Maximum bounded 3-dimensional matching is MAX SNP-complete

Viggo Kann

Royal Institute of Technology, Stockholm

`viggo@nada.kth.se`

Abstract

We prove that maximum three dimensional matching is a MAX SNP-hard problem. If the number of occurrences of elements in triples is bounded by a constant the problem is MAX SNP-complete. As corollaries we prove that maximum covering by 3-sets and maximum covering of a graph by triangles are MAX SNP-hard. The problems are MAX SNP-complete if the number of occurrences of the elements and the degree of the nodes respectively are bounded by a constant.

Keywords

Approximation, combinatorial problems, computational complexity, MAX SNP-complete problems.

Introduction

Identifying which combinatorial problems are easy to solve and which are hard is an important and challenging task. One of the most accepted ways to prove that a problem is hard is to prove it NP-complete. If an optimisation problem is NP-complete we are almost certain that it cannot be solved optimally in polynomial time. In practice though, it is often sufficient to find an approximative solution, which is near the optimal solution, and in many cases this can be done quite fast.

For example the TSP (Travelling Salesman Problem) with triangular inequality is NP-complete, but in polynomial time it can be solved approximately within a factor $\frac{3}{2}$, i.e. one can find a trip of length at most $\frac{3}{2}$ times the shortest trip possible [Ch]. Another even more striking example is the bin-packing problem, which is NP-complete but can be approximated within every constant in polynomial time [KaKa]. Such a scheme for approximating within every constant is called a PTAS (Polynomial Time Approximation Scheme).

In [PaYa] Papadimitriou and Yannakakis defined (syntactically) a complexity class, called MAX SNP, together with a concept of reduction, called L-reduction, which preserves approximability with constants. All problems in MAX SNP can be approximated within a constant, because MAX SNP is closed under L-reduction, but many are not known to admit a PTAS. Several problems were shown to be complete in MAX SNP, for example maximum 3-satisfiability and maximum independent set in a graph with bounded degree. All attempts to construct a PTAS for a MAX SNP-complete problem have failed and hence it seems reasonable to conjecture that no such scheme exists, in particular since if one problem had a PTAS then every problem in MAX SNP would admit a PTAS.

The general feeling is that some NP-complete problems may be harder than others to approximate in polynomial time. An important task is to classify the classical NP-complete problems according to approximability. As seen above we know that some of them have a PTAS, some are in MAX SNP and some seem to be even harder to approximate, for example unbounded maximum independent set [BeSch].

In this paper we show that the optimisation version of the NP-complete 3-dimensional matching problem is MAX SNP-hard. This is shown by a reduction from the problem of maximum bounded 3-satisfiability, which is known to be MAX SNP-complete [PaYa]. By reducing from 3-dimensional matching we also show that the optimisation versions of two related problems, covering by 3-sets and node covering by triangles, are MAX SNP-hard.

Definition of the treated problems

MAX 3DM$-B$ *Maximum bounded 3-dimensional matching.*
Instance: A set $M \subseteq W \times X \times Y$ of ordered triples where W, X and Y are disjoint sets. The number of occurrences in M of an element in W, X or Y is bounded by a constant B.
Problem: Find the largest matching, that is, the largest subset $M' \subseteq M$ such that no two elements of M' agree in any coordinate.

MAX 3SAT$-B$ *Maximum bounded 3-satisfiability.*
Instance: A set $U = \{u_1, ..., u_n\}$ of variables and a multiset $C = \{c_1, ..., c_m\}$ of disjunctive clauses, each involving at most three literals (a variable or a negated variable) and such that the total number of occurrences of each variable is bounded by a constant B.
Problem: Find the truth assignment for U that satisfies the greatest number of clauses in C.

INDEPENDENT SET$-B$ *Maximum bounded independent set.*
Instance: A graph $G = (V, E)$ with degrees of nodes bounded by a constant B.
Problem: Find the largest independent set in V.

MAX 3SC$-B$ *Maximum bounded covering by 3-sets.*
Instance: A collection C of subsets of a finite set A where every $c \in C$ contains three elements and every element $a \in A$ is contained in at most B of the subsets in C.
Problem: Find the largest covering $C' \subseteq C$ of A. A covering is here a collection of mutually disjoint sets.

MAX COVERING BY TRIANGLES$-B$ *Maximum covering of a bounded graph by triangles.*
Instance: A graph $G = (V, E)$ with degrees of nodes bounded by a constant B.
Problem: Find the largest covering of G by triangles, that is, the largest collection $\{V_i\}$ of mutually disjoint 3-sets of nodes such that every V_i induces a 3-clique in G.

Definitions of MAX SNP and L-reductions

MAX SNP is the class of optimisation problems P such that $OPT(I) = \max_S \left|\{\overline{x} : \Psi_P(\overline{x}, S, I)\}\right|$ where Ψ_P is a quantifier free formula, I an instance of P encoded as a relation, $OPT(I)$ the optimal measure for I and S the optimal relation sought.

For example, assume that an instance I of MAX 3SAT is specified by four relations A_0, A_1, A_2, A_3 where A_i contains all clauses with i negative literals and $(x_1, x_2, x_3) \in A_i$ means that there is a clause with $x_1, ..., x_i$ appearing negatively and $x_{i+1}, ..., x_3$ appearing positively. Setting $\overline{x} = (x_1, x_2, x_3)$ and $\Psi = \big((x_1, x_2, x_3) \in A_0 \Rightarrow x_1 \in S \vee x_2 \in S \vee x_3 \in S\big) \wedge ... \wedge \big((x_1, x_2, x_3) \in A_3 \Rightarrow x_1 \notin S \vee x_2 \notin S \vee x_3 \notin S\big)$ we see that MAX 3SAT is in MAX SNP. (Here S is the set of true variables.)

Let Π_1 and Π_2 be two optimisation problems and let $f : \Pi_1 \to \Pi_2$ be a polynomial-time transformation. f is an *L-reduction* if there are positive constants α, β such that for every instance I of Π_1

1. $OPT\big(f(I)\big) \leq \alpha OPT(I)$,
2. for every solution of $f(I)$ with measure c_2 we can in polynomial time find a solution of I with measure c_1 such that $\left|OPT(I) - c_1\right| \leq \beta \left|OPT\big(f(I)\big) - c_2\right|$.

Papadimitriou and Yannakakis have shown that the composition of L-reductions is an L-reduction and that if Π_1 L-reduces to Π_2 and there is a polynomial-time approximation algorithm for Π_2 with worst-case error ϵ, then there is a polynomial-time approximation algorithm for Π_1 with worst-case error $\alpha\beta\epsilon$. MAX 3SAT and even MAX 3SAT$-B, B \geq 4$ are complete for MAX SNP under L-reductions [PaYa].

Theorem 1: MAX 3DM$-B, B \geq 3$ is MAX SNP-complete.

In the proof we will use a polynomial-time transformation f from MAX 3SAT$-B$ to MAX 3DM$-B$. It is an extension of the reduction used to prove that 3DM is NP-complete in [GaJo]. In that proof, for each variable a ring of triangles is formed (see fig 1), the triangles alternately represent the variable and its negation, so that a perfect matching corresponds to fixing a truth value. The rings are interconnected by triangles representing clauses. For optimisation, the problem with this reduction is that, since we allow solutions which are not perfect matchings, a better matching may result if, instead of including every other triple in the rings, we include many clause triples (see fig 1). We are able to solve this problem by creating many rings for every variable, and connecting these rings with binary trees.

Fig 1 *Two matchings of the structure described in [GaJo] corresponding to the clauses* $\{x, x, \overline{x}, \overline{x}\}$. *Each triangle is a triple in* M *and each dot is an element in either* W, X, *or* Y. *The matched triples are marked with rings. The left matching is optimal (with 7 matched triples) but the right is the matching corresponding to* x *true (with 6 matched triples).*

Let I be an instance of MAX 3SAT$-B$ with $U = \{u_1, ..., u_n\}$ and $C = \{c_1, ..., c_m\}$.
Let d_i be the total number of occurrences of u_i in C, either as u_i or $\overline{u}_i$. We know that $d_i \leq B$ for each i.
Let $K = 2^{\lfloor \log_2(\frac{3}{2}B+1) \rfloor}$, i.e. the largest power of two such that $K \leq \frac{3}{2}B + 1$.

For every variable u_i we construct K identical rings, each ring containing $2d_i$ *ring triples*, connected as in fig 2. The free elements in the ring triples of ring k are called $v_i[\gamma, k]$ and $\overline{v}_i[\gamma, k]$ $(1 \leq \gamma \leq d_i, 1 \leq k \leq K)$ as in fig 2.

Fig 2 *The first of* K *rings for a variable* u_i *with* d_i*=4 occurrences. The shaded triangles are the same as the shaded triangles in fig 3.*

The K rings are connected by *tree triples* in $2d_i$ binary trees in such a way that the elements $v_i[1,1], v_i[1,2], ..., v_i[1,K]$ are leaves in the first tree and the root of this tree is called $u_i[1]$, $\overline{v}_i[1,1], ..., \overline{v}_i[1,K]$ are leaves in the second tree and the root is called $\overline{u}_i[1]$, et cetera. Thus there are $2d_i(2K-1)$ triples which originate in the single variable u_i (There are K rings with $2d_i$ ring triples in each which are connected with $2d_i$ binary trees with $K-1$ tree triples in each.) This structure we call the *ring of trees* corresponding to u_i.

Finally *clause triples* connect some of the roots. For each clause c_j we introduce two new elements $s_1[j]$ and $s_2[j]$ and (at most) three triples connecting them to the appropriate root elements. If the variable u_i

occurs in this clause and this is its γ-th occurrence in C then root element in the triple is to be $u_i[\gamma]$ or $\overline{u}_i[\gamma]$, depending on whether the occurrence was u_i or $\overline{u}_i$ and whether the binary tree contains an even or odd number of levels.

Now we have constructed all elements and triples which are needed to define the transformation f.

Fig 3 *An example of binary trees for u_i and the adjacent clause triple and ring triples where the first occurrence of u_i in C is in the 17-th clause. The triples are marked with R, T or C for ring, tree and clause triples, respectively. If u_i occurs as u_i and the tree contains an even number of levels (as in this example) or if u_i occurs as $\overline{u}_i$ and the tree contains an odd number of levels then the clause triple is connected to the root of the left tree. Otherwise it is connected to the right tree.*

For the ring, tree and clause triples to define $M \subseteq W \times X \times Y$ we have to label the elements with W, X or Y. All trees are labelled identically in the following way. Start at the root, label it W, and label the elements in every tree triple W, X and Y anti-clockwise. The ring triples are labelled anti-clockwise in u_i-trees and clockwise in $\overline{u}_i$-trees, in some suitable planar representation of each ring. $s_1[j]$ and $s_2[j]$ are labelled X and Y respectively. In this way every element gets a unique label, see fig 4.

Fig 4 *An example of element labelling in a tree with two levels. Dots which represent identical elements are connected with arcs.*

Now the transformation f is fully described. It contains $\sum_{i=1}^{n} d_i$ clause triples, $\sum_{i=1}^{n} K \cdot 2d_i$ ring triples and $\sum_{i=1}^{n} 2d_i \cdot (K-1)$ tree triples, that is

$$|M| = \sum_{i=1}^{n} d_i + \sum_{i=1}^{n} 2d_i(K+K-1) \leq 3m + \sum_{i=1}^{n} 2d_i(3d_i+1) \leq 3m + \sum_{i=1}^{n} 6B^2 + 2 \cdot 3m = 9m + 6B^2 n.$$

f can be computed in polynomial time in m and n. Moreover, every element in a ring or tree triple occurs exactly two times in M, except for half of the root elements $u_i[\gamma]$ which occur only once. The only remaining elements are $s_1[j]$ and $s_2[j]$ in the clause triples; they occur at most three times each, because a clause contains at most three literals. Thus $f(I)$ is a problem in MAX 3DM$-B$ for $B \geq 3$.

We now study three dimensional matchings of the ring of trees corresponding to the variable u_i. First we show that in an optimal matching of this structure all rings belonging to the same variable are matched in the same way, i.e. all triples in the same positions (e.g. the ring triples which contain $v_i[1, k], 1 \leq k \leq K$) are either in the matching or not in the matching. We say that a triple is *chosen* if it is included in the matching.

Consider a *pair* of rings, that is, two rings connected to the same lowest-level tree triples. Suppose that the first ring has t_1 and the second ring t_2 chosen triples and that $t_1 \geq t_2$. Look at the matching for the two rings and the $2d_i$ tree triples that connect them. Construct a new matching of this structure from the old matching in the following way. Observe that for each of the t_1 chosen triples in the first ring, the connected tree triple cannot be chosen. Therefore we can match the second ring in the same way as the first. This matching is at least as large as the original matching. For every lowest-level tree triple, choose it if the connected ring triples are not chosen (if this is not possible because the connected tree triple on the next level is chosen, we first delete it from the matching before choosing the new tree triple). This will only make the matching larger.

Now we do the same thing one level up. We look at two adjacent pairs of rings (which are connected to the same $2d_i$ second lowest level tree triples) including the $2 \cdot 2d_i$ lowest-level tree triples which connect each pair and the $2d_i$ second lowest level tree triples which connect these tree triples. Suppose that the two rings in the first pair have been equally matched as in the preceeding paragraph and likewise that the two rings in the second pair are equally matched. Suppose that the first half (the first pair of rings and the $2d_i$ tree triples that connect them) has t_1 chosen triples and that the second half (the second pair and $2d_i$ tree triples) has t_2 chosen triples, where $t_1 \geq t_2$. In exactly the same way as above we can get a larger matching if we match all rings in the same way as in the largest ring matching, for every lowest-level tree triple choose it if the connected ring triples are not chosen and for every second lowest level tree triple choose it if the connected lowest-level tree triples are not chosen.

This procedure can continue in $\log_2 K - 1$ steps, adding a new tree level in every step. We have shown that in an optimal matching of the $2d_i(2K-1)$ triples in the ring of trees corresponding to u_i all rings must be equally matched and that the triples in every other level of the trees must be contained in the matching.

Now say that only $d_i - \delta$ triples from every ring are included in the matching but all rings are equally matched. Then we will get a matching of the ring of trees of size (# matched ring triples) + (# matched tree triples in trees connected to matched ring triples) + (# matched tree triples in trees connected to unmatched ring triples) $\leq$

$$\leq \begin{cases} (d_i - \delta)K + (d_i - \delta)\frac{K-1}{3} + (d_i + \delta)\frac{2K-2}{3} = d_i(2K-1) - \delta\frac{2K+1}{3}, & \text{if } K = 2^\alpha, \alpha \text{ even;} \\ (d_i - \delta)K + (d_i - \delta)\frac{K-2}{3} + (d_i + \delta)\frac{2K-1}{3} = d_i(2K-1) - \delta\frac{2K-1}{3}, & \text{if } K = 2^\alpha, \alpha \text{ odd.} \end{cases}$$

Obviously $\delta = 0$ gives us a maximum matching with $d_i(2K-1)$ triples. $\delta = 1$ gives us a matching of the structure which is at least $\frac{2K-1}{3}$ triples smaller than the maximum matching.

Finally, still concentrating on the i-th variable's triples, we have d_i clause triples which are situated at the root of some of the binary trees, corresponding to the clauses to which the variable belongs and whether it is negated or not in that clause. We would like the property that every other ring triple is included in the matching to be more valuable than the inclusion of any number of clause triples. Assume that we sacrifice this property by including only $d_i - \delta$ triples from every ring in the matching and that we include p clause triples instead. From these p triples one can always choose $\lceil \frac{p}{2} \rceil$ triples which occur at even distances from each other (corresponding to the fact that one of u_i or $\overline{u}_i$ appears in at least $\lceil \frac{p}{2} \rceil$ clauses). Thus we can always obtain a matching with $d_i(2K-1) + \lceil \frac{p}{2} \rceil$ triples without sacrificing the aforementioned property. We want that

$$d_i(2K-1) + \left\lceil \frac{p}{2} \right\rceil > d_i(2K-1) - \delta\frac{2K-1}{3} + p \text{ for } 1 \leq \delta \leq d_i, p \leq d_i$$

$$\Leftarrow \frac{1}{3}(2K-1) > p - \left\lceil \frac{p}{2} \right\rceil = \left\lfloor \frac{p}{2} \right\rfloor \text{ for } p \leq d_i \Leftarrow K > \frac{3}{2}\left\lfloor \frac{d_i}{2} \right\rfloor + \frac{1}{2}$$

This is true because

$$K = 2^{\lfloor \log(\frac{3}{2}B+1)\rfloor} \geq 2^{\lfloor \log(\frac{3}{2}d_i+1)\rfloor} = 2^{\lfloor \log(\frac{3}{4}d_i+\frac{1}{2})\rfloor+1} > 2^{\log(\frac{3}{4}d_i+\frac{1}{2})} = \frac{3}{4}d_i + \frac{1}{2} \geq \frac{3}{2}\left\lfloor \frac{d_i}{2} \right\rfloor + \frac{1}{2}.$$

Let A be the structure consisting of the rings of trees corresponding to all variables (i.e. A is M without the clause triples). It is easy to find a maximum matching for A, namely every other triple, and there are two ways to obtain it for every i. These two ways correspond to the truth values of u_i. With the choice of K above we know that any maximum matching of the whole problem must contain a maximum matching of the substructure A. In order to obtain the optimal matching one has to include as many of the clause triples as possible. At most one triple from every C_j can be included. A triple $(u_i[\gamma], s_1[j], s_2[j])$ can be included only if the root triple in the corresponding tree is not included, and this depends on the manner in which the substructure for variable i is matched.

We see that solving a MAX 3SAT$-B$-problem I is equivalent to solving the MAX 3DM-problem $f(I)$.

Lemma 1: The transformation f : MAX 3SAT$-B \rightarrow$ MAX 3DM-3 is an L-reduction.

Proof: Assume that I is an instance of MAX 3SAT$-B$. We have to show the two following inequalities.

1. Show that $OPT\big(f(I)\big) \leq \alpha\, OPT(I)$ for a constant $\alpha > 0$.

$$OPT\big(f(I)\big) = \sum_{i=1}^{n}\big(d_i(2K-1)\big) + OPT(I) \leq \Big(\sum_{i=1}^{n} d_i\Big)\big(2K-1\big) + OPT(I) \leq$$
$$\leq 3m\big(2\cdot(\frac{3}{2}B+1)-1\big) + OPT(I) \leq (18B+7)\, OPT(I)$$

   because $OPT(I) \geq \frac{m}{2}$ (it is always possible to satisfy at least half of the clauses in I). Thus $\alpha = 18B+7$ in the inequality above.

2. Show that for every matching for $f(I)$ of size c_2 we can, in polynomial time, find a solution of I with c_1 clauses satisfied and $OPT(I) - c_1 \leq \beta\big(OPT\big(f(I)\big) - c_2\big)$ where $\beta = 1$.
   If the given matching is not optimal on the substructure A (defined above) then it will increase in size if we make it optimal on A (as seen above). Thus we may presume that the given matching for $f(I)$ is optimal on A. By setting the variables of I as the matching indicates (i.e. by looking at the matching for every ring) we will get an approximate solution to I which satisfies c_1 clauses and $OPT(I) - c_1 = OPT\big(f(I)\big) - c_2$. ∎

In order to show that MAX 3DM$-B$ is in MAX SNP we define a new polynomial-time transformation g from MAX 3DM$-B$ to INDEPENDENT SET$-B$. Let $V = M$, that is, for every triple in M we have a node in the graph. There is an edge in E between two triples in M if they have at least one element in common. Thus, for every element in $W \cup X \cup Y$ which occurs k times in M we have a k-clique between the corresponding nodes in V.

Fig 5 *An example of a matching problem I and the corresponding independent set problem $g(I)$.*

The degree of a node in $g(I)$ is at most $3 \cdot (B-1)$ where B is the bound of occurrences in M. Thus $g(I)$ is a bounded independent set problem.

Lemma 2: The transformation g : MAX 3DM$-B \to$ INDEPENDENT SET$-\big(3(B-1)\big)$ is an L-reduction.

Proof: Assume that I is an instance of MAX 3DM$-B$ with set of triples M. Two triples in M are adjacent if and only if the two corresponding nodes in $g(I)$ are adjacent. We see that the three dimensional matching problem of I corresponds exactly to the independent set problem of $g(I)$. Thus we get $OPT\big(g(I)\big) = OPT(I)$ and for every independent set for $g(I)$ of size c we immediately have a three dimensional matching for I of size c. g is an L-reduction since both of the needed inequalities are satisfied (with $\alpha = \beta = 1$). ■

Proof of theorem 1: By Lemma 1 MAX 3SAT$-B$ L-reduces to MAX 3DM-3 and by Lemma 2 MAX 3DM$-B$ L-reduces to INDEPENDENT SET$-\big(3(B-1)\big)$. MAX 3SAT$-B, B \geq 4$ and INDEPENDENT SET$-B, B \geq 5$ are MAX SNP-complete [PaYa]. Thus MAX 3DM$-B, B \geq 3$ is MAX SNP-complete. ■

Corollary 1: MAX 3SC$-B, B \geq 3$ is MAX SNP-complete.

Proof: The difference between maximum three dimensional matching and maximum cover by three-sets is that in the former problem the elements are of different types (W, X or Y) but in the latter problem all elements are of the same type A.

Define f : MAX 3DM$-B \to$ MAX 3SC$-B$ by $A = W \cup X \cup Y$ and $C = M$. Now $OPT\big(f(I)\big) = OPT(I)$ and every solution to $f(I)$ can be translated to an equally good solution to I by f^{-1}.

Let g : MAX 3SC$-B \to$ INDEPENDENT SET$-\big(3(B-1)\big)$ be the same transformation as in the reduction from MAX 3DM$-B$, that is, two nodes in the graph are connected if the corresponding sets have a common element.
Both f and g are L-reductions and MAX 3DM$-B, B \geq 3$ and INDEPENDENT SET$-B, B \geq 5$ are MAX SNP-complete. Thus MAX 3SC$-B, B \geq 3$ is MAX SNP-complete. ■

Corollary 2: MAX COVERING BY TRIANGLES$-B, B \geq 6$ is MAX SNP-complete.

Proof: Define f : MAX 3SC$-B \to$ MAX COVERING BY TRIANGLES$-2B$ as the reduction from 3SC to PARTITION INTO TRIANGLES [GaJo].
Let $V = A \cup \bigcup_{i=1}^{|C|} \{a_i[j] : 1 \leq j \leq 9\}, E = \bigcup_{i=1}^{|C|} E_i$ where E_i is as in fig 6 if $C_i = \{x_i, y_i, z_i\}$.

Fig 6 *The edges E_i corresponding to a 3-set $\{x_i, y_i, z_i\}$ in MAX 3SC.*

It is easy to see that every solution to $f(I)$ with measure c_2 can be transformed into an equally good or better solution where all the E_i-s are covered by either triangles number $\{1,5,6,7\}$ or $\{2,3,4\}$. Suppose the solution to I obtained by selecting C_i for membership in C if and only if the triangles $\{1,5,6,7\}$ are covered in E_i has measure c_1. Then $OPT(I) - c_1 \leq OPT\big(f(I)\big) - c_2$ and

$$OPT\big(f(I)\big) = 4 \cdot OPT(I) + 3\big(|\, C\,| - OPT(I)\big) \leq OPT(I) + 3B \cdot OPT(I) = (3B+1)OPT(I)$$

Define g : MAX COVERING BY TRIANGLES$-B \rightarrow$ INDEPENDENT SET$-\big(3(B-2)\big)$ such that there is a node in the independent set graph for every triangle in the original graph and there is an edge in the independent set graph if the two corresponding triangles have at least one element in common. Then $OPT\big(g(I)\big) = OPT(I)$ and solutions can immediately be translated from one problem to the other.

Both f and g are L-reductions and MAX 3SC$-B, B \geq 3$ and INDEPENDENT SET$-B, B \geq 5$ are MAX SNP-complete. Thus MAX COVERING BY TRIANGLES$-B, B \geq 6$ is MAX SNP-complete. ∎

Discussion

We have seen that bounded maximum three dimensional matching is MAX SNP-complete, but how about the harder, unbounded problem? The transformation g above takes an unbounded MAX 3SAT problem to an unbounded INDEPENDENT SET problem. Unfortunately the latter problem is not in MAX SNP. If there is no polynomial-time approximation scheme for some MAX SNP-complete problem, then INDEPENDENT SET cannot be approximated within any constant, as shown in [BeSch]. But MAX 3DM is not this hard. The trivial algorithm in fig 7 approximates the optimal solution within 1/3. (For any triple in the found matching, consider it and all its neighbour triples. At most three of these triples can be in the optimal matching.) Thus it is still possible that the unbounded version of MAX 3DM is in MAX SNP.

$T := M; M' := \emptyset;$
while $T \neq \emptyset$ **do begin**
  $t :=$ any element in T;
  $M' := M' \cup \{t\}$;
  $T := T - \{t$ and all its neighbours in $T\}$;
end;

Fig 7 *A simple algorithm for approximating MAX 3DM. M' will contain the matching.*

Acknowledgement

I would like to thank Johan Håstad for introducing me to the problem and for many valuable discussions.

References

[BeSch] Berman P. and Schnitger G. (1989), "On the Complexity of Approximating the Independent Set Problem", *Proc 6th Annual Symposium on Theoretical Aspects of Computer Science*, 256-268.

[Ch] Christofides N. (1976), "Worst-case analysis of a new heuristic for the travelling salesman problem", Technical Report, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh.

[GaJo] Garey M. J. and Johnson D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman and Company.

[KaKa] Karmakar N. and Karp R. M. (1982), "An efficient approximation scheme for the one-dimensional bin packing problem", *Proc 23rd Annual Symposium on Foundations of Computer science.*

[PaYa] Papadimitriou Ch. H. and Yannakakis M. (1988), "Optimization, Approximation and Complexity Classes", *Proc 20th Annual ACM Symposium on Theory of Computing*, 229-234.