POLYNOMIAL TIME ALGORITHMS FOR NP-HARD PROBLEMS WHICH ARE OPTIMAL OR NEAR-OPTIMAL WITH PROBABILITY ONE

by

Routo Terada

Computer Sciences Technical Report #351

April 1979

ABSTRACT

This paper presents algorithms for five NP-hard problems: the vertex set cover of an undirected graph, the set cover of a collection of sets, the clique of an undirected graph, the set pack of a collection of sets, and the k-dimensional matching of an undirected graph. Each algorithm has its worst case running time bounded by a polynomial on the size of the problem instance. Furthermore, we show that each algorithm gives an optimal or near-optimal solution with probability one, as the size of the corresponding problem instance increases.

Polynomial Time Algorithms for NP-hard Problems which are Optimal or Near-Optimal with Probability One (*)

Routo Terada (**)

Computer Sciences Department
University of Wisconsin-Madison
Madison, WI. 53706

1. Introduction

Recent efforts in the design and analysis of algorithms have been directed toward designing the so-called "probabilistic algorithms", i.e., fast algorithms which may give an incorrect or a non-optimal solution with small probability, but which solve otherwise intractable (i.e., NP-complete or NP-hard) problems. For example, [R] gives a fast algorithm for testing whether a number is prime, [AV] and [P] give polynomial time algorithms to find Hamiltonian circuits in a graph, [GM] has an algorithm to color a graph, and [HT] and [K] give fast algorithms to solve the Euclidean Traveling Salesman Problem.

This approach has been successful even for problems for which it seems to be hard to get a solution of some guaranteed

(*) This research was supported in part by NSF Grant number MCS76-17323. It was also facilitated by the use of Theory Net (NSF Grant number MCS78-01689).

(**) Supported by Fundação de Amparo à Pesquisa do E. de S. Paulo under Grant number 75/509, and by the Institute of Mathematics and Statistics of the University of S. Paulo, Brazil.

accuracy. In this paper, we examine five NP-hard problems of this type. Although these problems have distinct structures and constraints, we have a uniform method to derive a fast probabilistic algorithm to solve each of them.

Our main result is Theorem A in Section 2. In Section 3 we consider two minimization NP-hard problems: the vertex set cover of a graph and the set cover of a collection of sets. In Section 4 we consider three maximization NP-hard problems: the clique of an undirected graph, the set pack of a collection of sets, and the k-dimensional matching of a graph. For each of these problems we present an algorithm, derived from the algorithm of Theorem A, with its worst case running time bounded by a polynomial on the size of the problem instance. Furthermore, as corollaries of Theorem A, we show that each algorithm gives an optimal or near-optimal solution with probability one, as the size of the corresponding problem instance increases.

For the problems studied in this paper, we have been unable to find in the literature any result stronger than our algorithms and the corresponding theorems on their probabilistic performances.

By designing and analyzing algorithms for different NP-hard problems using our uniform method for all of them, we intend to provide some insight on a uniform and general probabilistic approach to solve all the NP-hard problems derived from NP-complete problems, in spite of their different structural characteristics. One can certainly expect to see additional applications of our method to other NP-hard problems in the near future.

In the following sections, let $I_n = \{1,2,\ldots,n\}$. For a finite set R, let <u>random</u> (R) be a function which returns an ele-

ment of R chosen at random with equal probability among the elements of R. Let log denote the natural logarithm function let $[x]^+$ and $[x]_-$ denote the smallest integer not less than x (i.e., the ceiling of x) and the largest integer not greater than x (i.e., the floor of x), respectively.

2. Lemmas and Theorem A

Let Δ be a finite set, and let $\rho \subset \Delta \times \Delta$ be an irreflexive and symmetric relation defined on Δ. Let us say that a subset S of Δ has the property ρ iff for any $a, b \in S$, $a \rho b$.

In this section we use the following condition.

Condition A: there is a fixed p, $0<p<1$, such that for any $a, b \in \Delta$, we have $a \rho b$ with probability p, independent of other pairs of elements in Δ being related by ρ.

To prove Theorem A, we need two lemmas as follows.

Lemma 1: For $0<p<1$, $n>0$, $0<\epsilon<1$, and $b(n)=[(1-\epsilon) \log n/|\log p|]_-$ we have that

$$b(n) \{1 - 1/ n^{(1-\epsilon)}\}^{[n/b(n)]^+ -1} = o (1/n^2).$$

Proof:

We want to show that

$$n^2 b(n) \{1 - 1/n^{(1-\epsilon)}\}^{[n/b(n)]^+ -1} \to 0, \text{ as } n \to \infty. \quad (2.1)$$

Consider the log of the left-hand side of (2.1) :

$$2 \log n + \log b(n) + \{[n/b(n)]^+ -1\} \log (1 - 1/n^{(1-\epsilon)})$$

$$\leq 2 \log n + \log b(n) + \{[n/b(n)]^+ -1\} (-1/n^{(1-\epsilon)})$$

$$\sim 2 \log n + \log[(1-\epsilon)\log n/|\log p|] - \frac{n^\epsilon |\log p|}{(1-\epsilon)\log n} + 1/n^{(1-\epsilon)} \quad (2.2)$$

since $\log(1-x) \leq -x$, for $0<x<1$.

The last expression in (2.2) tends to $-\infty$, as $n \to \infty$ (the third term increases faster than the other terms), so that (2.1) is true.

Q.E.D.

The following lemma is proved in [M].

Lemma 2: Under Condition A, if M_n denotes the largest existing subset of Δ with property ρ,

$$\lim_{n \to \infty} M_n/\log n = 2/|\log p|, \text{ with probability one.}$$

(By convergence "with probability one" we mean "almost sure" convergence as defined , e.g., in [F]).

Theorem A: Under Condition A, if $|\Delta| = n$, there is an algorithm whose worst case running time is $O(q(n)n^2)$ (i.e., requiring at most $O(q(n)\ n^2)$ computational steps) , where $q(.)$ is a polynomial, such that

$$1 \geq \frac{A_n}{M_n} \geq \frac{1}{2} \quad \text{,as } n \to \infty, \text{ with probability one,}$$

where A_n denotes a subset of Δ with the property ρ computed by the algorithm, and M_n denotes the largest existing subset of Δ with the property ρ.

Proof:

Let us consider the following algorithm.

Algorithm A

(Let S and T be sets. S is the output)

(1) S := empty; T := Δ;

(2) while T is not empty do

(3) a := random (T); T := T - {a};

(4) if (for all b$\in$S, aρb) or (S = empty)

(5) then S := S $\cup$ {a} fi

(6) od

Assuming that there is an integer valued polynomial q(.) such that it takes at most q(n) number of steps to check whether aρb on line (4) above, the worst case running time of Algorithm A is $O(q(n)\ n^2)$, since in each iteration of the statements on lines (2)-(6) the cardinality of T decreases by one and the cardinality of S increases at most by one. In each iteration also, it is clear that S has the property ρ.

Let $S(\Delta,\rho)$ denote the output S when Algorithm A is applied on $\rho \subseteq \Delta \times \Delta$. As in the statement of this theorem, let $A_m = |S(\Delta,\rho)|$ if $|\Delta| = m$. Our probabilistic model will be assumed to be incremental in the sense that the sequence $A_0, A_1, A_2, \ldots$ of random variables is sampled as follows: [1] we increase $|\Delta|$ by one by augmenting Δ to get, say, $\Delta' = \Delta \cup \{a\}$ where a is not in Δ, and a is the element chosen by the function random (at line (3) of Algorithm A above) ; [2] the relation ρ is also augmented to get $\rho' \subseteq \Delta' \times \Delta'$, $\rho' = \rho \cup \rho_0$, where $\rho_0 \subseteq \Delta \times \{a\}$ and ρ_0 is sampled according to Condition A; [3] if $S(\Delta,\rho) \cup \{a\}$ has property ρ, then by Algorithm A $S(\Delta',\rho') = S(\Delta,\rho) \cup \{a\}$. Otherwise, $S(\Delta',\rho') = S(\Delta,\rho)$. Therefore, $A_{m+1} - A_m \leq 1$ for $m=0,1,2,\ldots$.

Let $s_i = \min\{|\Delta| : |S(\Delta,\rho)| = i\} = \min\{m : A_m = i\}$, for $i=1,2,3,\ldots$, and let $s_0 = 0$. Since the sequence $\{A_m : m \geq 0\}$ is

non-decreasing, the sequence $\{s_i : i \geq 0\}$ is also non-decreasing.

We now observe that if Algorithm A, at some iteration of the statements on lines (2)-(6) has $|S| = i$ (i.e., so far it has found i elements of Δ with property ρ) then p^i is the probability that the next element to be examined by the algorithm is related to all elements in S. Hence, $(1-p^i)^{j-1}$ for $j=1,2,3,\ldots$ is the probability that each of the next (j-1) elements to be examined is not related to at least one element in S. Thus we have, for all integers $i,j \geq 1$,

$$\Pr\{s_{i+1} - s_i = j\} = (1-p^i)^{j-1}p^i, \text{ and } s_1 - s_0 = 1 \qquad (2.3)$$

From (2.3), for any positive integer value k we have

$$\Pr\{(s_{i+1} - s_i) < k\} = \sum_{1 \leq j \leq k-1} \Pr\{(s_{i+1} - s_i) = j\}$$

$$= \sum_{1 \leq j \leq k-1} (1- p^i)^{j-1}p^i$$

$$= \frac{p^i\,(1-p^i)^{k-1} - 1}{(1-p^i) - 1}$$

$$= 1 - (1 -p^i)^{k-1} \qquad (2.4)$$

In the following, we want to show that, for any $\epsilon > 0$, $\sum_{0 \leq n \leq \infty} \Pr\{A_n \leq (1-\epsilon) \log n/|\log p|\}$ is finite.

For any real x, $A_n \leq x$ implies $A_n \leq [x]_-$, since A_n is an integer value. Thus, for any arbitrary $\epsilon > 0$ we have

$$\Pr\{A_n \leq (1-\epsilon) \log n/|\log p|\} \leq \Pr\{A_n \leq b(n)\} \qquad (2.5)$$

where $b(n) = [(1-\epsilon) \log n/ |\log p|]_-$ (2.6)

For any positive integer i, $A_n \leq i$ implies $s_i \geq n$ (as we noted before, the sequence $\{s_i : i \geq 0\}$ is non-decreasing). Thus we have

$$\Pr\{A_n \leq b(n)\} \leq \Pr\{s_{b(n)} \geq n\} \qquad (2.7)$$

Since $s_{b(n)} = \sum_{0 \le i \le b(n)-1} (s_{i+1} - s_i)$ for $b(n) \ge 1$, we have

$$\Pr\{s_{b(n)} \ge n\} \le \Pr\{ \bigcup_{0 \le i \le b(n)-1} (s_{i+1} - s_i) \ge n/b(n) \}$$

$$\le \sum_{0 \le i \le b(n)-1} \Pr\{(s_{i+1} - s_i) \ge n/b(n)\} \qquad (2.8)$$

For any real x, $(s_{i+1} - s_i) \ge x$ implies $(s_{i+1} - s_i) \ge [x]^+$ since $(s_{i+1} - s_i)$ is an integer value. Thus from (2.4) and (2.8)

$$\sum_{0 \le i \le b(n)-1} \Pr\{ (s_{i+1} - s_i) \ge n/b(n) \}$$

$$\le \sum_{0 \le i \le b(n)-1} \Pr\{ (s_{i+1} - s_i) \ge [n/b(n)]^+ \}$$

$$= \sum_{0 \le i \le b(n)-1} [\, 1 - \Pr\{ (s_{i+1} - s_i) < [n/b(n)]^+ \,]$$

$$= \sum_{0 \le i \le b(n)-1} (1 - p^i)^{[n/b(n)]^+ - 1}$$

$$\le b(n) \{1 - p^{b(n)}\}^{[n/b(n)]^+ - 1}$$

$$\sim b(n) \{1 - 1/n^{(1-\epsilon)}\}^{[n/b(n)]^+ - 1} \qquad (2.9)$$

since $p^{b(n)} \sim p^{(1-\epsilon) \log n/|\log p|} = 1/n^{(1-\epsilon)}$

By Lemma 1, the last expression in (2.9) is $o(1/n^2)$ so that there exists a positive integer n_0 such that

$$\sum_{0 \le n \le \infty} \Pr\{ A_n \le (1-\epsilon) \log n/|\log p| \}$$

$$\le \sum_{0 \le n \le n_0 - 1} b(n) [1-p^{b(n)}]^{[n/b(n)]^+ - 1} + \sum_{n_0 \le n \le \infty} 1/n^2 < \infty \qquad (2.10)$$

By the Borel-Cantelli lemma, (2.10) implies that, with probability one, for any choice of $\epsilon > 0$,

$$\liminf_{n \to \infty} \frac{A_n}{\log n} > (1-\epsilon)/|\log p| \qquad (2.11)$$

Since ϵ is arbitrary, (2.11) implies that

$$\liminf_{n \to \infty} \frac{A_n}{\log n} \geq 1/|\log p| \qquad (2.12)$$

with probability one.

On the other hand, by Lemma 2, M_n is such that

$$\lim_{n \to \infty} \frac{M_n}{\log n} = 2/|\log p| \text{ , with probability one.} \qquad (2.13)$$

From (2.12) and (2.13) we have

$$\liminf_{n \to \infty} \frac{A_n}{M_n} \geq \frac{1}{2} \text{ , with probability one.} \qquad (2.14)$$

Since we know that $A_n/M_n \leq 1$, the theorem follows.

Q.E.D.

3. Minimization Problems

3.1 Vertex Set Cover Problem

Let G = (V, E) be an undirected graph (V is the set of vertices, E is the set of edges). A vertex cover of G is a subset S of V such that each edge of G is incident upon some vertex in S. The vertex cover problem (VC) is to find the smallest vertex cover of G. This problem is known to be NP-hard [AHU].

Algorithm VC

(Let $V = I_n$, and let S, S', and T be sets. S is the output)

```
S := V; S' := V;
T := empty;
while S' is not empty do
   v := random (S');
   S' := S' - {v};
   if v is not connected to all vertices in T
      then T := T ⋃ {v}; S := S - {v} fi
 od
```

Clearly, the worst case running time of Algorithm VC is $O(n^2)$, and S is a vertex cover of G.

For the probabilistic analysis of Algorithm VC we assume the following (as in [AV], [GM], [M], and [P]).

Condition VC: there is a fixed p, $0<p<1$, such that any pair of vertices $\{v,v'\}$ has probability p of being a member of E, independent of other pairs of vertices being members of E.

Corollary VC: Under Condition VC, let VC(n) denote the cardinality of S computed by Algorithm VC, and let m_n denote the cardinality of the minimal vertex cover of G. Then

$$\frac{VC(n)}{m_n} \sim 1, \text{ as } n \to \infty, \text{ with probability one.}$$

Proof:

For the vertex cover problem, the set Δ of Theorem A is interpreted to be the set V, the statement $a \rho b$ to mean a "not connected to" b. Then Condition VC is equivalent to Condition A, and the set T in Algorithm VC has the property ρ. Therefore, from (2.12), since $|V| = n$, we have

$$\frac{|T|}{\log n} \geq 1/|\log p|, \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.1)$$

In Algorithm VC, $S \cup T = V$, and S and T are disjoint. Then $|S| = n - |T|$, and (3.1) implies that, as $n \to \infty$,

$$\frac{|S|}{\log n} = \frac{n}{\log n} - \frac{|T|}{\log n}$$

$$\leq \frac{n}{\log n} - \frac{1}{|\log p|}$$

$$\leq \frac{n}{\log n}, \text{ with probability one.} \qquad (3.2)$$

On the other hand, if M_n denotes the largest existing subset of V with property ρ, then Lemma 2 says that

$$\frac{M_n}{\log n} \sim 2/|\log p|, \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.3)$$

Since $m_n + M_n = n$, (3.3) implies that

$$\frac{m_n}{\log n} \sim \frac{n}{\log n} - 2/|\log p|, \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.4)$$

Then (3.2) and (3.4) imply that

$$\frac{|S|}{m_n} \leq 1 + o(1), \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.5)$$

Since we know $|S|/m_n \geq 1$, the corollary follows.

Q.E.D.

3.2 Set Cover Problem

Let n and k be positive integers such that $k = \max(3,n)$, and let $C = \{S_1, S_2, \ldots, S_n\}$ be a collection of sets of, let us say, positive integers such that $|S_i| \leq k$ for $1 \leq i \leq n$. A set cover

of C is a subcollection $S_{i_1}, S_{i_2}, \ldots, S_{i_h}$ such that

$$\bigcup_{1\le j\le h} S_{i_j} = \bigcup_{1\le j\le n} S_j .$$

The set cover problem(SC) is to find the smallest set cover of C. This problem is known to be NP-hard even if (i) $|S_i| \le m$, for a fixed $m\ge 3$, and for $1\le i\le n$ (see e.g. [GJ]); and (ii) if $s \in S_i$ for some i, $1\le i\le n$, then there is at least one $j\ne i$, $1\le j\le n$, such that $s \in S_j$ (see e.g. [AHU]). Throughout this section, we assume (ii).

Algorithm SC

(Let S, S', and T be collections of sets. S is the output)

```
S := C; S' := C;
T := empty ;
while S' is not empty do
   S_i := random (S'); S' := S' - {S_i};
   if S_i is disjoint from all sets in T
      then T := T ⋃ {S_i}; S := S - {S_i}
      fi
   od
```

Clearly, the worst case running time of Algorithm SC is $O(k^2n^2)$, and S is a set cover of C.

For the probabilistic analysis of Algorithm SC we assume the following:

Condition SC: there is a fixed p, $0<p<1$, such that given any pair of sets S_1 and S_2 in C, we have that Pr{ S_1 and S_2 are disjoint } = p, independent of other pairs of sets in C.

Corollary SC: Under Condition SC, let SC(n) denote the cardinality of the set S computed by Algorithm SC, and let m_n denote the cardinality of the minimal set cover of C. Then

$$\frac{SC(n)}{m_n} \sim 1, \text{ as } n \to \infty, \text{ with probability one.}$$

Proof:

This proof is very similar to the proof of Corollary VC.

For the set cover problem, the set Δ of Theorem A is interpreted to be the collection C, the statement a ρ b to mean a "disjoint from" b. Then Condition SC is equivalent to Condition A, and the collection T in Algorithm SC has the property ρ. Moreover, an incremental sampling of an SC-instance, as described in the proof of Theorem A, is feasible. Therefore, from (2.12), since $|C| = n$, we have

$$\frac{|T|}{\log n} \geq 1/|\log p|, \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.6)$$

In Algorithm SC, $S \cup T = C$, and S and T are disjoint. Then $|S| = n - |T|$, and (3.6) implies that, as $n \to \infty$,

$$\frac{|S|}{\log n} = \frac{n}{\log n} - \frac{|T|}{\log n}$$

$$\leq \frac{n}{\log n} - 1/|\log p|$$

$$\leq \frac{n}{\log n}, \text{ with probability one.} \qquad (3.7)$$

On the other hand, if M_n denotes the largest existing subcollection of C with property ρ, then Lemma 2 says that

$$\frac{M_n}{\log n} \sim 2/|\log p|, \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.8)$$

Since $m_n + M_n = n$, (3.8) implies that

$$\frac{m_n}{\log n} \sim \frac{n}{\log n} - 2/|\log p|, \text{ as } n \to \infty, \text{ with probability one.} \qquad (3.9)$$

Then (3.7) and (3.9) imply that

$$\frac{|S|}{m_n} \leq 1 + o(1), \text{ as } n \to \infty, \text{ with probability one.} \quad (3.10)$$

Since we know $|S|/m_n \geq 1$, the corollary follows.

Q.E.D.

4. Maximization Problems

4.1 Clique Problem.

Let G = (V, E) be an undirected graph. A clique of G is a complete subgraph of G (i.e., any pair of vertices in the subgraph is connected to each other by an edge). The clique problem (CL) is to find the largest clique of G. This problem is known to be NP-hard [AHU].

Algorithm CL

(Let n be a positive integer, let |V| = n, and let S and T be sets. S is the output)

```
S := empty; T := V;
while T is not empty do
   v:= random (T); T := T - {v};
   if S ⋃ {v} is a clique
      then S := S ⋃ {v} fi
od
```

Clearly, the worst case running time of Algorithm CL is $O(n^2)$, and S is a clique of G. (By duality, Algorithm CL may be changed to find a feasible solution to the maximum independent set problem, i.e., the problem of finding the largest set S of vertices in G such that no two vertices in S are connected. For the maximum independent set problem, an algorithm which does not select the vertices at random was independently studied in [GM],

assuming a sampling model which is not incremental).

For the probabilistic analysis of Algorithm CL, we assume Condition VC for the graph G = (V, E).

Corollary CL: Under Condition VC, let CL(n) denote the cardinality of the set S computed by Algorithm CL, and let M_n denote the cardinality of the maximal clique in G. Then

$$1 \geq \frac{CL(n)}{M_n} \geq \frac{1}{2} \quad \text{as } n \to \infty, \text{ with probability one.}$$

Proof:

For the clique problem, the set Δ of Theorem A is interpreted to be the set V, the statement a ρ b to mean a "connected to" b. Then Condition CL is equivalent to Condition A, and the set S in Algorithm CL has property ρ.

Then Theorem A directly implies this corollary.

Q.E.D.

4.2 Set Packing Problem

Let k and n be positive integers such that k = max (3,n), and let $C = \{S_1, S_2, \ldots S_n\}$ be a collection of sets of, let us say, positive integers such that $|S_i| \leq k$ for $1 \leq i \leq n$. A set pack of C is a subcollection $S_{i_1}, S_{i_2}, \ldots, S_{i_h}$ of pairwise disjoint sets. The set packing problem (SP) is to find the largest set pack of C. This problem is known to be NP-hard, even if $|S_i| \leq m$, for a fixed $m \geq 3$ and for $1 \leq i \leq n$. (see, e.g., [GJ]).

Algorithm SP

(Let S and T be sets. S is the output)

S := empty; T := C;

while T is not empty do

S_i := random (T); T := T - $\{S_i\}$;

if S_i is disjoint from all sets in S

then S := S $\cup$ $\{S_i\}$ fi

od

Clearly, the worst case running time of Algorithm SP is $O(k^2n^2)$, and S is a set pack of C.

For the probabilistic analysis of Algorithm SP, we assume Condition SC for the collection C.

Corollary SP: Under Condition SC, let SP(n) denote the cardinality of the set S computed by Algorithm SP, and let M_n denote the cardinality of the maximal set pack of C. Then

$$1 \geq \frac{SP(n)}{M_n} \geq \frac{1}{2} \quad \text{, as } n \to \infty, \text{ with probability one.}$$

Proof:

For the set packing problem, the set Δ of Theorem A is interpreted to be the collection C, the statement a ρ b to mean a "disjoint from" b. Then Condition SC (as in the proof of Corollary SC) is equivalent to Condition A, and the collection S in Algorithm SP has the property ρ. Moreover, an incremental sampling of an SP-instance, as described in the proof of Theorem A, is feasible.

Then Theorem A directly implies this corollary.

Q.E.D.

4.3 k-Dimensional Matching Problem

Let k and n be positive integers such that $k = \max(3,n)$, and let $A_1 = \{a_{11}, a_{12}, \ldots, a_{1n}\}$, $A_2 = \{a_{21}, a_{22}, \ldots, a_{2n}\}$, $\ldots$, $A_k = \{a_{k1}, a_{k2}, \ldots, a_{kn}\}$ be pairwise disjoint sets, and let T be a subset of $A_1 \times A_2 \times \ldots \times A_k$, with $|T| = n$. A matching of T is a subset S of T such that no two elements of S agree in any coordinate. The k-dimensional matching problem (DM) is to find the largest matching of T. This problem is known to be NP-hard even if we have a fixed $k = 3$ [GJ].

Algorithm DM

(Let S and U be sets. S is the output)

```
S := empty; U := T;
while U is not empty do
   u := random (U); U := U - {u};
   if S ⋃ {u} is a matching of T
      then S := S ⋃ {u} fi
od
```

Clearly, the worst case running time of Algorithm DM is $O(k\ n^2)$, and the set S is a matching of T.

For the probabilistic analysis of Algorithm DM we assume the following:

Condition DM: there is a fixed p, $0<p<1$, such that, given any pair of elements t_1 and t_2 in T, we have that Pr {t_1 and t_2 disagree in all k coordinates} = p, independent of other pairs of elements in T.

Corollary DM: Under Condition DM, let DM(n) denote the cardinality of S computed by Algorithm DM, and let M_n denote the cardinality of the maximal matching of T. Then

$$1 \geq \frac{DM(n)}{M_n} \geq \frac{1}{2} \quad \text{, as } n \to \infty, \text{ with probability one.}$$

Proof:

For the matching problem, the set Δ of Theorem A is interpreted to be the set T, the statement a ρ b to mean a "disagree in all k coordinates with" b. Then Condition DM is equivalent to Condition A, and the set S in Algorithm DM has the property ρ. Moreover, an incremental sampling of a DM-instance, as described in the proof of Theorem A, is feasible.

Then Theorem A directly implies this corollary.

Q.E.D.

References

[AHU] A.V. Aho, J.E. Hopcroft and J.D. Ullman [1975]: "The Design and Analysis of Computer Algorithms", Addison Wesley.

[AV] D. Angluin and L.G. Valiant [1977]: "Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings", 9th. ACM Symp. on Th. of Comp., 30-41.

[F] W. Feller [1968]: "An Introduction to Probability Theory and its Applications", Wiley.

[GJ] M.R. Garey and D.S. Johnson [1978]: "Computers and Intractability", Freeman.

[GM] G.R. Grimmett and C.J.H. McDiarmid [1975]: "On Colouring Random Graphs", Math. Proc. Cambridge Philo. Soc., 77:313-324.

[HT] J.H. Halton and R. Terada [1978]: "An Almost Surely Optimal Algorithm for the Euclidean Traveling Salesman Problem", Tech. Rept. CS 335, Univ. of Wisconsin-Madison.

[K] R.M. Karp [1977]: "Probabilistic Analysis of Partitioning Algorithms for the Traveling Salesman Problem in the Plane", Math. Op. Res.,2:3, 209-244.

[M] D.W. Matula [1976]: "The Largest Clique Size in a Random Graph", Tech. Rept. CS 7603, Southern Metho. Univ..

[P] L. Posa [1976]: "Hamiltonian Circuits in Random Graphs", Discr. Math., 14, 359-364.

[R] M. O. Rabin [1977]: "Probabilistic Algorithms", in "Algorithms and Complexity", ed. by J.F. Traub, Academic Press, 21-39.