is augmented with an advice that tells the greedy which minimum degree vertex to choose if it is not unique.

Our main contribution is a new mathematical theory for the design of such greedy algorithms with efficiently computable advice and for the analysis of their approximation ratios. With this new theory we obtain the ultimate approximation ratio of 5/4 for greedy on graphs with maximum degree 3, which completely solves the open problem from the paper by Halldórsson and Yoshihara (1995). Our algorithm is the fastest currently known algorithm with this approximation ratio on such graphs. We also obtain a simple and short proof of the (D+2)/3-approximation ratio of any greedy on graphs with maximum degree D, the result proved previously by Halldórsson and Radhakrishnan (1994). We almost match this ratio by showing a lower bound of (D+1)/3 on the ratio of any greedy algorithm that can use any advice. We apply our new algorithm to the minimum vertex cover problem on graphs with maximum degree 3 to obtain a substantially faster 6/5-approximation algorithm than the one currently known.

We complement our positive, upper bound results with negative, lower bound results which prove that the problem of designing good advice for greedy is computationally hard and even hard to approximate on various classes of graphs. These results significantly improve on such previously known hardness results. Moreover, these results suggest that obtaining the upper bound results on the design and analysis of greedy advice is non-trivial.

1 Introduction

Given a graph G, an *independent set* in G is a subset of the set of its vertices such that no two of these vertices are connected by an edge in G. The problem of finding an independent set of maximum size in a graph, the Maximum Independent Set problem (MIS), is one of the most fundamental NP-hard combinatorial optimization problems. A polynomial time algorithm for MIS is an r-approximation algorithm if it finds an independent set in the input graph of size at least opt/r, where opt is the size of the maximum size independent set, and $r \geq 1$ is an *approximation ratio*, *guarantee* or *factor*.

We study MIS on graphs with maximum degree bounded by Δ, degree-Δ graphs. This problem is known for its inherent hardness of approximation guarantee. Even if $\Delta = 3$, MIS is known to be APX-hard, see [1]. As Δ grows, a stronger asymptotic hardness of approximation of $\Omega(\Delta/\log^4 \Delta)$ is known, unless P = NP [6]. The best known polynomial time approximation ratio for this problem for small $\Delta \geq 3$ is arbitrarily close to $\frac{\Delta+3}{5}$, see [3, 4, 7]. It is a local search approach with huge running time, e.g., n^{50} [12], where n is the number of vertices in the graph. We are primarily interested in MIS on graphs with small to moderate Δ and therefore not asymptotic.

Probably the best known method to find large independent sets is the *minimum degree greedy* method, which repeatedly chooses a minimum degree vertex in the current graph as part of the solution and deletes it and its neighbors until the remaining graph is empty. This algorithm is extremely simple and time-efficient. The first published approximation guarantee $\Delta + 1$ of the greedy for MIS we are aware of can be inferred from the proof of a conjecture of Erdős, by Hajnal and Szemerédi [9, 2]. The best known analysis of greedy by Halldórsson and Radhakrishnan [13, 11] for MIS implies the approximation ratio of $(\Delta + 2)/3$, and better ratios are known for small Δ.

Halldórsson and Yoshihara [14] asked the following fundamental question: what is the power of the greedy algorithm when we augment it with an *advice*, i.e., a fast method that tells the greedy which minimum degree vertex to choose if there are many? They, e.g., proved that no advice implies better than 5/4-approximation of greedy for MIS with $\Delta = 3$. On the other hand, they provide an advice for greedy leading to a 3/2-approximation, and this is the best known to date greedy approximation for MIS when $\Delta = 3$, that is in

*Department of Computer Science, University of Liverpool, Liverpool, UK. E-mail: pkrysta@liverpool.ac.uk. The work of this author has been partially supported by the Network Sciences and Technologies (NeST) initiative of the University of Liverpool.

†Department of Computer Science, École Normale Supérieure, Paris, France. E-mail: mathieu.mari@ens.fr.

‡Department of Computer Science, University of Liverpool, Liverpool, UK. E-mail: n.zhi@liverpool.ac.uk.

subcubic graphs.[1]

Our new positive results: upper bounds. Our main technical contribution are new payment schemes for proving improved/tight approximation ratios of greedy with advice for MIS on degree-Δ graphs. As a warm-up, we first apply these techniques to MIS on degree-Δ graphs to obtain:

- A simple and short proof of the $(\Delta + 2)/3$-approximation ratio of any greedy algorithm (i.e., without any advice), the result proved previously by Halldórsson and Radhakrishnan [13, 11]. We extend a lower bound construction from [13] to prove that any greedy algorithm (with any, even exponential time, advice) has an approximation at least $(\Delta + 1)/3 - O(1/\Delta)$.

- A simple proof of the $(\Delta+6)/4$-approximation ratio of any greedy algorithm on triangle-free degree-Δ graphs, improving the previous best known greedy ratio of $\Delta/3.5 + O(1)$ [11] for MIS on such graphs. Compared to the proof in [11] our proof is extremely simple and short.

As we cannot obtain much better ratios than $(\Delta + 2)/3$ by using any advice for greedy, we focus on small Δ. We have to develop our payment schemes significantly more compared to the above applications, and obtain the following results for MIS and Minimum Vertex Cover (MVC) problems:

- We completely resolve the open problem from Halldórsson and Yoshihara [14] and design a fast, ultimate advice for greedy obtaining a 5/4-approximation, i.e., the best possible greedy ratio for subcubic MIS. A lower bound of 5/4 on the ratio of greedy with any advice on such graphs was proved in [14], and the best previously known ratio of greedy was 3/2 [14]. Our new greedy 5/4-approximation algorithm has running time $O(n^2)$, where n is the number of vertices. The best known algorithm for this problem is a local search 6/5-approximation algorithm of Berman and Fujito [3], with running time no less than n^{50} [12]. If the approximation ratio of this local search algorithm is fixed to 5/4, then the running time is $n^{18.27}$ [7].

- We obtain a greedy 4/3-approximation algorithm for MIS on subcubic graphs with linear running time. By using our payment scheme, we can also provide a simple proof of a 3/2-approximation ratio of the greedy algorithm called MoreEdges in [12], which was the best previously known approximation ratio of greedy for MIS on subcubic graphs.

- We obtain an $O(n^2)$-time 6/5-approximation for the subcubic MVC problem. To compare, the best known algorithm is a 7/6-approximation with run time of at least n^{50} [12]. Even obtaining the 6/5-approximation for MVC on subcubic graphs required time of $n^{18.27}$ [7].

Our new negative results: lower bounds. We complement our upper bound results with hardness, lower bounds, results which suggest that our upper bounds on the design of good greedy advice are essentially (close to) best possible, or non-trivial computational problems. We believe this also suggests that the design of good greedy advice is a non-trivial task on its own.

To prove our lower bounds, we study the complexity of computing a good advice for the greedy algorithm for MIS. Towards this goal, Bodlaender et al. [5] defined a problem MaxGreedy, which given an input graph asks to find the largest independent set obtained by any greedy algorithm. Thus, MaxGreedy asks for the best advice for greedy. They proved that computing an advice which finds an r-approximate solution to the MaxGreedy problem is co-NP-hard for any fixed $r \geq 1$ and NP-complete for $r = 1$ [5]. We significantly improve the previous hardness results for MaxGreedy:

- We prove that the MaxGreedy problem is NP-complete even on cubic planar graphs, significantly strengthening the NP-completeness result by Bodlaender et al. [5] who prove it on arbitrary graphs. This result suggests that the problem of design/analysis of good advice for greedy even on cubic planar graphs is difficult.

- We prove that MaxGreedy is NP-hard to approximate to within a ratio of $n^{1-\varepsilon}$ for any $\varepsilon > 0$, and hard to approximate to within $n/\log n$ under the Exponential Time Hypothesis. We also prove that MaxGreedy remains hard to approximate to within $(\Delta+1)/3 - O(1/\Delta) - O(1/n)$ on degree-Δ graphs, nearly matching the approximation ratio $(\Delta+2)/3$ of the greedy algorithm.

- We prove that the MaxGreedy problem remains hard to approximate on bipartite graphs. This is quite interesting because the MIS problem itself is polynomially solvable on bipartite graphs.

We also extend a lower bound construction of Halldórsson and Radhakrishnan [13] to prove that any greedy algorithm (with any, even exponential time, advice) has an approximation ratio at least $(\Delta+1)/3 - O(1/\Delta)$ on degree-Δ graphs. We note here that due

[1]They claimed an improved ratio of 9/7 by a greedy like algorithm but they have retracted this result, see [10].

to the lack of space we have only included the precise statements of our lower bound results without proofs in this proceedings version. Likewise, proofs of some of our upper bounds results have not been included.

Our technical contributions. Our main technical contributions are a class of potential functions and payment schemes, and an inductive proof technique, used to pay for solutions of greedy algorithms for MIS. These techniques lead to very precise and tight analyses of the approximation ratios of greedy algorithms. Here we will explain intuitions about our proof of the 5/4-approximation ratio of the greedy algorithms on subcubic graphs, which uses our full technical machinery. Let G be an input graph with an optimal independent set OPT. Greedy algorithm executes *reductions* on G, i.e., a reduction is to pick a minimum degree vertex v in the current graph (*root* of the reduction) into the solution and remove its neighbors, see example reductions in Figure 1. Suppose the first reduction executed by greedy is R and it is *bad*: its root v has degree 2, $v \notin OPT$ and both neighbors of v are in OPT. Then, locally, the approximation ratio is 2. To bring the approximation ratio down to 5/4, we must prove that, in the future, there will exist equivalent of at least three reductions, called *good*, each of which adds one vertex to the solution and removes only one vertex from OPT. For each executed bad reduction, there must exists an unique (equivalent of) three good ones.

Consider, for example, instances H_{i+1} of MIS in Figure 4, due to Halldórsson and Yoshihara [14], where the base graph H_0 is H_0', and black vertices are in OPT, and white not. Any greedy executes on this instance many bad reductions, but only at the end, good reductions, triangles, are executed. There is just enough good reductions to uniquely map three of those to any executed bad reduction (there is exactly one unused good reduction). This shows a lower bound of 5/4 on the ratio of any greedy when $i \longrightarrow +\infty$. Thus, the "payment" for bad reductions arrives, but very late! Such a "payment" may not only be late, but such "good" reductions might be very irregularly distributed. For instance, suppose that the first reduction in H_0' on Figure 4, let us call it R, has two of its *contact* edges (i.e., the four edges going down from R's two black vertices) going to an identical white vertex, creating a follow up reduction of degree one. Then that degree one reduction is good and when executed, it can immediately (partially) pay for the bad reduction R.

Question: How do we prove an existence of such a highly non-local and irregular payment scheme? We will define a special potential of a reduction, in such a way that each executed reduction can be "almost paid for" locally. Thus, at every point in time we will keep the potential of each connected component of at least -1. For example in the instance from Figure 4 the execution of the first bad reduction in H_{i+1} creates 4 connected components H_i and then greedy executes reductions in each H_i independently. We will have an intricate inductive argument, see Lemma 11, showing that an execution of reductions in a connected graph will have the potential at least -1. In the induction step, some reductions R may create components with potential -1. In such cases when we cannot keep potential at least -1, we will make sure that before the execution of R, such components contain reductions with higher greedy priority than that of R, leading to a contradiction.

This process is complicated by the fact that vertices can be *black* (in OPT) and *white* (outside of OPT) and whether a reduction is "bad" depends on the distribution of black/white vertices in the reduction. For instance, a reduction like the first one in graph H_0' on Figure 4 might have a black root and thus the two root's neighbors will be white. Such a reduction will in fact be "good" when executed. The final value of -1 of the potential will be paid by the "past".

Intuitively, our potential will imply that we can ship the payments from good reductions executed by greedy *anywhere* in the graph into the places where bad reductions need those payments. Such a shipment is unique, in the above sense that there exists (equivalent) 3 good reductions for each bad reduction. We will realize this shipment by deferring the need of payment into the future along edges, *contact edges*, which are incident to the neighbors of the reduction's root. These contact edges created by a bad reduction R will be called *loan* edges. Each loan edge e created by R will have a "dual" edge (identical to e), called a *debt* edge, which will be inherited by the future reductions directly created by R via its contact edges. Loan contact edges will help us "predict" the future graph structure. And debt edges enable us to keep track of the past reductions.

Our complete theory that realizes such precise payment scheme allows us to achieve a very interesting kind of result, namely, to (essentially) characterize all graphs that can have negative potential! See Definition 10 and Lemma 11. This leads to an extremely tight analysis, up to an additive unit in the following sense. We prove that a version of our 5/4-approximate greedy finds a solution of size at least $\frac{4}{5}|OPT| + \frac{1}{5}$ in subcubic graphs, and when run on the lower bound instances of Halldórsson and Yoshihara [14], it finds a solution of size precisely $\frac{4}{5}|OPT| + \frac{1}{5}$. An unusual aspect of our result is that we can prove that *any* lower bound example showing exact tightness of our guarantee $\frac{4}{5}|OPT| + \frac{1}{5}$ must be an infinite family of graphs, see remark after the proof of Theorem 7.

Differences between our proof and Halldórsson and Yoshihara [14]. According to our potential, there are two kinds of reductions that are particularly problematic to deal with. These are odd isolated cycles with maximum independent sets in them, and reductions like reduction (d) in Figure 2, which we call an odd-backbone reduction. Their potential is -1 (for the cycle it can also be -2, but we can prevent that case). This means that each such reduction when executed would need a unit of payment originated from some good reduction. Consider an instance H_i in Figure 4 when i tends to ∞ (with the base graph H_0). Suppose that greedy executes the top bad reduction and then recursively executes the following four created bad reductions. Then at the very end it will reach a collection of 7-cycles and each such cycle will need a payment of 1. As we see this can only lead to a ratio $17/13 > 5/4$. Already on any odd cycle, the potential of [14] tells us that it actually needs a payment of 2 (which is one unit more than our potential); we mention here however that it is not possible to pay 2 units to such odd cycles. Our approach is to either prevent greedy from ever ending up with such isolated problematic odd cycles or to show that we can actually pay for such cycles in some cases. The key to a solution is to carefully prioritize certain reductions that would "break" the cycle before it becomes isolated, or to pay for it when there is a spare reduction that can do so. For the bad odd-backbone reductions in Figure 2(d), observe that we could wisely execute them on a black degree-2 vertex which would make them good. But then how do we know which of these two adjacent degree-2 vertices is black/white? One way, pursued in [14], is to try to pay for such odd cycles or odd-backbone reductions by some kind of local analysis which tries to collect locally good reductions that can pay. We can show that such local analysis/payment is not possible and a global payment or explicit exclusion of such reductions are necessary. Instead, what we do is to impose a special greedy order on such odd-backbone reductions, and with this order we prove that we can pay for them whenever they are executed as bad reductions. The source of these payments, however, is non-local and our scheme proves their unique existence.

To achieve the above payments or avoid bad cycles, we introduce a powerful analysis tool which is a special kind of reductions, called *black* and *white* reductions, see Definition 9. We also introduce an inductive process to argue about existence of such reductions in Lemma 11. These techniques let us prove that when a reduction, say R, that cannot pay is executed, there will exist a strictly higher priority reduction (black or white) in the graph before the execution of R, leading to a contradiction with the greedy order. This argument is quite delicate because their existence depends crucially on what kind of contact edges R has. But it also depends on the previously executed reductions.

Our potential definition is in perfect harmony with our inductive proof, that the potential can be kept at least -1. This lets us link the potential directly to the graph structure of the reductions, see Definition 10. This lets us characterize the potentially problematic graphs, i.e., with negative potential, which is the core of our proof. The main tool that helps us in this task is our Inductive Low-debt Lemma 11, which enables us to design the greedy order and characterizes the problematic graphs.

We have managed to prove the existence of appropriate payments coming from good reductions by using only "local" inductive arguments, but our payment scheme is inherently non-local. This means that the payment from good reductions, can be very far from bad reductions they pay for.

This version of the paper contains, among other results, the full description of our new theory outlined above with full proofs of our main positive result of the 5/4-approximation of greedy for subcubic MIS.

Definitions and preliminaries. Given a graph $G = (V, E)$, we also denote $V(G) = V$ and $E(G) = E$. For a vertex $v \in V$, let $N_G(v) := \{u \in V \mid uv \in E\}$ and $N_G[v] := N_G(v) \cup \{v\}$ denote respectively the open and closed neighborhood of v in G. The degree of v in G denoted $d_G(v)$ is the size of its open neighborhood. More generally, we define the closed (*resp.* open) neighborhood of a subset $S \subseteq V$ as the union of all closed (*resp.* open) neighborhoods of each vertex in S.

A graph is called *subcubic* or *sub-cubic* if its maximum degree is at most 3. If the degree of each vertex in a graph is exactly 3 then it is called *cubic*. Given an independent set I in G, we call *black* vertex a vertex v in I and a *white* vertex otherwise. We denote by $\alpha(G)$ the independence number of G, that is the number of black vertices when I is of maximum size in G.

2 Greedy

The greedy algorithm, called a *basic greedy*, or just **Greedy**, on a graph $G = (V, E)$ proceeds as follows. It starts with an empty set S. While the graph G is non empty, it finds a vertex v with minimum degree in the remaining graph, adds this vertex to S and removes v and its neighbors from G. It is clear that at the end, S is an independent set. Let $S = \{v_1, \ldots, v_k\}$ be the ordered output. Let G_i denote the graph after removing vertex v_i and its neighboring vertices. More precisely, $G_0 = G$ and $G_i = G[V \setminus N_G[\{v_1, \ldots, v_i\}]]$, where v_i is a vertex in G_{i-1} that satisfies $d_{G_{i-1}}(v_i) = \min\left\{d_{G_{i-1}}(v) : v \in V(G_{i-1})\right\}$.

Each iteration of the algorithm is called a *basic reduction*, denoted by R_i, which can be described by a pair (v_i, G_{i-1}). An *execution* $\mathcal{E} := (R_1, \ldots, R_k)$ of our greedy algorithm is the ordered sequence of basic reductions performed by the algorithm.

To analyse an execution, we will only require local information for each basic reduction. Given a basic reduction $R_i = (v_i, G_{i-1})$, we call v_i its *root vertex*, its neighbors the *middle vertices*, and together they form the *ground* of the reduction, namely the set of vertices which are removed when the reduction is executed, written $ground(R_i)$. Vertices at distance two from the root are the *contact vertices*. The set of contact vertices is denoted by $contact(R_i)$. Then, the edges between middle and contact vertices are called *contact edges*.

From now on, we will consider that two basic reductions $R = (v, G)$ and $R' = (v', G')$ are isomorphic if there exists a one-to-one function $\phi : N_G[v] \longrightarrow N_{G'}[v']$ such that $\phi(v) = v'$, u and w are adjacent in G if and only if $\phi(u)$ and $\phi(w)$ are adjacent in G', and if each middle vertex u is incident in G to the same number of contact edges than $\phi(u)$ in G'. Finally, the *degree* of a basic reduction is defined as the degree of its root vertex.

Figure 1 presents a table of all possible basic reductions of degree at most two in sub-cubic graphs. Notice that the middle vertices must have degrees equal to or greater than the degree of the reduction.

2.1 Potential function of reductions Suppose that we are given an independent set I in a connected graph $G = (V, E)$ and an execution $\mathcal{E} = (R_1, \ldots, R_k)$ of a greedy algorithm on the input graph G. This execution is associated to a decreasing sequence of sub-graphs: $G = G_0 \supset \cdots \supset G_k = \emptyset$, where $G_i = G\left[V \setminus \bigcup_{j=1}^{i} ground(R_j)\right]$ is the induced sub-graph of G on the set of vertices $V \setminus \bigcup_{j=1}^{i} ground(R_j)$.

Given a basic reduction $R_i = (v_i, G_{i-1})$, we define *loan edges* of R_i as all contact edges with a white contact vertex. Notice that the middle vertex of a loan edge can either be black or white. The *loan* of reduction R_i, denoted by $loan_I(R_i)$ corresponds to its total number of loan edges.

We also define the *debt* of a *white* vertex in the ground of R_i as the number of times this vertex was incident to a loan edge, let us call it e', of a reduction that was previously executed. Such loan edge e' is also called a *debt* edge of reduction R_i. It turns out that the debt of a white vertex corresponds exactly to the difference between its degree in the original graph G and in the current graph G_{i-1}. Similarly, we define the debt of a reduction as the sum of the debts of the vertices of its ground.

$$debt_{G,I}(R_i) = \sum_{u \in ground(R_i) \setminus I} \left(d_G(u) - d_{G_{i-1}}(u)\right)$$

Given two parameters $\gamma, \sigma \geqslant 0$, we now define the *exact potential* of a reduction R_i, for $1 \leqslant i \leqslant k$, as

$$\Phi_{G,I}(R_i) := \gamma - \sigma \cdot |I \cap ground(R_i)| + loan_I(R_i) - debt_{G,I}(R_i)$$

The *exact potential* of an execution $\mathcal{E} = (R_1, \ldots, R_k)$ is the sum of the exact potentials of all reductions: $\Phi_{G,I}(\mathcal{E}) = \sum_{i=1}^{k} \Phi_{G,I}(R_i)$.

Because the independent set produced by the greedy algorithm is maximal and the total debt and the total loan are equal, we obtain the following property.

Proposition 1. *Given an execution $\mathcal{E} = (R_1, \ldots, R_k)$, we have: $\Phi_{G,I}(\mathcal{E}) = \gamma k - \sigma|I|$.*

Proof. By the definition of the exact potential, this can easily be seen by a simple counting argument. □

Suppose we want to analyse the approximation ratio of a greedy algorithm for a given class of graphs $\mathcal{G}$. Then if we manage to find suitable values γ, σ such that all possible reductions have non negative potential, then a direct corollary of Proposition 1 is that **Greedy** is an (γ/σ)-approximation algorithm in $\mathcal{G}$.

In order to measure the potential of each reduction, we now define a new potential, called simply *potential* that is a lower bound on the exact potential. This lower bound is obtained by supposing that the debt of each white vertex is maximal, or equivalently, that its degree in the original graph was equal exactly to Δ.

$$debt_I(R_i) := \sum_{u \in ground(R_i) \setminus I} \left(\Delta - d_{G_{i-1}}(u)\right) \geqslant debt_{G,I}(R_i)$$

Then we define the *potential* of reduction R_i, which is now independent from the original graph, as

$$\Phi_I(R_i) := \gamma - \sigma \cdot |I \cap ground(R_i)| + loan_I(R_i) - debt_I(R_i)$$

To evaluate the potential of a reduction, we do not need anymore to know the set of reductions previously executed but simply the structure of the graph formed by the vertices at distance two from the root, and also which vertices are black/white, which reduces to a relatively small number of cases. We define the *potential of an execution* similarly. Obviously, this new potential is a lower bound on the exact potential defined previously, and more precisely, we have the following fact.

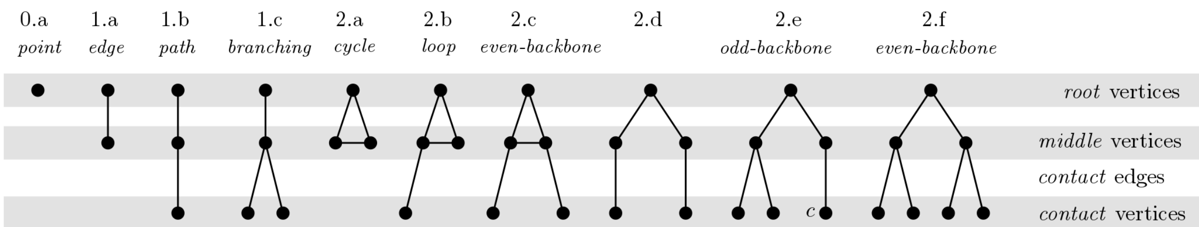

Figure 1: Basic reductions of degree at most two in sub-cubic graphs. We will refer later to these basic reductions by their names, for instance 2.b is a *basic loop reduction*. In this picture, we have drawn contact vertices as distinct vertices, but in a reduction, several contact edges may be incident to the same contact vertex. When the right-most contact vertex c of 2.e has degree three, this reduction is an odd-backbone reduction. Notice that in this case, the middle vertex of degree two is also the root of a basic odd-backbone reduction.

Claim 2. *Let G be a graph with maximum degree Δ, I an independent set in G and $\mathcal{E}$ an execution in G. Then, $\Phi_{G,I}(\mathcal{E}) = \Phi_I(\mathcal{E}) + \sum_{v \notin I}(\Delta - d_G(v))$.*

Proof. By the definition of the two potentials, this can easily be seen by a simple counting argument. □

2.2 Warm-up: New proof of $\frac{\Delta+2}{3}$-ratio for greedy on degree-Δ graphs Halldórsson and Radhakrishnan [13, 11] proved that for any graph with maximum degree Δ, the basic greedy algorithm obtains a $\frac{\Delta+2}{3}$-approximation ratio. In here, we present an alternative proof for the same result, but using our payment scheme. Our proof is simpler and shorter compared to the proof in [13].

Let us use the potential from the previous section, with parameters $\gamma = (\Delta + b)\frac{\Delta+2}{3}$ and $\sigma = \Delta + b$ where $b = 1$ if $\Delta \equiv 2 \pmod 3$, and $b = 0$ otherwise. The choice of the value b is simply to ensure that the potential value is integer. As we remark before, if we can prove that the potential of any reduction is non-negative, then the approximation ratio of **Greedy** in graphs with maximum degree Δ is $\gamma/\sigma = (\Delta + 2)/3$.

Lemma 3. *Let G be a graph with maximum degree Δ. For any basic reduction R and any independent set I we have*

$$\Phi_I(R) := (\Delta + b)\frac{\Delta + 2}{3} - (\Delta + b)\cdot|I \cap ground(R)| + loan_I(R) - debt_I(R) \geqslant 0$$

where $b = 1$ if $\Delta \equiv 2 \pmod 3$, and $b = 0$ otherwise.

Proof. Let $\mathcal{R}$ be the set of all possible basic reductions, and let I be a maximum independent set in the input graph. We note that although there are many types of reductions in $\mathcal{R}$, their structure is highly regular. The idea of the proof is to find the worst type reduction and show that its potential is non-negative. Observe that, if we want to find a reduction R^* to minimize the potential, $R^* = \arg\min_{R\in\mathcal{R}} \Phi_I(R)$, such reduction intuitively needs more debt edges and vertices in I and less loan edges. Also, if v^* is the root of reduction R, then for each $v \in V(R) \setminus \{v^*\}$, if $d_R(v^*) = k$, then $d_R(v) \geq k$, by the greedy rule. For any reduction R, let i be the number of vertices in $I \cap ground(R)$ and let ℓ be the number of vertices in $ground(R) \setminus I$. We have the following formulas:

$$loan_I(R) \geq (i + \ell - 1 - \ell)\cdot i,$$
$$debt_I(R) \leq (\Delta - i - \ell + 1)\cdot \ell.$$

We will justify these bounds now. Let G' be the current graph just before R is executed. Note first that the degree of the root of R is $i + \ell - 1$. The lower bound on $loan_I(R)$ depends on the vertices in I, by the definition. By the greedy order, for each of vertex $v \in I$, $d_{G'}(v) \geq i + \ell - 1$. There are at most ℓ vertices not in I which can be connected to v, thus, the total number of loan edges of v is at least $(i+\ell-1-\ell)$, and we have i such vertices. Note that in this argument we have possibly missed all loan edges that are contact edges of R with both end vertices from $ground(R)\setminus I$. The upper bound on $debt_I(R)$ depends on Δ, the degree of the root vertex and the number vertices not in I. The number of debt edges is at most $\Delta - i - \ell + 1$, as otherwise it violates the greedy order, and we have ℓ vertices not in I.

$$\begin{aligned}\Phi_I(R) &= \frac{\Delta + b}{3}\cdot(\Delta + 2) - (\Delta + b)|I \cap ground(R)| \\ &\quad + loan_I(R) - debt_I(R) \\ &\geq \frac{\Delta + b}{3}(\Delta + 2) - (\Delta + b)i \\ &\quad + (i - 1)i - (\Delta - i - \ell + 1)\ell \\ &= \ell^2 - (\Delta - i + 1)\ell + \frac{\Delta + b}{3}(\Delta + 2) \\ &\quad - (\Delta + b)i + (i - 1)i.\end{aligned}$$

Let $F(\Delta, i, \ell) = \ell^2 - (\Delta - i + 1)\ell + \frac{\Delta+b}{3}(\Delta+2) - (\Delta+b)i + (i-1)i$. Then, the question now is to find the minimum value of $F(\Delta, i, \ell)$ with constrains $\Delta, i, \ell \in \mathcal{Z}^+ \cup \{0\}$. We will first prove that $F(\Delta, i, \ell) \geq b/3 - b^2/3 - 1/3$ for any $\Delta, i, \ell \in \mathcal{R}^+ \cup \{0\}$. For any fixed Δ and i let us treat the function $F(\Delta, i, \ell)$ as a function of ℓ. We know that it is a parabola with the global minimum at point ℓ such that $\frac{\partial F}{\partial \ell} = 0$, which gives us that $\ell = (\Delta - i + 1)/2$. Plugging $\ell = (\Delta - i + 1)/2$ into $F(\Delta, i, \ell)$, we obtain the following function

$$
\begin{aligned}
F(\Delta, &i, (\Delta - i + 1)/2) = F(\Delta, i) \\
&= -\frac{1}{4}(\Delta - i + 1)^2 + \frac{\Delta + b}{3}(\Delta + 2) \\
&\quad - (\Delta + b)i + (i - 1)i \\
&= \frac{3}{4}i^2 - (\Delta/2 + 1/2 + b)i + \frac{\Delta + b}{3}(\Delta + 2) \\
&\quad - \frac{1}{4}\Delta^2 - \frac{1}{2}\Delta - \frac{1}{4}.
\end{aligned}
$$

Similarly as above for any fixed Δ, we see that the function $F(\Delta, i) = \frac{3}{4}i^2 - (\Delta/2 + 1/2 + b)i + \frac{\Delta+b}{3}(\Delta + 2) - \frac{1}{4}\Delta^2 - \frac{1}{2}\Delta - \frac{1}{4}$ as a function of i is a parabola with the global minimum for i such that $\frac{\partial F}{\partial i} = 0$, which gives us that $i = \frac{2}{3}(\frac{\Delta}{2} + \frac{1}{2} + b)$. Plugging $i = \frac{2}{3}(\frac{\Delta}{2} + \frac{1}{2} + b)$ in $F(\Delta, i)$ we obtain the following:

$$
F\left(\Delta, \frac{2}{3}\left(\frac{\Delta}{2} + \frac{1}{2} + b\right)\right) = F(\Delta) = \frac{b}{3} - \frac{b^2}{3} - \frac{1}{3}.
$$

From the above we have that $F(\Delta, i, \ell) \geq b/3 - b^2/3 - 1/3$ for any $\Delta, i, \ell \in \mathcal{R}^+ \cup \{0\}$.

Now, let us observe that if $\Delta \equiv 0, 1 \pmod 3$, then $F(\Delta, i, \ell)$ with $b = 0$ is an integer whenever Δ, i and ℓ are integers. This means that in those cases we have $F(\Delta, i, \ell) \geq -1/3$ which implies that $F(\Delta, i, \ell) \geq 0$. In case when $\Delta \equiv 2 \pmod 3$, we have that $F(\Delta, i, \ell)$ with $b = 1$ is an integer whenever Δ, i and ℓ are integers. This again means that in those cases $F(\Delta, i, \ell) \geq -1/3$, again implying $F(\Delta, i, \ell) \geq 0$. □

COROLLARY 4. ([13]) *For* MIS *on a graph with maximum degree* Δ, ***Greedy*** *achieves an approximation ratio of* $\frac{\Delta+2}{3}$.

This theorem implies only an approximation of 5/3 for sub-cubic graphs. To do significantly better we need a stronger potential, better advice for greedy and a new method of analysis.

3 Subcubic graphs

The exact potential that we use for subcubic graphs is given by the values $\gamma = 5$ and $\sigma = 4$. The table in Figure 2 shows the potential of several basic reductions for some different independent sets. Unfortunately, as one can see in Figure 2, there exists reductions with negative potential. The goal of our additional advice for greedy will be to deal with these cases. The first step is to collect some consecutive basic reductions into one *extended reduction* so that the potential of some basic reductions is balanced by others. For instance, one way to deal with the basic reduction 2.d in Figure 1, which can have potential -2 (see (a) in Figure 2), is to force **Greedy** to prioritize a vertex of degree two with a neighbor with degree three. Therefore, if at some point the reduction 2.d is executed it means that the current graph is a disjoint union of cycles. This allows us to consider that the whole cycle forms an extended reduction — that we will call as *cycle reduction* — and we will see later that its potential is now at least -1. This *advised* greedy algorithm, called **MoreEdges** in [14], improves the approximation ratio from 5/3 to 3/2 in sub-cubic graphs. This result can easily be proved by using our potential function with parameters $\gamma, \sigma = 6, 4$. Such approximation simply follows from the fact that all reductions have now non-negative potential.

An useful observation in order to define an appropriate extended reduction is to notice that the (basic) path reduction (1.b from Figure 1) has potential at least zero. This observation suggests to introduce the following notion. Given a graph $G = (V, E)$ we will say that the set $B = \{w, v_1, \ldots, v_b, w'\} \subset V$ is a *backbone* if the induced subgraph $G[\{v_1, \ldots, v_b\}]$ is a path and if w and w' have both degree three. In this case, w and w' are called the *end-points* of the backbone B. Moreover, when b is odd (*resp.* even), we will say that B is an *even* (*resp. odd*) backbone — notice the asymmetry — which corresponds to the parity of the number of *edges* between the end-points. As an example, the ground of the basic even-backbone reductions (2.f and 2.c in Figure 1) are special case of an even-length-backbone (of edge-length two).

3.1 Extended reductions An *extended reduction* $\overline{R} = (R_1, \ldots, R_s)$ is a sequence of basic reductions R_i of special type that we will precisely describe in the next paragraph. All different extended reductions are summarised in Proposition 5. To facilitate the discussion, when there is no risk of confusion, we will simply call it a *reduction*. The *size* of an extended reduction $\overline{R}$, written $|\overline{R}|$ is the number of executed basic reductions. Its *ground* naturally corresponds to the union of the grounds of its basic reductions, $ground(\overline{R}) := \bigcup_i ground(R_i)$ and its *root* is the same as the root of the first basic reduction. Finally, the *contact vertices* corresponds to all contact vertices of its basic reductions that are not in $ground(\overline{R})$. The

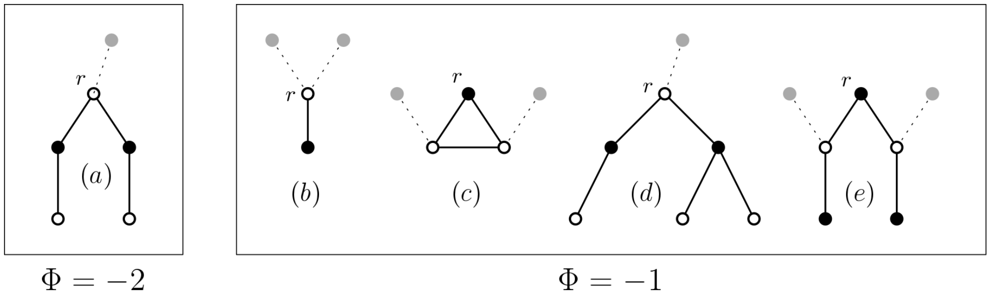

Figure 2: Basic reductions with negative potential. The root vertex of reductions is denoted by the letter r. Dotted edges translate the debt of each white vertex. Grey vertex can either be black or white.

degree of a reduction is the degree of the first executed basic reduction. All basic reductions of Figure 1 except 2.d will be considered as (extended) reductions of size one. In particular, all (extended) reductions of degree one considered by the algorithm have size one. Other considered (extended) reductions of degree two have a ground which is a backbone (except the case of odd-backbone where one end-point is excluded). When the two end-points of the backbone are the same vertex, it corresponds to a *loop reduction*. Otherwise, reductions associated to an even and odd backbone are respectively called *even-backbone* and *odd-backbone* reductions. When these reductions have size at least two, they correspond to a sequence $\overline{R} = (R_1, \ldots, R_s)$ of basic reductions where: the first (basic) reduction R_1 is 2.e from Figure 1, intermediate reductions R_i, with $2 \leqslant i \leqslant s-1$, are basic path reduction (1.b from Figure 1), where the root vertex of R_i is the contact vertex of R_{i-1}, and the final (basic) reduction R_s corresponds to:

- branching (1.c) or path (1.b), when $\overline{R}$ is an *even-backbone* reduction. The case $R_s = \textit{path}$, occurs when the end-points are adjacent.
- path (1.b), when $\overline{R}$ is an *odd-backbone* reduction.
- point (0.a) or edge (1.a), when $\overline{R}$ is a *loop* reduction, depending on the parity of the length of the backbone. Recall that the two end-points are identical in this case.

We give examples of different types of extended reductions of degree two in Figure 3. Some further remarks are in place here:

- The following basic reductions in Figure 1 are special case of (extended) reduction of size one. **2.a:** cycle reduction, **2.c and 2.f:** even-backbone reduction, **2.e:** odd-backbone reduction (this applies only when the right-most contact vertex c has degree three), **2.b :** loop reduction.
- The root of an even-backbone, odd-backbone or loop reduction is always the neighbor of one of the end-points of the backbone. For even-backbone and loop reduction of even-length, any of the two choices leads to the same solution. In the case of an odd-loop reduction, the size of the solution — and therefore also the potential — and the ground of the reduction is exactly the same. For loop and even-backbone reductions, the ground of the reduction is the full backbone. However, for odd-backbone reductions, given one backbone, there are *two distinct* possible roots, associated to two distinct grounds. For each odd-backbone reduction, only one end-point is contained in the ground. See Figure 3.
- All basic reductions of an extended reduction, except the first one have degree at most one, so that at any given moment, any executed basic reduction has minimum degree in the current graph. This means that we are allowed to execute the full extended reduction without violating our original greedy rule.

In what follows, we refer to an extended reduction in two different (and equivalent) ways. We will either write its name with the first capital letter or we will write its name with the first lower-case letter followed by the word "reduction". Thus, for example, we will say a loop reduction or just Loop, or an even-backbone reduction, or just Even-backbone, etc. Note that basic reductions are special case of extended reductions, and therefore they also may follow this convention.

3.2 Ultimate advices for Greedy We now describe the additional rules used to reach the best possible approximation. This advised greedy algorithm will be called **Greedy*** . The first of these rules is to execute basic reductions such that the obtained sequence can be grouped in a sequence of extended reductions as described above. This is justified by Proposition 5.

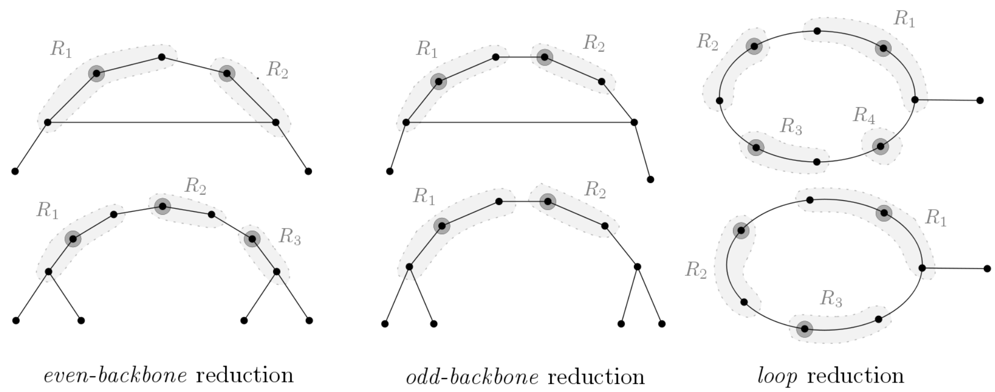

Figure 3: Some examples of (extended) reductions of degree two and size at least two. Light grey areas indicate the ground of each executed basic reduction, that together form the ground of the (extended) reduction. Vertices surrounded by grey rings are the roots of the corresponding basic reductions.

This choice is always possible since all basic reductions from Figure 1, except 2.d, are special cases of extended reductions. In the case where any minimum degree vertex is the root of a basic reduction 2.d, the graph must be a disjoint union of cycles. In this case we are able to execute **Greedy*** so that its execution corresponds to a sequence of cycle reductions. This argument leads to Proposition 5.

Proposition 5. *For each sub-cubic graph with minimum degree at most two, it is always possible to execute one of the following (extended) reductions:*

Point - Edge - Path - Branching - Loop - Cycle - Even-backbone - Odd-backbone.

Greedy* order. When several choices of reductions are possible, **Greedy*** will have to select one with the *highest priority*, according to the following *order* from the highest to the lowest priority:

1. Point, Edge, Path, Branching,
2. Cycle or Loop,
3. Even-backbone,
4. Odd-backbone.

Any two reductions among the first group or any two reductions among the second group can be arbitrarily executed first, as soon as both have the minimum degree. We say that a reduction is a *priority reduction* if there exists no reduction in the same graph with strictly higher priority. Thus a priority reduction is one of the highest priority reductions in the current graph. One implication of this order is that when an Even-backbone is executed, it means that the current graph does not contain any degree one vertices, or any loop reduction. Additionally, when the priority reduction is Odd-backbone, the graph does not contain any Even-backbone. These structural observations will be useful later.

When the priority reduction is an even-backbone or an odd-backbone reduction, **Greedy*** applies the following two additional rules.

Even-backbone rule. Suppose that the priority reduction in the current graph G is the even-backbone reduction, and several choices are possible. Unfortunately, picking arbitrarily one of these reductions can lead to a solution with poor approximation ratio. For instance, consider the graph H_i in Figure 4, highly inspired by [8].

It turns out that the difficulty comes from the fact that executing an even-backbone reduction can split the graph into several connected components, each of them having a negative potential. To address this issue, we want to make sure that we are able to "control" the potential of almost all of these connected components. This right choice, followed by **Greedy*** , is given by the following lemma. For any reduction $\overline{R}$ in a graph G we will say that $\overline{R}$ *creates* connected components $H_1, \ldots, H_s$ if they are the connected components of the graph $G[V \setminus ground(\overline{R})]$. Intuitively, it suffices to execute an even-backbone reduction $\overline{R}$ such that all other even-backbone reductions are all present in the same connected component created by $\overline{R}$.

Lemma 6. *Let G be a connected graph, with no degree one vertices, and no loop reduction. Let $\mathcal{B} = \{\overline{R}_1, \ldots, \overline{R}_p\}$ be the set of all even-backbone reductions in G. Each even-backbone reduction $\overline{R}_i$ has two root vertices r_i and r'_i. In the case when $\overline{R}_i$ has only one root, we set $r_i = r'_i$. Then, there exists one even-backbone*

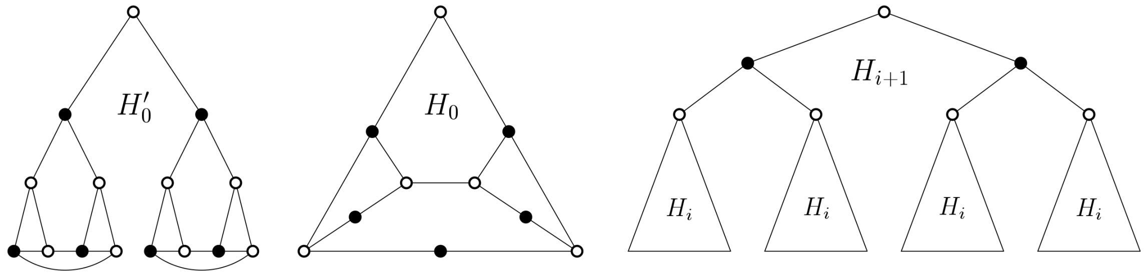

Figure 4: For any $i \geqslant 0$, the highest priority reduction in H_i is Even-backbone, and picking recursively the top vertex leads to a solution where the approximation ratio tends to $17/13 > 5/4$ when i tends to infinity, when H_0 is used as the base gadget. However, if we use H'_0 as the base gadget, because the greedy choice of the reduction is essentially unique at each stage, these instances show that *any* greedy algorithm has an approximation guarantee that tends to $5/4$ when i tends to infinity. This second family is due to Halldórsson and Yoshihara [14].

reduction, say $\overline{R}_1$, that satisfies the following property. Let $H_1, \ldots, H_t$ be the connected components created by $\overline{R}_1$, with $1 \leqslant t \leqslant 4$. Then either $t = 1$, or $t \geq 2$ and then the following is true. If there exist r_i, r_j for some $i, j \geq 2$ and $i \neq j$, such that $r_i \in V(H_1)$ and $r_j \in V(H_2)$ (in words: r_i and r_j belong to two different connected components among $H_1, \ldots, H_t$), then at least one of r_i, r'_i, r_j, r'_j is a contact vertex of $\overline{R}_1$.

Proof. (Lemma 6) Let $a \in V(G)$ be any degree three vertex. Consider a graph $\widetilde{G}$ obtained from G by replacing each backbone from r_i to r'_i by a single degree two vertex which is also called r_i. On this *contracted* graph, let $d_{\widetilde{G}}(u, v)$ denote the shortest path distance (*i.e.* with minimum number of edges) between vertices $u, v \in V(\widetilde{G})$ in $\widetilde{G}$. Now, let us pick the root r_i in $\widetilde{G}$ that has the largest distance $d_{\max} := \max_i d_{\widetilde{G}}(r_i, a)$ from a. Without loss of generality this is r_1 : $d_{\widetilde{G}}(r_1, a) = d_{\max}$. Denote by $H_1, \ldots, H_t$ the connected components created after executing the corresponding even-backbone reduction $\overline{R}_1$. At most one connected component, say H_1, contains a. Suppose that there is another connected component, i.e., H_2, that contains a vertex r_j. Any path from r_j to a intersects *ground*$(\overline{R}_1)$, including the shortest one, and $d_{\widetilde{G}}(r_j, a) = d_{\widetilde{G}}(r_1, a) = d_{\max}$. It follows that r_j and r_1 have one common neighbor b, so that $d_{\widetilde{G}}(r_1, r_j) = 2$. In particular, in the original graph G, r_1 (or r'_1) is at distance two from r_j (or r'_j) which means that this vertex is a contact vertex of $\overline{R}_1$. □

Notice that this proof is constructive and allows us to find the appropriate Even-backbone in time $\mathcal{O}(|V|)$.

Odd-backbone rule. Suppose now that the priority reduction is the odd-backbone reduction. In this case, **Greedy*** chooses the one that was created latest. More formally, suppose that we are given a partial execution $\overline{R}_1, \ldots, \overline{R}_j$ in a graph G such that the priority reduction in G_j is an odd-backbone reduction, where G_i is the subgraph of G obtained from G after the execution of $\overline{R}_1, \ldots, \overline{R}_i$, for $i = 1, \ldots, j$. We associate to each vertex v of degree two a *creation time* $t_v \in \{0, \ldots, j-1\}$, such that t_v is the greatest integer such that v had degree three in $G_{t_v - 1}$. Moreover, if v had already degree two in the original graph G, then set $t_v = 0$. When $t_v \geqslant 1$, this means that v was a contact vertex of t_v-th reduction. Then, the *creation time* of an (odd) backbone B is the greatest creation time over all vertices of degree two in B.

Greedy* picks the odd-backbone reduction that has the greatest creation time, among all possible odd-backbone reductions. If several choices are possible, it can pick any of them.

We believe that this rule is not necessary in the sense that it does not improve the approximation ratio. However, this rule makes the algorithm easier to analyse. Intuitively, with this rule, we can not have several successive reductions with negative potential within the same connected component.

Rule for cubic graphs. When the input graph is cubic, *i.e.* each vertex has degree exactly three, then the first reduction has degree three. However, this is the only degree three reduction executed during the whole execution since there will always be a vertex with degree at most two after the execution of the first reduction. In such a situation, we guess the first degree three vertex to pick, so that the potential of the associated execution is positive. By guessing, we mean choosing *any single fixed* vertex u and then trying all *four* executions, each starting from a vertex in the closed neighborhood of vertex u. We show later that the first step can only increase the total potential of the whole sequence. After this step, all reductions have degree at most two, and therefore in the following sections, we will always consider graphs with at least one vertex of degree at most two.

We present below a formal description of our algorithm **Greedy*** .

```
Algorithm Greedy★
Input: a graph G
Output: an independent set S in G
1. S ← ∅
2. if all vertices have degree three then
3.     Let u be any vertex.
4.     Execute four times while loop (line 6.) starting
       with S = {v} and G = G \ N_G[v], for all v ∈
       N_G[u] and output the maximum size solution.
5. end
6. while G ≠ ∅ do
7.     if the priority reduction (w.r.t. Greedy★ order)
       in G is Even-backbone then
8.         Let R̄ be the Even-backbone given by Lemma
           6 (Even-backbone rule).
9.     end
10.    if the priority reduction in G is Odd-backbone
       then
11.        Let R̄ be the latest created Odd-backbone
           (Odd-backbone rule).
12.    end
13.    else
14.        Let R̄ be any priority reduction.
15.    end
16.    Let v_1, ..., v_s be roots of the basic reductions of R̄.
17.    S ← S ∪ {v_1, ..., v_s}
18.    G ← G \ ground(R̄)
19. end
20. return S.
```

It is clear that the set returned by Algorithm **Greedy*** is an independent set. In the next section, we establish its approximation ratio.

3.3 Analysis of the approximation ratio

THEOREM 7. ***Greedy**** *is a* $5/4$*-approximation algorithm for* MIS *in subcubic graphs.*

Let $\mathcal{E} = (\overline{R_1}, \ldots, \overline{R_\ell})$ is a sequence of extended reductions performed by **Greedy*** on an input graph G. In order to analyse the approximation ratio of **Greedy*** , we use our potential function in sub-cubic graphs ($\Delta = 3$) with parameters $\gamma, \sigma = 5, 4$. Given an independent set I in G, the *potential* of an (extended) reduction $\overline{R}$ is

$$\Phi_I(\overline{R}) = 5 \cdot |\overline{R}| - 4 \cdot |I \cap ground(R)| + loan_I(R) - debt_I(R).$$

We start by looking at the potential of (extended) reductions.

3.3.1 Potential of extended reductions

CLAIM 8. *For any independent set I we have the following potential estimates for the reductions:*

$$\Phi_I(\mathsf{Edge}, \mathsf{Cycle}, \mathsf{Odd\text{-}backbone}) \geqslant -1$$
$$\Phi_I(\mathsf{Path}, \mathsf{Loop}, \mathsf{Even\text{-}backbone}) \geqslant 0$$
$$\Phi_I(\mathsf{Point}, \mathsf{Branching}) \geqslant 1$$

For basic reductions, one can easily check by hand all possible cases. Notice that the worst case potential always arises when I is maximum in the ground of a reduction. Figure 2 presents these worst cases for basic reductions: Edge, Cycle and Odd-backbone. Figure 6 shows the worst potential cases of the remaining basic reductions: Path, Even-backbone, Loop, Point, and Branching. From the worst case potential of basic reductions, we can lower-bound the potential of (extended) reductions.

Proof. (Claim 8) It remains to prove lower-bounds for reductions of arbitrary size. See Figure 5. An odd-backbone reduction is a sequence of basic reductions starting with 2.e (in Figure 1), which has a potential at least -1 (Figure 2 (d)), and a certain number of path reductions, with potential at least zero (Figure 6 (a) and (b)), so that the total potential is at least -1. More generally, the potential of an extended reduction is lower-bounded by the sum of the potentials of the first and the last basic reduction. For Even-backbone with non-adjacent backbone end-points, these first and last basic reductions are 2.e ($\Phi \geqslant -1$) and Branching ($\Phi \geqslant 1$, see Figure 6 (g) and (h)) so that the sum is non-negative.

Consider now a cycle reduction $\overline{R}$ of length $n \geqslant 3$. Let us denote by b and w, respectively, the number of black and white vertices, *i.e.* $b = |I \cap ground(\overline{R})|$ and $b + w = n$. Since **Greedy** is optimal in degree at most two graphs and the size of I is at most $\lfloor n/2 \rfloor$, we have:

$$\Phi_I(\overline{R}) = 5\left\lfloor \frac{n}{2} \right\rfloor - 4b - w = 5\left\lfloor \frac{n}{2} \right\rfloor - 3b - n$$
$$\geqslant 5\left\lfloor \frac{n}{2} \right\rfloor - 3\left\lfloor \frac{n}{2} \right\rfloor - n \geqslant -1. \quad (3.1)$$

For Loop and Even-backbone with adjacent end-points, simply observe that their ground is a cycle with one or two additional edges. Each of these edges is either a debt edge — the loan increases by one — or the corresponding middle white vertex has now degree three — the debt decreases by one. In any case, the potential increases by the number of added edges, so that we proved what we wanted. ∎

Notice that the worst potential of Cycle and Loop depends on the length of the ground and the worst case

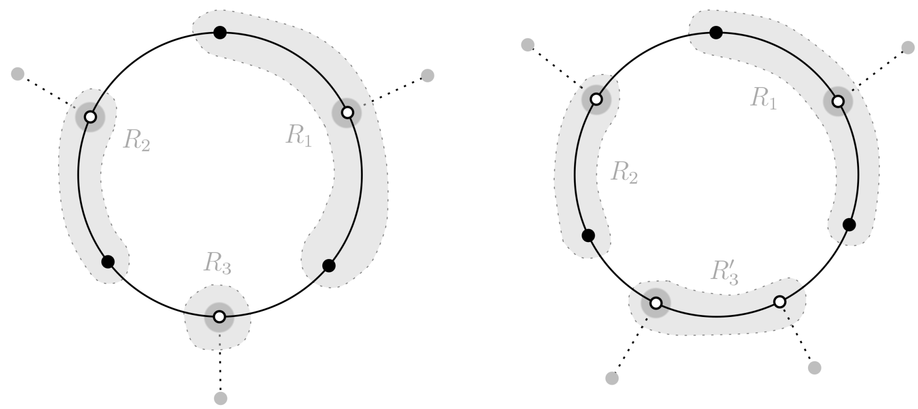

Figure 5: A *cycle* reduction. On the left, a even cycle with potential $\Phi_I(R_1, R_2, R_3) = -2+0+2 = 0$ and on the right a odd cycle with potential -1. Vertices surrounded by a grey disk are the ones picked by the algorithm and dotted edges are debt edges.

corresponds to odd-length cycles. Moreover, notice that when its two end-points are adjacent, the potential of an Even-backbone is at least one.

3.3.2 Proof of the inductive lemma As we have seen before, we are able to avoid reduction with potential -2 (Figure 2 (a)) by grouping this basic reduction with the following ones, so that the resulting (extended) reduction, a cycle reduction, has now only potential -1. Unfortunately, we can not use the same trick to avoid reductions with potential -1. Moreover, such reductions: edge, (odd) cycle and odd-backbone reductions can not be avoided if we want to respect the original greedy constraint which is to pick a vertex with minimum degree.

In order to prove that **Greedy*** delivers a 5/4-approximation, we show that the *exact potential* of any execution is non-negative. Unfortunately, it is not true that the *potential* of any execution is non-negative. For instance, picking the top vertex of H_0 in Figure 4 produces an execution with potential -1. Hopefully, we will prove that -1 is the worst value for the potential of any execution.

A potential problem may arise when an execution creates a lot of connected components where each corresponding execution has negative potential. This could possibly lead to an execution with arbitrarily negative potential. Such connected components might be created by reductions having many contact vertices, such as the even-backbone reduction. Our Even-backbone rule was designed to keep the potential of the created connected components under control, ensuring that at most one (or two) of them have negative potential.

This suggests that to solve our problem we could try to characterize the type of graphs that can have negative potential, *i.e.* , for which there exists an execution with negative potential. Finding such a characterization seems difficult but hopefully, since we know which reduction's potential is -1, we are able to formulate a necessary condition together with a suitable induction hypothesis. In the following, we describe this class of graphs called *potentially problematic* graphs. Additionally, all executions starting with a negative potential reduction, namely the *bad* odd-backbone reduction, can possibly have a negative potential. Notice that 'potentially' refers to the potential function and at the same time to the fact that there exist some executions on some potentially problematic graphs with non-negative potential.

Given a graph G and an independent set I in G, recall that a vertex $v \in V(G) \cap I$ from I is *black* and *white* otherwise. We say that a backbone $B = \{w, v_1, \ldots, v_b, w'\}$ is *alternating for I* (or simply *alternating*) when v_i is black (or white) if and only if i is even. See Figure 7 for an example of an odd alternating backbone. Notice that there is no restriction on the types of the end-points of the backbone.

The next definition of the white and black type reductions is absolutely crucial to our proof. The main intention of this definition is that the black and white reductions should have the greedy priority strictly higher than that of an odd-backbone reduction, and in many cases, also higher than that of an even-backbone reduction. However we must be very careful here because what will matter will be in some sense a "parity" of the reductions. Namely, the first observation is that the potential of an odd-backbone reduction, let us call it $\overline{R}$, can be -1 in the case when all its contact vertices are white (see Figure 2 (d)). Therefore when such a reduction $\overline{R}$ is executed first, then we need to "pay" for it

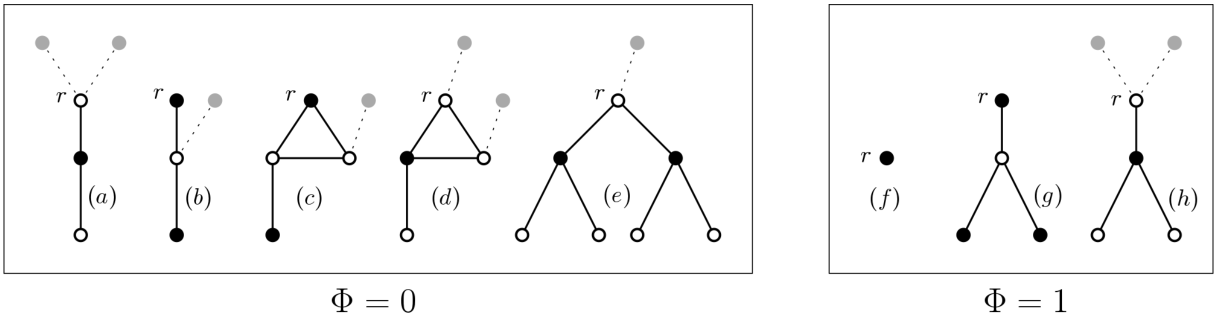

Figure 6: Basic reductions with worst potential equal to 0 or 1. Dotted edges translate the debt of each white vertex. Grey vertex can either be black or white.

by showing that the potential of the following reductions in the connected components it creates will in total be zero (as this is the only way of keeping the total value of the potential to be at least -1). What we now want is that if a potential of a connected component H created by $\overline{R}$ is negative, then because $\overline{R}$ has only white contact vertices in H, even *before* $\overline{R}$ is executed, H must contain a reduction with higher priority than $\overline{R}$, thus leading to a contradiction. Our definition below will ensure this by guaranteeing that such a reduction in H exists with a black root vertex in H by $\overline{R}$'s contact edges.

A kind of a "dual" such high priority reduction is also needed in H. Imagine namely that the first executed reduction $\overline{R}$, that created component H, is also an odd-backbone reduction but its contact vertices are all black. Then they will "block" the black vertices in H. But then our definition below still guarantees an existence of a reduction in H, before executing $\overline{R}$, which has a white root vertex and has priority strictly higher than $\overline{R}$.

The third possibility is when $\overline{R}$ is an odd-backbone reduction with a white and black contact vertex. This is not a problem because the potential of such a reduction is non-negative or even positive. It turns out that we can deal with $\overline{R}$ being an even-backbone reduction in a different way.

There are some further technicalities and details and they can be read in the details of our full proof.

Definition 9. (Black and white type reductions) *Given a graph G and an independent set I in G, we define the black or white type reductions in G by the following rules:*
(1) *Any path reduction or branching reduction in G with black root and white middle vertex (*resp. *with white root vertex and black middle vertex) is a black (*resp. *white) type reduction.*
(2) *Any Loop reduction $\overline{R}$ in G, where $I \cap ground(\overline{R})$ is a maximum independent set in $ground(\overline{R})$, whose both root vertices are white (*resp. *whose at least one root vertex is black) is called a white (*resp. *black) type reduction.*
(3) *Any even-backbone reduction $\overline{R}$ in G, with an alternating backbone, whose both root vertices are white (*resp. *black) is called a white (*resp. *black) type reduction.*

We note here that the black and white reductions correspond to the worst case of the potential (see Figures 2 and 6). We also say that an odd-backbone reduction $\overline{R}$ is *bad for I* (or simply *bad*) if $\Phi_I(\overline{R}) = -1$, see Figure 2 (d). More generally, given an independent set I, we will say that a reduction $\overline{R}$ is *bad* when its potential is minimized by I, *i.e.* $\Phi_I(\overline{R}) = \min\{\Phi_{I'}(\overline{R}), I' \text{ independent set}\}$.

Notice that an odd-backbone reduction does not appear in the definition of the black and white type reductions. Recall that our intention was that any black/white reduction has a strictly higher priority than Odd-backbone.

Definition 10. *Let G be a connected graph with minimum degree at most 2 and I an independent set in G. We say that G is* potentially problematic *for I (or just* potentially problematic*), if either:*
(1) *G is an odd cycle or an edge and I is maximum in G.*
(2) *or, there exists one reduction of black type* and *one reduction of white type in G.*

Notice that Cycle (and also Edge) reduction has potential -1 if and only if its ground has odd-length and I is maximum (Claim 14). In this situation, we will call these graphs *bad* cycle and *bad* edge.

The following Lemma states that any execution has a potential always at least -1. We prove this result by induction together with a necessary condition characterizing executions with minimum potential.

Lemma 11. (Inductive low-debt Lemma) *Let G be a connected graph with minimum degree at most two, I an independent set in G and $\mathcal{E} = (\overline{R}_1, \ldots, \overline{R}_\kappa)$ an execution on G. Then*

(1) $\Phi_I(\mathcal{E}) \geqslant -1$

(2) If $\Phi_I(\mathcal{E}) = -1$*, then either*

  (a) The first reduction $\overline{R}_1$ is a bad odd-backbone reduction.

  (b) or, G is potentially problematic.

Along the proof of Lemma 11 we will refer to several technical claims proved in Section 3.4 about extended reductions.

Proof. (of Lemma 11) We prove this result by induction on the number κ of extended reductions in the execution $\mathcal{E}$. First, if $\kappa = 1$, then since G is connected, the reduction is a terminal reduction, *i.e.* point, edge or cycle reduction, and by Claim 8 we know that their potential is at least -1. Moreover, if the potential is exactly -1, it is not difficult to see from the proof of Claim 8 that the reduction is either an edge or odd cycle reduction, which are potentially problematic graphs. For a detailed proof of this fact see Claim 14.

Suppose now that $\mathcal{E}$ consists of $\kappa \geqslant 2$ extended reductions. We will treat all cases depending on the first extended reduction $\overline{R}_1$. We denote its root and the contact vertices by letters r and c_i, with $1 \leqslant i \leqslant 4$. In all these cases we will apply the induction hypothesis to each connected component of the graph after executing the first reduction. Recall that we say that $\overline{R}_1$ *creates* connected components $H_1, \ldots, H_s$ if they are connected components of the graph $G[V \setminus ground(\overline{R}_1)]$. Here, s with $1 \leqslant s \leqslant 4$, denotes the number of connected components created by $\overline{R}_1$. Reductions executed by **Greedy*** in distinct connected components are independent. Therefore, without loss of generality we can assume that each execution on H_i corresponds to a sub-execution $\mathcal{E}_i$ of $\mathcal{E}$ so that $\mathcal{E} = (\overline{R}_1, \mathcal{E}_1, \ldots, \mathcal{E}_s)$. Notice that according to Proposition 1, we have

$$\Phi_I(\mathcal{E}) = \Phi_I(\overline{R}_1) + \Phi_I(\mathcal{E}_1) + \cdots + \Phi_I(\mathcal{E}_s).$$

We first prove hypothesis (1) and (2) if the first reduction is Path, Branching or Loop.

When $\overline{R}_1$ is Path, Branching or Loop. These are easy cases since the number of connected components created (and therefore the potential of each corresponding execution) is always balanced by the potential of the reduction. This is precisely written in the following fact that can be easily verified thanks to Figure 1 and Claim 8.

Observation 12. *For any independent set I, and any reduction $\overline{R}$ that is Path, Branching or Loop:*

$$\Phi_I(\overline{R}) \geqslant |contact(\overline{R})| - 1.$$

Remark: The inequality is also valid for an even-backbone reduction whose both end-points are adjacent.

Then, if $\overline{R}_1$ is a path, branching, or loop reduction then the number of connected components is at most the number of contact-edges, therefore by the induction hypothesis (1) on each connected component of $G[V \setminus ground(\overline{R}_1)]$, we have

$$\begin{aligned}\Phi_I(\mathcal{E}) &= \Phi_I(\overline{R}_1) + \Phi_I(\mathcal{E}_1) + \cdots + \Phi_I(\mathcal{E}_s) \\ &\geqslant \Phi_I(\overline{R}_1) + (-1)s \\ &\geqslant \Phi_I(\overline{R}_1) + (-1)|contact(\overline{R}_1)| \geqslant -1.\end{aligned}$$

Moreover, if $\Phi_I(\mathcal{E}) = -1$, then all these inequalities are tight and in particular, the potential of $\overline{R}_1$ is minimum. This implies that $\overline{R}_1$ must be a reduction of black or white type by Claim 15, and additionally, it must create exactly $|contact(R)|$ connected components, and each one must have potential -1. Applying the inductive assumption (2) to these connected components, together with the following Claim 13 implies property (2b) for G.

Claim 13. *Let G be a connected graph, and I an independent set in G. Consider $\mathcal{E} = (\overline{R}_1, \ldots, \overline{R}_\kappa)$ an execution in G. Let H be a connected component created by the first reduction $\overline{R}_1$, such that all contact vertices of $\overline{R}_1$ that are in H are all white (*resp. *all black). Then, if H is potentially problematic* or *if the first reduction executed in H is a bad odd-backbone reduction, then there exists a black (*resp. *white) reduction $\overline{R}$ in G such that $ground(\overline{R}) \subseteq V(H)$.*

We will prove that in this situation, $\overline{R}_1$ can not be an Odd-backbone, since this would not be the priority reduction according to **Greedy*** order. This claim is useful in the sense that, when the potential of $\overline{R}_1$ is minimum then $\overline{R}_1$ is white (or black) and its contact vertices are all white (or all black) (Claims 14, 16), so that G is a potentially problematic graph.

Proof. (Claim 13) Suppose that H is created by $\overline{R}_1$ with contact vertices $c_1, \ldots, c_t \in V(H)$ all white (*resp.* all black), with $1 \leqslant t \leqslant 4$. We show that there exists one black (*resp.* white) reduction $\overline{R}$ in G such that $ground(\overline{R}) \subseteq V(H)$.

First, assume that the first reduction executed in H, say $\overline{R}_2$, is a bad odd-backbone reduction. Denote by $B = \{w, v_1, \ldots, v_{2b}, w'\}$ its backbone, with end-points w and w'. Since $\Phi_I(\overline{R}_2) = -1$, its backbone is alternating

(Claim 17) and without loss of generality we assume that v_j is white if and only if j is odd.

According to the Odd-backbone-rule, at least one of the contact vertices c_i must be one vertex of its backbone. Since all c_i are white (*resp.* black), the distance along B between any pair of $\{c_1, \ldots, c_t, w'\} \cap B$ (*resp.* $\{w, c_1, \ldots, c_t\} \cap B$) is even. Therefore, there is an alternating even-length-backbone between any two consecutive ones. This implies the existence of a black (*resp.* white) even-backbone reduction in G, see Figure 7.

Now, assume that H is a potentially problematic graph, and suppose first that there is one black (*resp.* white) reduction $\overline{R}_2$ with a black (*resp.* white) root vertex r in H. Notice that this vertex is distinct than all c_i, and then has the same degree in G. Then, if $d_H(r) = d_G(r) = 1$, then r is also the root of a black (*resp.* white) reduction in G. Then, suppose that $d_H(r) = d_G(r) = 2$.

If r is the root of a black (*resp.* white) even-backbone reduction $\overline{R}_2$ in H, with backbone end-points w and w', then any two consecutive vertices of the set $\{w, w', c_1, \ldots, c_t\}$ along this backbone form a alternating even-length-backbone. In particular, r is the root of a black (*resp.* white) Even-backbone in G.

The case when $\overline{R}_2$ is a Loop is very similar, but slightly more technical. First, if there is no c_i in the ground of $\overline{R}_2$, the root of this reduction is obviously also the root of an black (*resp.* white) Loop in G. Now, suppose that there is at least one contact vertex in $ground(\overline{R}_2)$. Let w the vertex of degree three in $ground(\overline{R}_2)$, and r, r' its two neighbors. Let us focus on the first two contact vertices c and c' met when we sweep the loop from w in each direction (or just c its the only contact vertex present). We claim that at least one of r of r' a the black (*resp.* white) root of an (alternating) black (*resp.* white) Even-backbone in G. See Figure 8 (a) and (b) (*resp.* (c)). Form left to right this vertex is respectively r', r and r.

We now turn our attention to the case when H is a bad[2] edge or cycle. If H is a bad edge, then its black (*resp.* white) vertex has degree one in G, and therefore is the root of black (*resp.* white) path or branching reduction in G. Similarly as the case when $\overline{R}_2$ is a Loop, if H is a bad cycle, then there exists a black (*resp.* white) vertex $r' \in V(H)$ that is the root of a black (*resp.* white) Loop in G, when $\overline{R}_1$ has one contact vertex in H, or the root of a black (*resp.* white) Even-backbone in G, when $\overline{R}_1$ has more that one contact vertex in H. □

We now turn our attention to the case when $\overline{R}_1$ is a backbone reduction.

$\overline{R}_1$ is an even-backbone reduction. If the first reduction executed in G is an Even-backbone, it means according to Greedy order, that the graph G does not contain any degree one vertices or any loop reductions. All degree two vertices are contained in some backbones linking two distinct degree three vertices. If the end-points of the backbone of $\overline{R}_1$ are adjacent, then $\overline{R}_1$ satisfies Observation 12, so that this case was treated in the previous section. From now, let us assume that these end-points are independent. In the following, we use the same terminology as in Lemma 6.

• If all contact vertices of $\overline{R}_1$ are white (*resp.* black), then at most one connected component created by $\overline{R}_1$ has potential -1. Indeed, suppose for a contradiction, that H_i, with $i \geqslant 2$, has potential -1. By induction hypothesis, it must satisfy assumptions (2a) or (2b) of Lemma 11. According to Claim 13, there was a black (*resp.* white) reduction in H_i before $\overline{R}_1$ was executed. This reduction is neither a degree one nor a loop reduction, so it must be an even-backbone reduction. Since they are black (*resp.* white), the root vertices of this reduction are distinct than $\overline{R}_1$'s contact vertices, which contradicts Lemma 6. We proved (1) when all contact vertices of $\overline{R}_1$ are all white (*resp.* black). For (2), if $\Phi_I(\mathcal{E}) = -1$, then one connected component created, say H_1 has potential -1, which by induction satisfies (2a) or (2b) and $\Phi_I(\overline{R}_1) = 0$, so that $\overline{R}_1$ is a white reduction with only white contact vertices (Claim 16). Claim 13 guarantees the existence of a black reduction in G, so that G is potentially problematic.

• If some of $\overline{R}_1$ contact vertices are are both black and white, then we argue that $\Phi_I(\mathcal{E}) \geqslant 0$. First, the potential of $\overline{R}_1$ is at least two[3] (Claim 16). Therefore we should argue that there are *at most two* connected components with potential -1. This is true since there are at most two connected components with strictly more than one contact vertex, and at most one connected component with exactly one contact vertex can have potential -1. Indeed, for connected components with only one contact vertex, Claim 13 applies so that we can use the same argumentation than in the previous paragraph.

$\overline{R}_1$ is an odd-backbone reduction. Assume first that $\overline{R}_1$ has potential $\Phi_I(\overline{R}_1) = -1$ (*resp.* $\Phi_I(\overline{R}_1) = 0$). Then, Claim 17 indicates that all its contact vertices are white (*resp.* black).

(1) We prove that all connected components created by $\overline{R}_1$ have potential at least zero. Assume that it is not true for the component H_1. By induction it

[2] Meaning here that I is maximum in H.

[3] We assume here that the two end-points of the corresponding backbone are not adjacent.

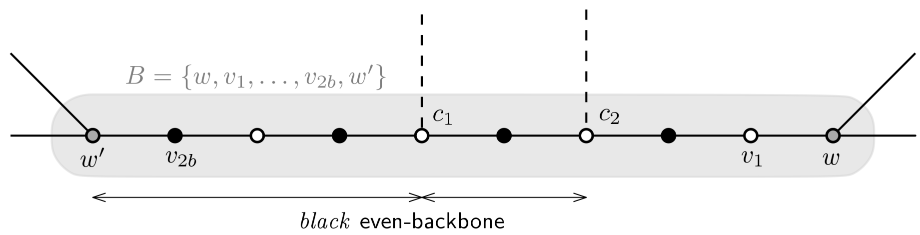

Figure 7: Existence of a black reduction in proof of Claim 13. B is an alternating odd backbone in H. Grey end-points illustrate the fact that these vertices can either be black or white. Dashed edges are contact edges from $\overline{R}_1$. Before the execution, there exists a black (alternating) Even-backbone in G between w', c_1 and c_1, c_2.

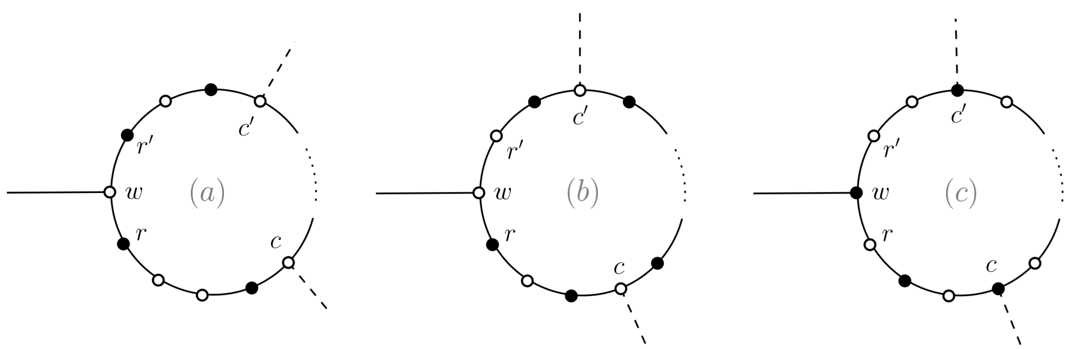

Figure 8: Different types of black, (a) and (b), and white Loop (c). Dashed edges are contact edges.

satisfies (2a) or (2b). According to Claim 13, there exists a black (*resp.* white) reduction in H_1 before $\overline{R}_1$ is executed which contradicts Greedy order[4] , since any black or white reduction has a priority strictly higher than Odd-backbone.

(2) When $\overline{R}_1$ is supposed to be bad odd-backbone by assumption, (2a) is always true, and otherwise, if $\Phi_I(\overline{R}_1) = 0$, we just proved that $\Phi_I(\mathcal{E}) \geqslant 0$.

Suppose now that $\Phi_I(\overline{R}_1) \geqslant 1$. We claim that at most one component created by $\overline{R}_1$ can have potential -1. First note that $\overline{R}_1$ has three contact edges and thus at most three contact vertices. Indeed, at most one connected component created has at least two contact vertices, and any connected component H created with exactly one contact vertex can not have potential -1 since Claim 13 would imply the existence of an highest priority reduction in H.

This concludes the proof of Lemma 11. ∎

Proof. (of Theorem 7) We first treat the case where the input graph has at least one vertex with degree at most 2. Let G be a connected graph, I a *maximum* independent set in G, and $\mathcal{E}$ an execution. Our goal is to show that the *exact* potential is non-negative :

$$\Phi_{G,I}(\mathcal{E}) \geqslant 0 \tag{3.2}$$

Suppose this is true. Then from Proposition 1 we have that $5|\mathcal{E}| - 4|I| \geqslant 0$, which can be re-written as $\frac{|I|}{|\mathcal{E}|} \leqslant \frac{5}{4}$, and since this is true for any independent set, then we have established the desired approximation.

In Lemma 11, we proved that $\Phi_I(\mathcal{E}) \geqslant -1$. Suppose that the inequality is tight. Then, the first reduction is a bad odd-backbone reduction or G is potentially problematic. In any case there exists in G a white vertex with degree at most two — that is the root of the first Odd-backbone or of the white reduction in G — so that, using Claim 2 we have:

$$\Phi_{G,I}(\mathcal{E}) = \Phi_I(\mathcal{E}) + \sum_{v \notin I} (3 - d_G(v)) \geqslant -1 + 1 = 0$$

Suppose now that all vertices in G have degree three, and assume that I is maximal, so that for any vertex u, $|N_G[u] \cap I| \geqslant 1$. Then, let us consider any degree 3 vertex u and we see that one of the four executions of the algorithm will be execution with $v \in N_G[u]$ being black, and let us call this first reduction R^*. We detect which of those four executions to take by taking the one

[4] This is where the Odd-backbone-rule is used : in any execution we can not have two consecutive odd-backbone reductions with minimum potential.

that leads to the largest size solution. By Claim 18 it implies that this execution has the largest potential.

On the other hand let us consider the execution $\mathcal{E}$ of R^* with its black root v. Let $H_1, \ldots, H_s$ denote the connected components created by R^*, and $\mathcal{E}_1, \ldots, \mathcal{E}_s$ the corresponding executions. Without loss of generality we have $\mathcal{E} = (R^*, \mathcal{E}_1, \ldots, \mathcal{E}_s)$. To prove that the exact potential of $\mathcal{E}$ in G is positive, the trick is to consider that the first reduction R^* does not use any loan from its loan edges, so that its potential is exactly 1. This implies also that each connected component will not have any debt edges. Then as we proved before, since each connected component H_i has a vertex with degree at most two, its exact potential in H_i is non-negative. Therefore, we have $\Phi_{G,I}(\mathcal{E}) = 1 + \sum_{i=1}^{s} \Phi_{H_i,I}(\mathcal{E}_i) \geqslant 1$. ∎

Remark: Notice that our analysis implies that the size of the final solution is in fact at least $(4/5)OPT + 1/5$ if the first executed reduction is of degree 3. Moreover, if the first reduction $\overline{R}$ is a bad even-backbone reduction, and G is not a problematic graph, then our analysis proves that the **Greedy*** solution's size is a at least $(4/5)OPT + 1/5$. This precisely matches the size of the lower bound example of Halldórsson and Yoshihara from Figure 4. Our analysis even indicates that this lower bound example has the worst possible approximation ratio for any ultimate greedy algorithm. Indeed, this counter example is actually a sequence of graph $(G_n)_n$, and the solution returned by any Greedy has size $(4/5)OPT(G_n) + 1/5$, and therefore the corresponding approximation ratio tends to 5/4 when the size of G_n tends to infinity. However, one may wonder if there exists a (finite) graph G, such that any greedy produces a solution of size at most $(4/5)OPT(G)$. Our previous analysis indicates that such a graph can not exist.

Indeed, this graph must satisfy (2a) of (2b) from Lemma 11, otherwise our **Greedy*** outputs a solution of size at least $(4/5)OPT(G) + 1/5$. Then, for any maximum independent set, this graph must have at least one *black* minimum degree vertex, and in this situation, we could for instance try all possible minimum degree vertices (only for the first step), and pick the execution of maximum size. This modified greedy algorithm returns a solution of size at least $(4/5)OPT(G)+1/5$, for any input graph G.

Remark: Through basically the same technique, we can achieve a greedy algorithm with an approximation ratio of $\frac{4}{3}$ but with linear running time. Observe that if we set $\gamma = 4$ and $\sigma = 3$, the minimum potential of an odd-backbone reduction changes from -1 to 0. Now, only Cycle has potential value of -1. With this observation, we are able to extend Loop in a particular way, which implies, finally, that we are able to exclude even-backbone reductions from the definition of black and white type reductions, and also preserve the induction hypothesis in our inductive proof. It implies that the Even-backbone rule is not necessarily needed, and then the running time of such greedy algorithm will be linear.

3.4 Technical claims

CLAIM 14. *Given a connected graph G, where any execution consists of* one *extended reduction $\overline{R}$,* i.e. *Point, Edge or Cycle, then and if $\Phi_I(\overline{R}) = -1$, for an given independent set I in G, then G is either a bad odd-length cycle, or a bad Edge, and I is* maximum *in G.*

Proof. First, by Claim 8, if $\overline{R}$ is a Point then $\Phi_I(G) \geq 1$. Then, if it is an edge reduction, since it is a basic reduction, it is easy to check that $\Phi_I(\overline{R}) = -1$ only when one vertex is black *i.e.* I is maximum, see Figure 2 (*b*).

Finally, when G is a cycle then, the corresponding cycle reduction has potential -1 when all inequalities from equation (3.1) in the proof of Claim 8 are tight. In particular the number b of black vertices must be maximum, and $\left\lfloor \frac{n}{2} \right\rfloor - \left\lceil \frac{n}{2} \right\rceil = 1$, that only arises when the length n of the cycle is odd. ∎

CLAIM 15. *Any Path, Branching, Loop with minimum potential have type black or white, and have contact vertices all black, or all white.*

Proof. For (basic) path and branching reductions, worst case potential (Figure 6 (a),(h) — *resp.* (b),(g)) are white (*resp.* black) reductions, with white (*resp.* black) contact vertices.

Next, as noticed before, the potential of a Loop, is correlated to the potential of the cycle reduction obtained by removing its contact edge, because adding an edge to a ground of reduction either increases the loan by one or decreases the debt by one, so that the potential increases by one or by two. In particular, when Loop has minimum potential, the corresponding cycle has also minimum potential. By Claim 14, we know that the independent set must be maximum, and its backbone must have odd-length, so that the loop reduction must have type black or white. Moreover when the potential is minimum, the contact edge can not be incident to two white vertices, otherwise making the contact vertex black would decrease the potential. Therefore, when $\Phi_I(\overline{R}) = -1$, the contact vertex has the same type than the reduction. ∎

CLAIM 16. *Let $\overline{R}$ be Even-backbone, then for any independent set I we have $\Phi_I(\overline{R}) \geq 0$, and moreover,*

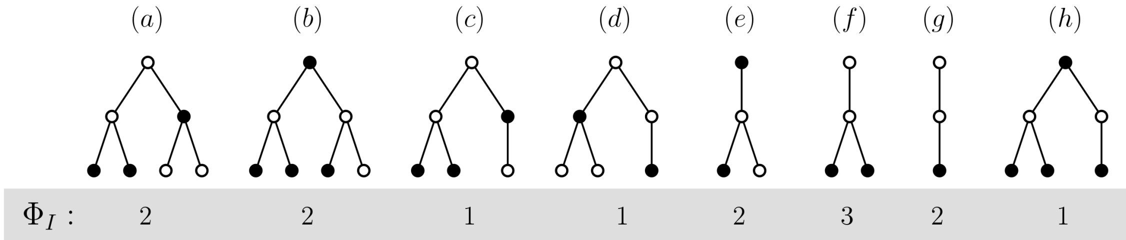

Figure 9: Some examples of good case potential value.

(1) if $\Phi_I(\overline{R}) = 0$, then $\overline{R}$ is a white reduction with only white contact vertices.

(2) if the end-points of $\overline{R}$'s backbone are not adjacent, and if its contact vertices are not all white or all black, then $\Phi_I(\overline{R}) \geq 2$.

Proof. **(1)** Suppose that the potential of an Even-backbone $\overline{R} = (R_1, \ldots, R_t)$ is minimum : $\Phi_I(\overline{R}) = 0$. We know that its end-points are not adjacent, otherwise it has potential at least one, and the potential of all basic reductions R_i must be minimum. When $\overline{R}$ has size one, the worst case arises only when I is such as in Figure 6 (e), that is a particular case of white reduction with white contact vertices. For greater sizes, this implies that the first basic reduction R_1 is like Figure 2 (d). Then R_2 must be a Path with minimum potential and a white root, *i.e.* Figure 6 (a), so on and so long for all path reductions. The final branching reduction R_t has minimum potential and a white root, as in Figure 6 (h). Finally we proved that this backbone is alternating and the root is white, so that $\overline{R}$ is a white reduction. Moreover, all its end-points must be white.

(2) In the following we will say that a reduction is *mixed* if it has two different type contact vertices. When $\overline{R}$ has size one and is mixed, we can easily check by hand, that its potential is at least two (Figure 9 (a), (b)). Consider now a mixed even-backbone $\overline{R} = (R_1, \ldots, R_t)$, and without loss of generality[5], assume that the last branching reduction R_t has at least one black contact vertex.

First, if R_1 or R_t are mixed, then their potential is respectively at least 1 and 2 (see Figure 9 (c),(d) and (e)) so that $\Phi_I(\overline{R}) \geqslant 2$.

Otherwise, assume that R_1 and R_t have only respectively white and black contact vertices. In particular the root r of the first Path R_2 is white. If the root r' of the last branching reduction is white then[6], its potential is at least 3 (Figure 9 (f)), so that $\Phi_I(\overline{R}) \geqslant 2$. Then, if r' is black, since the distance between r and r' is even, we must have found a Path R_i with both root and middle white vertices. Such a reduction has potential at least 2 (Figure 9 (g)) so that $\Phi_I(\overline{R}) \geqslant 2$. ∎

Claim 17. *Let $\overline{R}$ be odd-backbone, then for any independent set I we have $\Phi_I(\overline{R}) \geq -1$, and moreover,*

(1) if $\Phi_I(\overline{R}) = -1$, then it has an alternating backbone, a white root and only white contact vertices.

(2) if $\Phi_I(\overline{R}) = 0$, then it has an alternating backbone, a black root and only black contact vertices.

Proof. These assumptions can be easily checked by hand for odd-backbone reduction $\overline{R}$ of size one, see Figure 2 (d) and Figure 9 (h). Suppose now that $\overline{R} = (R_1, \ldots, R_t)$ has size at least two, where R_1 is the basic odd-backbone reduction, and R_i are path reductions, for $i \geqslant 2$.

If $\Phi_I(\overline{R}) = -1$, then necessarily, $\Phi_I(R_1) = -1$ and for all Path, $\Phi_I(R_i) = 0$. The first reduction has only white contact vertices (Figure 2 (d)), and then R_2 has a white root, so it must be as in Figure 6 (a), in particular it must have a white contact vertex, so that R_3 has a white root, and so on and so forth. This implies that the corresponding backbone is alternating, and the root vertex and all contact vertices are white.

If $\Phi_I(\overline{R}) = 0$, then we either have **Case 1:** all basic reductions have potential zero, or we have **Case 2:** $\Phi_I(R_1) = -1$ and there is one Path R_j with potential one.

Case 1: R_1 must be like in Figure 9 (h), and using exactly the same argumentation than in the previous paragraph, we show that all the following path reductions are like in Figure 6 (b), so that the $\overline{R}$'s backbone is alternating, and $\overline{R}$ has black root and contact vertices.

Case 2: We show that this case is impossible. Indeed, as before all path reductions R_i, with $i < j$ have a white contact vertex (Figure 6 (a)), so that R_j has a white root. Since it has potential one, its middle vertex must be white (otherwise its potential is zero, see Figure 6 (a)), and in this case its potential is at least two (Figure 9 (g)). ∎

[5] Recall that we are free to choose the root of Even-backbone.
[6] We assume here that at least one contact vertex is black.

Claim 18. *Let G be a connected graph and I an independent set in G. Let $\mathcal{E} = (\overline{R}_1, \overline{R}_2, \ldots, \overline{R}_\kappa)$ be an execution of greedy on G, and let this execution produce a solution of size s_r. Let $\mathcal{E}' = (\overline{R}'_1, \overline{R}'_2, \ldots, \overline{R}'_{\kappa'})$ correspond to another execution of greedy on G producing a solution of size $s_{r'}$. Then if $s_r > s_{r'}$ then $\Phi_I(\mathcal{E}) > \Phi_I(\mathcal{E}')$, and otherwise, if $s_r < s_{r'}$ then $\Phi_I(\mathcal{E}') > \Phi_I(\mathcal{E})$.*

Proof. We know that $\Phi_I(\mathcal{E}) = 5s_r - 4|I \cap V(G)|$, and $\Phi_I(\mathcal{E}') = 5s_{r'} - 4|I \cap V(G)|$. Therefore, $\Phi_I(\mathcal{E}) < \Phi_I(\mathcal{E}')$ if and only if $s_r < s_{r'}$. ∎

4 Lower bounds

Due to lack of space, we present in this section only formal statements of our lower bound results.

Given a graph $G = (V, E)$, we say that I is a *greedy set* of G, if I is an independent set, and its elements can be ordered, $I = \{v_1, \ldots, v_k\}$ in such a way that, for all $1 \leqslant i \leqslant k$, the vertex v_i has minimum degree in the subgraph G_i, where $G_i := G[V \setminus N_G[\{v_1, \ldots, v_{i-1}\}]]$. The size of a maximum greedy set in G is denoted by $\alpha^+(G)$.

4.1 Ultimate lower bounds for high degree graphs

Theorem 19. *The ratio of any Greedy like algorithm in graphs with degree at most Δ is at least $\dfrac{\Delta+1}{3} - \mathcal{O}(1/\Delta)$.*

This result is an extension of Theorem 6 in [13]. In this result Halldórsson and Radhakrishnan present examples where the ratio between the worst execution of the basic Greedy and the optimal independent set is $\dfrac{\Delta+2}{3} - \mathcal{O}(\Delta^2/n)$. However, on these examples there exists several vertices with minimum degree and picking the right minimum degree vertex could lead greedy to an optimal solution. Equivalently, it means that there exists graphs of bounded-degree where the *minimum* greedy set is small compared to the maximum independent set. We prove that we can extend this observation to the *maximum* greedy set while keeping roughly the same ratio. Our extension of these examples consists in increasing the degree of some vertices of this graphs by one, so that any greedy set has the same size and the corresponding ratio is $\dfrac{\Delta+1}{3} - \mathcal{O}(1/\Delta)$.

4.2 Cubic planar graphs The maximization problem MaxGreedy consists in finding a maximum size greedy set in a given graph G. This problem was shown to be NP-hard [5]. We first prove that this problem remains NP-hard in the very restricted class of planar cubic graphs. The proof is a reduction from MIS in cubic planar graphs, which is known to be NP-hard.

Theorem 20. MaxGreedy *is NP-complete for planar cubic graphs.*

4.3 Hardness of approximation We are able to prove the following results, showing that not only exact but also approximate version of the MaxGreedy problem is computationally hard.

Theorem 21.

- *For general graphs with n vertices,* MaxGreedy *is hard to approximate within a factor of $n^{1-\varepsilon}$, for any constant $\varepsilon > 0$, assuming $P \neq NP$.*
- *For general graphs,* MaxGreedy *is hard to approximate within a factor of $n/\log n$, assuming the Exponential Time Hypothesis.*
- *For graphs with maximum degree $\Delta \geqslant 7$,* MaxGreedy *is hard to approximate within a factor of $(\Delta+1)/3 - \mathcal{O}(1/\Delta) - \mathcal{O}(1/n)$, assuming $P \neq NP$.*
- *For bipartite graphs,* MaxGreedy *is hard to approximate within a factor of $n^{1/2-\varepsilon}$, for any constant $\varepsilon > 0$, assuming $P \neq NP$.*

The last result is especially interesting, because an optimal independent set can be computed in polynomial time in a bipartite graph. Therefore, **Greedy** is not a good algorithm for this class of graphs. However, this negative result suggests that even knowing a maximum independent set may not be helpful in order to design good greedy advises.

5 Conclusions and future steps

Our main technical contribution is a non-local payment scheme together with an inductive argument that can be embedded with greedy-style algorithms for MIS on bounded degree graphs. These techniques imply best possible approximation guarantees of greedy on subcubic graphs. We have also shown versatility of these techniques by proving (via simple proofs) that they imply close to best possible greedy guarantees on graphs with maximum degree Δ, for any Δ. Furthermore, they also imply fast improved approximation algorithms for the minimum vertex cover problem on bounded degree graphs. We have complemented these results by hardness results, showing that it is hard to compute good advice for the greedy MIS algorithms.

Our techniques have a potential to give further insights into the design of fast non-greedy algorithms that go past the greedy “barrier” in terms of approximation factors. Namely, a non-greedy algorithm can in certain situations choose degree-3 black vertex adjacent to the top white degree-2 vertex in the 5/4-lower bound instances H_i in Figure 4 with $H_0 = H'_0$. Observe that after such a non-greedy reduction, the algorithm can then follow greedy rules to finally compute an optimal solution on these instances. We have found a way of implementing such "super-advice" in $O(n^2)$ time. Moreover, our potential function and the inductive argument can be adapted to analyze the approximation guarantee of this method. This breaks through the 5/4 lower bound of greedy and could even lead to approximations close to 6/5. Recall, that 6/5 is essentially the best currently known polynomial time approximation for subcubic MIS, achieved by local search algorithms which however have exorbitant running times. "Super-advice" could also be used to design non-greedy algorithms that go beyond the proved lower bound of $(\Delta + 1)/3$ for greedy on Δ-bounded degree graphs.

Moreover, our techniques have a potential for further generalizations and applications. Our potential function and the inductive argument are quite general and they could be applied to other related problems on bounded degree graphs. Such general problem should have the following features: given a graph, the optimal solution should be ubiquitously "distributed" over the input graph, and therefore also a feasible solution should be computable sequentially/locally by "choosing parts of the graph", debt and loan should be definable on such a problem as problem specific, depending on the problem's constraints. Possible candidates are, for instance, the set packing and set covering problems with sets of bounded size and bounded element occurrences.

Acknowledgment. We would like to thank Magnús Halldórsson for his help with verifying our example and for his explanations to the results about greedy algorithms.

References

[1] Paola Alimonti and Viggo Kann. Some APX-completeness results for cubic graphs. *Theor. Comput. Sci.*, 237(1-2):123–134, 2000.

[2] Claude Berge. *Graphs and hypergraphs.* North-Holland, Ansterdam, 1973.

[3] Piotr Berman and Toshihiro Fujito. On approximation properties of the independent set problem for low degree graphs. *Theory of Computing Systems*, 32(2):115–132, Apr 1999.

[4] Piotr Berman and Martin Fürer. Approximating maximum independent set in bounded degree graphs. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 365–371, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

[5] Hans L. Bodlaender, Dimitrios M. Thilikos, and Koichi Yamazaki. It is hard to know when greedy is good for finding independent sets. *Information Processing Letters*, 61(2):101–106, 1997.

[6] Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3):27:1–27:32, 2016.

[7] Miroslav Chlebík and Janka Chlebíková. On approximability of the independent set problem for low degree graphs. In Ratislav Královic and Ondrej Sýkora, editors, *Structural Information and Communication Complexity*, pages 47–56, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[8] Miroslav Chlebík and Janka Chlebíková. On approximability of the independent set problem for low degree graphs. In Ratislav Královic and Ondrej Sýkora, editors, *Structural Information and Communication Complexity*, pages 47–56, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[9] András Hajnal and Endre Szemerédi. Proof of a conjecture of P. Erdős. In P. Erdős, A. Rényi, and V.T. Sós, editors, *Combinatorial Theory and its Applications*, pages 601–623. North-Holland, Amsterdam, 1970.

[10] Magnús M. Halldórsson. *Private communication. Erratum:* `https://www.ru.is/~mmh/papers/hy95-errata.html`. March, 2019.

[11] Magnús M. Halldórsson and Jaikumar Radhakrishnan. Greed is good: approximating independent sets in sparse and bounded-degree graphs. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, STOC 1994, Montréal, Québec, Canada, 23-25 May 1994*, pages 439–448, 1994.

[12] Magnús M. Halldórsson and Jaikumar Radhakrishnan. Improved approximations of independent sets in bounded-degree graphs. In *Algorithm Theory - SWAT '94, 4th Scandinavian Workshop on Algorithm Theory, Aarhus, Denmark, July 6-8, 1994, Proceedings*, pages 195–206, 1994.

[13] Magnús M. Halldórsson and Jaikumar Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18(1):145–163, May 1997.

[14] Magnús M. Halldórsson and Kiyohito Yoshihara. Greedy approximations of independent sets in low degree graphs. In John Staples, Peter Eades, Naoki Katoh, and Alistair Moffat, editors, *Algorithms and Computations*, pages 152–161, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. Full version in: `http://www.ru.is/~mmh/raunvis/Papers/yoshi.pdf`.