Vietnam Journal of Mathematics **36**:3(2008) 327–336

Vietnam Journal
of
MATHEMATICS

Performance Analysis of Greedy Algorithms for Max-IS and Min-Maxl-Match

Le Cong Thanh

Institute of Mathematics, 18 Hoang Quoc Viet Road, 10307 Hanoi, Vietnam

Received March 25, 2008
Revised July 07, 2008

Abstract. We consider the approximating behaviour of greedy algorithms, in terms of performance ratios, for the maximum independent set and minimum maximal matching problems, two classical NP-hard problems. The main results tend to confirm our observation that the performance of algorithms in almost every-case is generally much better than worst-case performance. In particular, we show that for almost every instance of the minimum maximal matching problem the greedy algorithm finds a feasible solution that is near-optimal.

2000 Mathematics Subject Classification: 68Q17.
Keywords: Independent set, matching, greedy algorithm, performance ratio.

1. Introduction

Greedy algorithms are simple and straightforward. They are shortsighted in their approach in the sense that they take decisions on the basis of information at hand without worrying about the effect these decisions may have in the future. They are easy to invent, easy to implement and most of the time quite efficient. However, many problems cannot be solved correctly by greedy approach. Greedy algorithms are often used to solve approximately optimization problems. They find in polynomial time very good solutions of some problems, but may find less-than-optimal solutions for some instances of other problems.

Approximation algorithms are usually evaluated by analysing the performance on some particular instances. Hence performance guarantees are in their nature worst-case bounds. Moreover, approximation algorithms often behave significantly better in practice than their guarantees would suggest. As an alternative to the "worst-case" performance guarantee approach, one might therefore attempt to do performance analysis from an "average-case" and "almost every-case" points of view. Of course "average-case" results do not tell us anything about how algorithms will perform for particular instances, whereas "worst-case" guarantees at least provide a bound on this performance. However, performance analysis from "almost every-case" point of view gives us information about performance guarantees for approximation algorithms on almost all instances one expects to encounter in practice.

Thus, if you wish to get as much information as possible about how our approximation algorithm will behave in practice, it is probably best to analyse them in as many ways as possible, from both the "worst-case" and the "almost every-case" points of view.

On the basis of such observations, in [6] the approximating behaviour of algorithms in almost every-case was proposed and elaborated on the view for the MAX-CUT problem. The present work deals with an analysis of greedy algorithms for some fundamental optimization NP-hard problems in graph theory. The main results tend to confirm our observation that the behaviour of algorithms in almost every-case is generally much better than worst-case behaviour.

In Sec. 2, notations concerning performance guarantees, in terms of ratios, for approximation algorithms will be recalled, and the performance guarantee for the greedy algorithms for the maximum independent set and the minimum vertex cover problems will be provided. The performance of the greedy algorithm for the minimum maximal matching problem will be analysed in Sec. 3.

2. Notations and Examples

Let us begin by presenting some definitions and notations concerning the performance of approximation algorithms for optimization problems with graph instances, as in [2, 6].

Let Π be an optimization problem with the instance set D_Π of all finite simple loopless undirected graphs, and let $A|\Pi$ be an approximation algorithm for Π. For any instance G in D_Π, the value of an optimal solution of the problem Π for G will be denoted by $OPT_\Pi(G)$, and the value of the feasible solution found by $A|\Pi$ when applied to G will be denoted by $A|\Pi(G)$.

If Π is a minimization (resp., maximization) problem, and if G is any instance in D_Π, then the *performance ratio* $R_{A|\Pi}(G)$ of an approximation algorithm $A|\Pi$ on an instance G is defined by

$$R_{A|\Pi}(G) = \frac{A|\Pi(G)}{OPT_\Pi(G)} \quad \left(\text{resp., } R_{A|\Pi}(G) = \frac{OPT_\Pi(G)}{A|\Pi(G)}\right).$$

The *absolute performance ratio* $R_{A|\Pi}$ and the *almost sure performance ratio* $R^{as}_{A|\Pi}$ of an approximation algorithm $A|\Pi$ are given by

$$R_{A|\Pi} = \inf\left\{r \geq 1 : R_{A|\Pi}(G) \leq r \text{ for every instance } G \in D_\Pi\right\}$$

and

$$R^{as}_{A|\Pi} = \inf\left\{r \geq 1 : R_{A|\Pi}(G) \leq r \text{ for almost every instance } G \in D_\Pi\right\}.$$

Essentially the absolute performance ratio is the approximating performance of an algorithm in the worst-case, and the almost sure performance ratio gives us the approximating behaviour of this algorithm in almost every-case, i.e., for almost every problem instance.

As an example, let us consider the approximating behaviour of a greedy algorithm for the maximum independent set (Max-IS) problem.

An *independent set* in a graph is a collection of vertices such that no two vertices are adjacent. The *Max-IS problem* asks to find an independent set of maximum cardinality. It is known that for the Max-IS problem there is no polynominal time approximation algorithm A with the absolute performance ratio $R_A < \infty$ unless $P = NP$ (see [3]).

The *greedy algorithm* $Gr|IS$ for the Max-IS problem consists in finding a collection of non-adjacent vertices that is maximal (with respect to vertex inclusion) by starting with some vertex and iteratively adding vertices until it is no longer possible. As for other approximation algorithms for the Max-IS problem, the algorithm $Gr|IS$ has the absolute performance ratio $R_{Gr|IS} = \infty$. Indeed, for example, it is obvious that for the input complete bipartite graph $K_{l,m}$ with $l = o(m)$, the performance ratio $R_{Gr|IS}(K_{l,m}) = \infty$.

Clearly, for a given graph the algorithm $Gr|IS$ always produces in polynomial time a maximal independent set. Hence, in order to analyse the performance of the algorithm $Gr|IS$ on almost every graph instance G, we need an upper bound for the optimal value $OPT_{IS}(G)$ and a lower bound for the value $Gr|IS(G)$. These bounds can be obtained by estimating the size of any maximal independent set. Some results concerning this estimation have been obtained (see for example [4, 5]). Here we have the following useful result: *For almost every graph G, the cardinality $|U|$ of any maximal independent set U in G satisfies*

$$\log_2 n - 2\log_2\log_2 n < |U| < 2\log_2 n - 2\log_2\log_2 n,$$

where n is the number of vertices of G. Then, the value $OPT_{IS}(G)$ of an optimal solution and the value $Gr|IS(G)$ of the feasible solution found by the algorithm $Gr|IS$ for almost every instance G of the Max-IS problem can be bounded, namely

$$OPT_{IS}(G) < 2\log_2 n - 2\log_2\log_2 n$$

and

$$Gr|IS(G) > \log_2 n - 2\log_2\log_2 n,$$

where n is the number of vertices of G. Hence, when n is sufficiently large, we have

$$R_{Gr|IS}(G) < 2 + 3\frac{\log_2 \log_2 n}{\log_2 n}.$$

Thus we are now ready to state the approximating behaviour of the greedy algorithm $Gr|IS$ for the Max-IS problem.

Proposition 1. *The greedy algorithm $Gr|IS$ for the Max-IS problem has the performance ratios such that:*

(i) $R_{Gr|IS} = \infty$,

(ii) *For almost every graph instance G*

$$R_{Gr|IS}(G) < 2 + 3\frac{\log_2 \log_2 n}{\log_2 n},$$

where n is the number of vertices of G, and hence

(iii) $R^{as}_{Gr|IS} = 2$.

Thus for almost every instance of the Max-IS problem, the greedy algorithm $Gr|IS$ finds a maximal independent set that is more than nearly half as large as the maximum possible independent set. However, for some particular instances, this algorithm produces very bad solutions that are substantially distinctive of optimal.

Now, as other examples, let us survey the approximating behaviour for the maximum clique (Max-Clique) problem and the minimum vertex cover (Min-VC) problem. These problems are quite closely related to the Max-IS problem.

The *Max-Clique problem* asks to find, for a given graph, a maximum cardinality set of vertices such that any two vertices are adjacent in the graph. Clearly, the approximating behaviour for the Max-Clique problem can be obtained much the same as for the Max-IS problem.

However, the better approximating behaviour can be obtained for the *Min-VC problem*, one is asked to find a minimum vertex cover for a given graph, i.e., a minimum cardinality set of vertices that contains at least one endpoint of each edge of the graph. This problem is closely related to the Max-IS problem by the following fact: For a graph $G = (V, E)$, the set $U \subseteq V$ is an independent set for G if and only if the complementary set $V \setminus U$ is a vertex cover for G. Furthermore, the set U is a maximal (resp., maximum) independent set for G if and only if $V \setminus U$ is a minimal (resp., minimum) vertex cover for G.

Therefore, by applying the algorithm $Gr|IS$, we have the following approximation *algorithm* $GrIS|VC$ for the Min-VC problem, which consists in finding a maximal independent set by the algorithm $Gr|IS$ and then constructing its complement with respect to the vertex set of a given graph. Thus the algorithm $GrIS|VC$ always finds any minimal vertex cover in polynomial time. Moreover, by using the above cardinality estimates for any maximal independent set, we can obtain the estimates for any minimal vertex cover. Furthermore, for almost every graph instance G of the Min-VC problem, the value $OPT_{VC}(G)$ of an optimal solution and the value $GrIS|VC(G)$ of the feasible solution found by the algorithm $GrIS|VC$ can be bounded, namely

$$OPT_{VC}(G) > n - 2\log_2 n + 2\log_2 \log_2 n$$

and

$$GrIS|VC(G) < n - \log_2 n + 2\log_2 \log_2 n,$$

where n is the number of vertices of G.

Then trivially, using these inequalities to bound the performance ratio $R_{GrIS|VC}(G)$ for almost every graph instance G, we have

Proposition 2. *The algorithm $GrIS|VC$ for the Min-VC problem has the performance ratios satisfying:*

(i) $R_{GrIS|VC} = \infty$,

(ii) *For almost every graph instance G*

$$R_{GrIS|VC}(G) < 1 + 2\frac{\log_2 n}{n},$$

where n is the number of vertices of G, and hence

(iii) $R^{as}_{GrIS|VC} = 1$.

The Min-VC problem is also solved by the simplest and best-known approximation algorithm $GrM|VC$ using the greedy maximal matching algorithm $Gr|M$, which will be considered in next section. The *algorithm $GrM|VC$* consists in finding a maximal collection of disjoint edges, by iteratively adding edges until it is no longer possible, and then gathering all vertices that are endpoints of these edges. This algorithm produces in polynomial time a vertex cover that is no more than twice as large as a minimum vertex cover. It is not difficult to show that the algorithm $GrM|VC$ has $R_{GrM|VC} = 2$. Moreover, for almost every graph G, an upper bound for the performance ratio $R_{GrM|VC}(G)$ of the algorithm $GrM|VC$ can be obtained by using the above lower bound for the optimal value $OPT_{VC}(G)$ and also the trivial upper bound for the value $GrM|VC(G)$, that is $GrM|VC(G) \leq n$, where n is the number of vertices of G.

Thus we have the following better approximating behaviour of the algorithm $GrM|VC$ for the Min-VC problem.

Proposition 3. *The algorithm $GrM|VC$ for the Min-VC problem has the performance ratios such that:*

(i) $R_{GrM|VC} = 2$,

(ii) *For almost every graph instance G*

$$R_{GrM|VC}(G) < 1 + 2\frac{\log_2 \log_2 n}{\log_2 n},$$

where n is the number of vertices of G, and hence

(iii) $R^{as}_{GrM|VC} = 1$.

Thus for the Min-VC problem, in the worst-case the algorithm $GrM|VC$ has performed much better than the algorithm $GrIS|VC$. However, in almost every-case both the algorithms have the same approximating behaviour, and basically for almost all instances of the problem they find feasible solutions that are near-optimal.

Finally, it is interesting to note that, for the Min-VC problem, no polynomial time approximation algorithm with the absolute performance ratio less than 2 is known. Furthermore, in [1] it is shown that the problem cannot be solved by a polynomial time approximation algorithm with the absolute performance ratio less than $\frac{7}{6}$ unless $P = NP$.

3. Greedy Algorithm for Min-Maxl-Match

In this section we investigate the performance of a greedy algorithm in almost every-case for the minimum maximal matching problem.

A *matching* in a graph is a collection of disjoint edges, i.e., a colection of edges with the property that no two of the edges share a common endpoint. A matching is said to be *maximal* if no other matching in the graph properly contains it. The *minimum maximal matching (Min-Maxl-Match) problem* asks to find a maximal matching of minimum cardinality. It is known that the Min-Maxl-Match problem is NP-complete and it cannot be solved by a polynominal time approximation algorithm with the absolute performance ratio less than $\frac{7}{6}$ unless $P = NP$ (see [1]).

One simple approximation algorithm for the Min-Maxl-Match problem is the *greedy algorithm* $Gr|M$. Recall that this algorithm constructs a maximal matching by starting with some edge and adding edges until it is no longer possible. It is fairly easy to see that any maximal matching has a cardinality that is at most twice the cardinality of the minimum maximal matching. Therefore, by analysing the algorithm $Gr|M$ for some particular graph instances, it follows that the absolute performance ratio $R_{Gr|M} = 2$.

Let us now analyse the performance of the algorithm $Gr|M$ in almost every-case. We consider the set $\mathcal{G}_n$ of all graphs with vertex set V of n elements and for it we write

$$\mathcal{G}_n = \{G_i | V(G_i) = V; \quad i = 1, 2, ..., p\}, \text{where } p = 2^{\binom{n}{2}}.$$

Let $K_n \in \mathcal{G}_n$ be a complete graph. Denote by $\mathcal{M}_k(n)$ the set of matchings with k edges in $K_n, 1 \leq k \leq \dfrac{n}{2}$. Clearly the set $\mathcal{M}_k(n)$ has $\dfrac{n!}{(n-2k)!k!2^k}$

elements. For this set we write

$$\mathcal{M}_k(n) = \{M_j \mid j = 1, 2, ..., q\}, \text{ where } q = \frac{n!}{(n-2k)!k!2^k}.$$

For every graph $G \in \mathcal{G}_n$ denote by $\mu_k(G)$ the number of maximal matchings with k edges in G and by $\mu_k(n)$ the mean value of $\mu_k(G)$ over $\mathcal{G}_n$, i.e.,

$$\mu_k(n) = \frac{1}{p}\sum_{i=1}^{p} \mu_k(G_i).$$

Lemma 1.

$$\mu_k(n) = \frac{n!}{(n-2k)!k!2^{\binom{n-2k}{2}+2k}}.$$

Proof. For every graph $G_i \in \mathcal{G}_n$ and every matching $M_j \in \mathcal{M}_k(n)$ we define the variable $x(G_i, M_j)$ by

$$x(G_i, M_j) = \begin{cases} 1 & \text{if } M_j \text{ is a maximal matching in } G_i, \\ 0 & \text{if otherwise,} \end{cases}$$

where $1 \leq i \leq p$ and $1 \leq j \leq q$. Then by the definition of $\mu_k(n)$ we have

$$\mu_k(n) = \frac{1}{p}\sum_{i=1}^{p}\sum_{j=1}^{q} x(G_i, M_j) = \frac{1}{p}\sum_{j=1}^{q}\sum_{i=1}^{p} x(G_i, M_j) = \frac{1}{p}\sum_{j=1}^{q} g(M_j),$$

where $g(M_j)$ is the number of graphs in $\mathcal{G}_n$ such that M_j is their maximal matching.

Clearly, for any $M_j (j = 1, 2, .., q)$,

$$g(M_j) = 2^{\binom{n}{2}-\binom{n-2k}{2}-k}.$$

Thus

$$\mu_k(n) = \frac{q}{p}2^{\binom{n}{2}-\binom{n-2k}{2}-k} = \frac{n!}{(n-2k)!k!2^{\binom{n-2k}{2}+2k}}$$

and the proof is complete. ■

The inequalities in the following lemma are obtained by some rough estimates and by applying Stirling's formula.

Lemma 2. *Let* $h = \left[\frac{n}{2} - \frac{1}{2}\sqrt{n \log_2 n}\right]$. *If* n *is sufficiently large, then for any* $k \leq h$ *the following inequalities hold:*

(i) $\mu_{k-1}(n) < \mu_k(n)$,

(ii) $\mu_k(n) \le \mu_h(n) < n^{-\log_2 \log_2 n}$.

In order to analyse the greedy algorithm $Gr|M$ in almost every-case, we now prove the following result:

Theorem 1. *For almost every graph G, any maximal matching contains at least $\frac{n}{2} - \frac{1}{2}\sqrt{n \log_2 n}$ edges, where n is the number of vertices of G.*

Proof. Let $\xi_{n,k}$ be a random variable taking the value l with the probability $H_{n,k}(l)/|\mathcal{G}_n|$, where $H_{n,k}(l)$ is the number of graphs $G \in \mathcal{G}_n$ such that $\mu_k(G) = l$. Denote by $E\xi_{n,k}$ the expectation of the variable $\xi_{n,k}$. Then we have

$$E\xi_{n,k} = \mu_k(n) \tag{1}$$

and so, for $t > 0$,

$$\mathrm{Prob}(\mu_k(G) < t.n^{-\log_2 \log_2 n}) = \mathrm{Prob}(\xi_{n,k} < t.n^{-\log_2 \log_2 n}).$$

Therefore, for any $k \le h$, by using inequalities (ii) in Lemma 2 we obtain

$$\mathrm{Prob}(\mu_k(G) < t.n^{-\log_2 \log_2 n}) > \mathrm{Prob}(\xi_{n,k} < t.\mu_k(n)) \tag{2}$$

and, by equality (1) and also Markov's inequality with $t > 1$,

$$\mathrm{Prob}(\xi_{n,k} < t.\mu_k(n)) = \mathrm{Prob}(\xi_{n,k} < t.E\xi_{n,k}) > 1 - \frac{1}{t}. \tag{3}$$

Moreover, from (2) and (3), it follows that, for any $k \le h$,

$$\mathrm{Prob}(\mu_k(G) < t.n^{-\log_2 \log_2 n}) > 1 - \frac{1}{t},$$

hence

$$\mathrm{Prob}(\sum_{k=1}^{h} \mu_k(G) < t.h.n^{-\log_2 \log_2 n}) > 1 - \frac{h}{t}.$$

Let us choose $t = n^2$. Then $t.h.n^{-\log_2 \log_2 n} \le n^{3-\log_2 \log_2 n} < 1$ when n is sufficiently large. We have

$$\mathrm{Prob}\left(\sum_{k=1}^{h} \mu_k(G) = 0\right) > 1 - \frac{h}{n^2} \longrightarrow 1 \text{ as } n \longrightarrow \infty.$$

This means that almost every graph G has no maximal matching with no more than h edges for $h = \left[\frac{n}{2} - \frac{1}{2}\sqrt{n \log_2 n}\right]$, where n is the number of vertices of the graph G. Thus the proof is complete. ∎

The result above gives us a basis to estimate the optimal values of the Min-Maxl-Match problem.

Corollary 1. *For almost every graph instance G of the Min-Maxl-Match problem, the optimal value $OPT_M(G)$ satisfies*

$$OPT_M(G) > \frac{n}{2} - \frac{1}{2}\sqrt{n \log_2 n},$$

where n is the number of vertices of G.

Any matching in a given graph G with n vertices contains at most $\frac{n}{2}$ egdes. It follows that $Gr|M(G) \leq \frac{n}{2}$, where $Gr|M(G)$ denotes the value of a feasible solution (maximal matching) found by the greedy algorithm $Gr|M$ applied to G.

We now close this section by the following conclusion about the performance of the greedy algorithm $Gr|M$ for the Min-Maxl-Match problem.

Theorem 2. *The greedy algorithm $Gr|M$ for the minimum maximal matching problem has the performance ratios satisfying:*

(i) $R_{Gr|M} = 2$,

(ii) *For almost every graph instance G*

$$R_{Gr|M}(G) < 1 + 3\sqrt{\frac{\log_2 n}{n}},$$

where n is the number of vertices of G, and hence

(iii) $R^{as}_{Gr|M} = 1$.

Thus the minimum maximal matching problem is solved appoximately by the simple greedy algorithm $Gr|M$. For almost all instances of the problem the algorithm $Gr|M$ finds near-optimal solutions, although in the worst-case it may find a maximal matching that is never more than twice as large as the minimum maximal matching.

References

1. M. Chlebìk and J. Chlebìková, Approximation hardness of minimum edge dominating set and minimum maximal matching. In *Proceedings of* 14^{th} *International Symposium on Algorithms and Computation(ISAAC)*, Lecture Notes in Computer Science, Springer-Verlag, 2003, 415-426.
2. M. R. Garey and D. S. Johnson, *Computers and Intractability - A Guide to Theory of NP-completeness*, W. H. Freeman, San Fransico, 1979.
3. D. S. Johnson, Approximation algorithm for combinational problems, *J. Comput. System Sci.* **9** (1974), 256-278.
4. C. McDiarmid, Colouring random graphs, *Annals of Operations Reseach* **1** (1980), 183-200.

5. L. C. Thanh, Estimations of some parameters of finite graphs and applications, *Elektron. Informationsverarb. Kybernet.* **13** (1977), 505-521 (in Russian).
6. L. C. Thanh, On the approximability of Max-Cut, *Vietnam J. Math.* **34** (2006), 389-395.